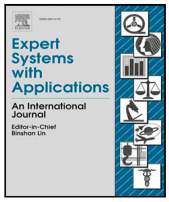
Expert Systems With Applications 250 (2024) 123773

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



LSNetv2: Improving weakly supervised power line detection with bipartite matching

Duy Khoi Tran [a,b,*], Van Nhan Nguyen [b], Davide Roverso [b], Robert Jenssen [a], Michael Kampffmeyer [a]

[a] Department of Physics and Technology, UiT The Arctic University of Norway, 9019 Tromsø, Norway
[b] Analytics Department, eSmart Systems, 1783 Halden, Norway

ARTICLE INFO

Keywords:
Line segment detection
Power line detection
Power line inspection
Deep learning

ABSTRACT

This paper addresses the crucial task of power line detection and localization in electrical infrastructure inspection using Unmanned Aerial Vehicles (UAVs) from weak supervision, polyline annotations. We first identify several limitations in the state-of-the-art approach LSNet. In particular, the inability of LSNet to detect line-crossings and lines in close proximity. To overcome these limitations, we propose LSNetv2, which enhances LSNet with multi-line segment detection capability facilitated via a bipartite matching loss. Additionally, we update LSNet's regression loss in order to stabilize training by reducing the interdependence between predicted coordinates. Finally, LSNetv2 makes use of an increased receptive field to extract global information, improving overall detection performance. Through extensive evaluations on various power line detection datasets, LSNetv2 demonstrates superior performance and robustness. On the public datasets PLDU, PLDM and TTPLA, it achieved F_β scores of 0.857, 0.875, and 0.671, respectively, while using only modified weak polyline annotation, establishing itself as an effective and efficient solution for power line detection in UAV-based electrical infrastructure inspections.

1. Introduction

Electricity is the lifeblood of modern society, thus, it is essential to ensure a stable electrical power supply across the nations. Hence, it is of utmost importance for utility companies to inspect and maintain their electrical facilities regularly. These tasks were conventionally done by human inspectors manually following, observing, and assessing the power grid (Yang et al., 2020). This inefficiency has long been recognized, and continuous efforts have been made to find more automated and unmanned solutions for the inspection tasks (Major et al., 2008, 2011). A prominent direction is to deploy Unmanned Aerial Vehicles (UAVs) to produce higher quality and more efficient observations due to their superior efficiency and ability to access higher altitudes and more hazardous environments (Deng, Wang, Huang, Tan, & Liu, 2014; Zu-jian, 2008). Additionally, to enhance the effectiveness and efficiency of inspection and maintenance tasks, there has been an increased focus on automating the assessment step of the observations obtained by the UAVs, which, in many cases, are RGB camera images (Nguyen, Jenssen, & Roverso, 2018). Towards this goal, several automatic assessment solutions have recently been proposed, many of which are fueled by the power of artificial intelligence and deep learning. Examples of such tasks are the detection of electrical components (insulators, cable suspension clamps, etc.) and the diagnosis of defects (pole breakage, insulator contamination, etc.) (Antwi-Bekoe, Zhan, Xie, & Liu, 2020; Liu, Lai, et al., 2021; Nguyen et al., 2018). Among these tasks, cable, wire, or power line recognition and localization are very crucial as power lines are one of the most essential elements in any utility infrastructure as they are directly responsible for electricity transmission. Accurate detection of power lines can facilitate better analysis of faults (tears, kinking, bird caging, etc.) and identification of hazards (vegetation encroachment, etc.) on the power lines. These tasks are vital as their failure can lead to significant negative consequences (Patnaik, 2019). Furthermore, the ability to detect power lines is imperative for UAVs, and other forms of low-altitude flights, to navigate safely.

However, it is nontrivial to detect power lines due to their inconspicuous appearances. Power lines can be very thin and might be missed by proximity sensors on UAVs. Similarly, on camera images, the width of power lines may only be a single pixel. Furthermore, cluttered backgrounds, occlusion, and same-colored backgrounds, such as white coating on snow, can cause the power lines to be imperceptible. Other

* Corresponding author at: Department of Physics and Technology, UiT The Arctic University of Norway, 9019 Tromsø, Norway.
E-mail addresses: dtr006@uit.no (D.K. Tran), nhan.v.nguyen@esmartsystems.com (V.N. Nguyen), davide.roverso@esmartsystems.com (D. Roverso), robert.jenssen@uit.no (R. Jenssen), michael.c.kampffmeyer@uit.no (M. Kampffmeyer).

https://doi.org/10.1016/j.eswa.2024.123773
Received 5 July 2023; Received in revised form 14 January 2024; Accepted 18 March 2024
Available online 24 March 2024

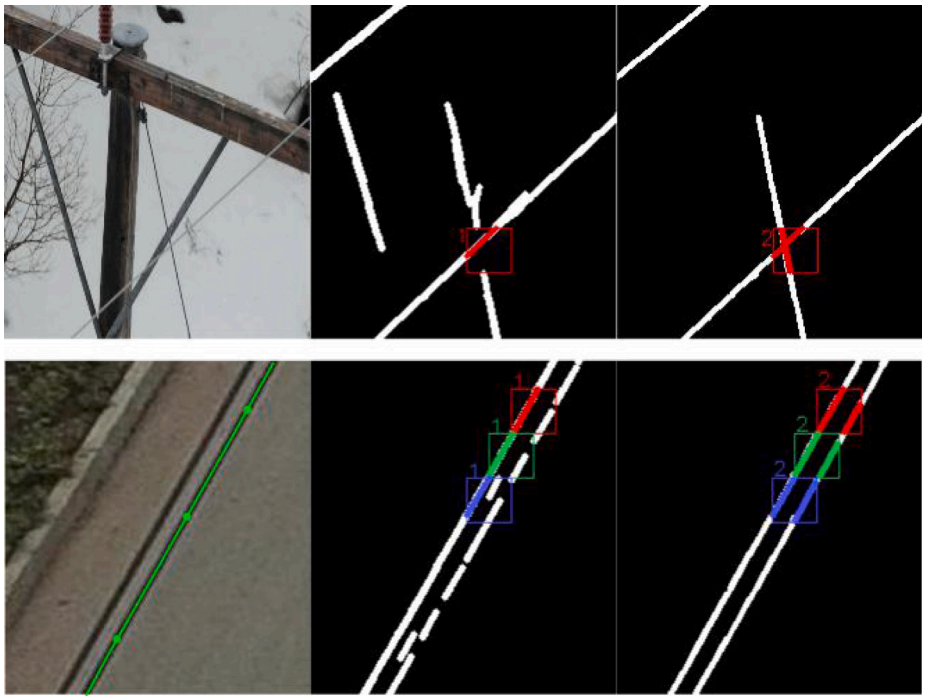


Fig. 1. LSNet and the proposed LSNetv2 divide the images into grids and infer the existence and location of power line segments in each grid cell. An example of a polyline, which is a path composed of multiple connected line segments, is shown in green in the image in the bottom row in the left column. LSNet can only output one prediction for each cell (middle column), thus, the combined output might miss some line segments, especially when multiple power lines intersect and/or are in close proximity to each other. LSNetv2 can detect multiple line segments in each grid cell (right column) and, hence, is able to produce more complete power line detections.

conditions, such as fog and lighting, can also negatively affect the discernability of power lines on images.

Towards automatic detection of power lines, early methods utilize popular classical line segment detection methods (Alpatov, Babayan, & Shubin, 2016; Candamo, Kasturi, Goldgof, & Sarkar, 2009; Golightly & Jones, 2005; Kasturi & Camps, 2002; Li, Liu, Hayward, Zhang, & Cai, 2008; Li, Liu, Walker, Hayward, & Zhang, 2010; Yan, Li, Zhou, Zhang, & Li, 2007). They rely on the assumption of clear visibility and straight line appearance of power lines. However, these methods require intricate post-processing steps to distinguish between spurious lines and power lines which can require expertise. Also, they are quite inaccurate and slow. More recently, facilitated by the increasing availability of data and hardware, deep learning has been gaining popularity for power line detection. Several deep learning approaches have been proposed that frame the power line detection task as semantic segmentation (Abdelfattah, Wang, & Wang, 2023; Jaffari, Hashmani, & Reyes-Aldasoro, 2021; Li, Xiao, Zhen, & Cao, 2019; Madaan, Maturana, & Scherer, 2017; Yang, Fan, Huo, Li, & Liu, 2022; Yang, Kong, Deng, Li, & Liu, 2023; Zhang, Yang, Yu, Zhang, & Xia, 2019). These methods can achieve quite competitive performance, however, gathering the pixel-level annotations required for these approaches is a big challenge, especially for objects with long and thin appearances like power lines. This difficulty of producing pixel-level ground truth can be problematic, as in order to obtain a power line detector that is effective and robust across various settings of input images (variations in lighting conditions, scenes, distances, power line types, utility infrastructures, etc.) a large amount of data should be procured.

To address this problem, LSNet (Line Segment Net) (Nguyen, Jenssen, & Roverso, 2020) proposes to use polylines that trace the power lines as ground truth (examples are shown in Fig. 1). These polyline annotations are cheaper to obtain, but they provide weaker supervision and can lead to imprecise width estimation of power lines due to the lack of width information in the ground truth. However, our observations indicate that in many real-world settings, predicting polylines or line segments that trace the power lines is adequate for downstream inspection tasks, as these primarily require accurate location information.

LSNet (Nguyen et al., 2020) is a Convolutional Neural Network (CNN) single-shot line segment detector that can be trained from polyline annotations. In LSNet, images are divided into overlapping grids and the detector leverages two output branches to first classify whether each cell grid has a line segment belonging to a power line and second regress the endpoints of these line segments. However, LSNet has several limitations. One major constraint is that LSNet only is able to detect and locate one line segment in each grid cell. While this design allows LSNet to obtain good performance on simpler datasets, where the power lines are arranged relatively far apart and not intersecting with each other, its design limits its application in more practical settings, where the power lines can appear visually to be of very close proximity or to cross each other resulting in cases where there are multiple line segments in a single cell. This is illustrated in Fig. 1. In addition, LSNet struggles to effectively detect more concealed line segments and discern spurious lines due to its limited ability to reason over extended image regions due to its small receptive field. Furthermore, the regression loss used to train the localization capability of LSNet can introduce training instabilities due to LSNet's swap mechanism that introduces an interdependence between the predicted endpoints of the line.

In this paper, to address the aforementioned problems and thereby improve overall model accuracy, we propose an improved version of LSNet, aptly named LSNetv2. First, to be able to detect power line crossings and power lines that are in close proximity, we provide LSNetv2 with the capability to detect multiple line segments. This is facilitated by allowing the model to make multiple predictions on each divided cell location and by leveraging a bipartite matching loss for training. In addition, to eliminate the interdependence of the line endpoints during line segment localization, we also modified the loss for the regression branch of LSNet by fixing the ordering of the endpoints, thus simplifying the model objective and facilitating better convergence. Finally, we show that a large receptive field is required in order for

power line inspection models to accurately detect the semantics of power lines and avoid misclassification of alternative lines. Based on this insight, we increase LSNetv2's capabilities by adopting a model from the state-of-the-art ConvNeXt family (Liu et al., 2022) instead of the original modified-VGG16 (Simonyan & Zisserman, 2014) to provide a larger receptive field to the detector. We empirically demonstrate the benefit of LSNetv2 on a wide range of power line detection tasks, illustrating its superiority over LSNet. In summary, the contributions are as follows:

1. We elaborately design LSNetv2 to have multi-guess capabilities. This enhances the model's ability to detect power line crossings and nearby lines.
2. We propose a new loss for the regressor branch. This ensures a more streamlined and stable training process.
3. We recognize the need for power-line inspection models to have more extensive receptive fields to reduce the need of misclassification and show that this can be achieved via a ConvNeXt backbone.
4. We conduct extensive experiments to validate the superiority of LSNetv2 over the original LSNet.

2. Related work

2.1. Traditional power line detection approaches

A notable early attempt at power line detection was proposed by Kasturi and Camps (2002), where Steger's method (Steger, 1998) was used to extract features, which are then filtered using the Hough transform to exclude short lines. Yan et al. (2007) propose instead to leverage the Radon transform to generate line segments, group these segments by slope and distance thresholding, and then finally use Kalman filters as a post-processing step. Li et al. (2010) remove clutter and noise in the background using a pulse-coupled neural filter before using Hough transforms and K-means clustering to detect and combine line segments. However, the aforementioned solutions and similar works (Alpatov et al., 2016; Candamo et al., 2009; Golightly & Jones, 2005; Li et al., 2008) hold the strict assumptions that power lines appear straight and have parallel orientations, and thus, do not always apply in reality. Song and Li (2014) aimed to address this problem and were able to detect curved power lines by using a normalized graph cut model to link line segments, which were produced from the responses of a matched filter and first-order derivative of a Gaussian. However, all of these traditional approaches are severely affected by the conditions in which the input images were taken. Any differences in camera settings, environment, lighting, and view angle require extensive tuning of hyper-parameters of these methods and require specialist expertise.

2.2. Deep learning based approaches

Deep learning aims to alleviate the problems of the abovementioned traditional computer vision approaches. With enough data, deep learning solutions for classification and detection can generalize well to different acquisition conditions. Pan, Cao, and Wu (2016) suggest training a CNN that takes in edge features, which are produced by steerable filters and classifies whether square patches from an input image contain a line or not. Then the Hough transform is used to detect the power line segments. Similarly, Gubbi, Varghese, and Balamuralidhar (2017) also propose using a CNN but, instead, use Histogram of Oriented Gradient features as input. The line segment detector proposed in Grompone von Gioi, Jakubowicz, Morel, and Randall (2010) was used as a post-processing step. Since the inputs of these two methods are individual patches, the CNN models lack the contextual information of the entire images. Hence the performance can be limited, especially for low-contrast images with a cluttered background. Yetgin, Benligiray, and Gerek (2019) finetuned a CNN, which was pretrained on ImageNet (Deng et al., 2009) via two methods. One is with a newly initialized linear layer with softmax that classifies the power line existence and is trained jointly with the feature extractor. The other method is to use dimension-reduced features from immediate layers of the pre-trained CNN to train a classifier separately. However, these two proposed methods only detect the existence of power lines for real-time warning systems without localization. More recent approaches (Abdelfattah et al., 2023; Jaffari et al., 2021; Li et al., 2019; Madaan et al., 2017; Zhang et al., 2019) frame power line detection as binary pixel-level classification problems where each pixel is classified whether it belongs to the power line or not. This type of problem can be undertaken by deep learning semantic segmentation networks. Madaan et al. (2017) investigate different dilated convolutional neural networks for segmenting power lines. Zhang et al. (2019) produce segmentation by fusing hierarchical feature maps from each layer of a VGG-16 model (Simonyan & Zisserman, 2014) as well as structure features, such as power line length, width, and orientation. In (Li et al., 2019), a CNN is introduced with two components: an information fusion module and an attention module. The information fusion module is in an encoder–decoder structure, where decoding stages are combined with their corresponding same-scaled encoding stages to fuse semantic and location information for accurate power line segmentation. The attention module produces, from the last feature map of the encoder, a weight map, which is multiplied elementwise with the decoder output to increase more focus on regions with power lines. Abdelfattah et al. (2023) trained a Generative Adversarial Network (GAN) to generate modified versions of the input images where the power lines are highlighted. A semantic decoder connected to an immediate layer of the generator is also trained jointly in order to perform the actual segmentation task. Jaffari et al. (2021) introduce a new type of focal loss based on Phi coefficient (Wang, Wang, Sun, & Chen, 2020) to improve power line segmentation performance of U-Net (Ronneberger, Fischer, & Brox, 2015)-based architectures. These semantic segmentation methods have achieved satisfying results, however, they require training data with pixel-level annotation, which can be laborious to obtain. Especially in the case of power lines, which can often be quite slender yet span across images, and require meticulousness in the annotating process. Lee et al. (2017) aims to mitigate this burden by relying only on image-level annotation from patches, which are extracted from the input images via sliding windows, to train a CNN to classify the existence of power line segments in patches of input images. Visualization of positive patches is performed using the Visualbackprop algorithm (Bojarski et al., 2018) to achieve localization. The visualizations are performed on multiple layers of the network and are merged together via bilinear interpolation and multiplication. Choi, Koo, Kim, and Kim (2021) also used image-level annotation to train classification CNN for patches on the input images. Visualbackprop visualization is performed to approximate pseudo segmentation so that another fully convolutional network can be trained to perform segmentation. However, while reducing the labeling effort required, these approaches have been reported to have low performance (Xu, Zhao, Wang, & Chen, 2023a) due to, among others, the sliding window mechanism that restricts the integration of global context when considering a given patch.

The predecessor of our proposed method, LSNet, reduces the need for expensive and labor-intensive pixel-level annotations by only relying on polyline annotations that trace along the power line. Polyline annotations require much less effort and are arguably more robust to labeling inaccuracy allowing more ground-truth images to be acquired and facilitating better deep learning models. LSNet is trained to detect and locate small line segments within the ground truth polylines, which are divided by four overlapping grids. LSNet is quite competitive in the power line detection task, however, it is unable to detect power lines that are in close proximity or appear to intersect each other. This drawback leads to LSNet not being able to guarantee the complete detection of all power lines as illustrated in Fig. 1.

While not directly addressing the power line detection problem, recent work has been conducted on addressing the wireframe parsing problem, which shares certain similarities. The wireframe parsing problem aims to find the boundary of objects, structures, and regions in the form of line segments and corresponding end-points for geometric reasoning. L-CNN (Zhou, Qi, & Ma, 2019) infer the endpoints in an end-to-end manner. From these points, proposed line segments are sampled and verified with a line of interest pooling layer (LoiPooling) that took inspiration from RoIPool (Girshick, 2015) and RoIAlign (He, Gkioxari, Dollár, & Girshick, 2017) layers from the object detection. HAWP (Xue et al., 2020) transforms the original line segment labels into Holistic Attraction Fields (HAT) where each pixel is parameterized based on its position in relation to its closest line segments. A model is trained to approximate these fields, from which line segments can be derived. HAWPv2 (Xue et al., 2022) combined the strengths of LoiPooling and HAT, along with some new techniques, to improve the performance. While the adaption of wireframe approaches to the power line detection problem is promising, we demonstrate empirically that a direct application of these methods leads to sub-optimal results.

3. Methodology

In this section, we first give a short description of LSNet and highlight its shortcomings. After that, we introduce the design of LSNetv2 and detail how it addresses these limitations.

3.1. Preliminaries: LSNet

LSNet (Nguyen et al., 2020) was proposed as a single-shot line-segment detector inspired by the Single Shot Multibox Detector (SSD) (Liu et al., 2015) and You Only Look Once (YOLO) model (Redmon, Divvala, Girshick, & Farhadi, 2015). Specifically, LSNet breaks down the problem of power line detection into detecting and locating line segments in four overlapping grids, which are superimposed onto the input image. For the case of input images having the size of 512×512, this results in a grid of 31×31 cells, each of which overlooks an area of size 32×32 in the image. For each cell of the grids, LSNet detects whether there exist parts of power lines within its borders and provides the coordinates of the line segment endpoints. These divided results can be combined to produce a complete segment map of power lines. The four-grid approach was proposed as opposed to the one-grid approach, commonly found in models such as SSD and YOLO, in order to encourage more thorough detection and localization of power lines and combat the problem of discontinuities at grid cell borders and corners. This is illustrated in Fig. 2.

To perform line segment detection, the architecture of LSNet involves a fully convolutional feature extractor (backbone) which branches out into a classifier module and a regressor module. These two modules aim to detect the presence of a line segment and the line segment endpoint coordinates respectively. The classifier module and regressor module shared a similar design and have the output of shape $B \times 31 \times 31 \times 2$ and $B \times 31 \times 31 \times 4$ respectively, with B being the batch size. The $B \times 31 \times 31$ vectors from the classifier module determine if line segments exist in their corresponding cells, and the $B \times 31 \times 31$ vectors from the regressor module determine the xy-coordinates of the endpoints of the line segments. This design leads to LSNet being able to only detect the existence and location of one line segment per cell and, thus, may not be effective in cases where more than one line segment exists in cells, such as when power lines appear to be in close vicinity or cross each other.

The classifier and regressor modules are trained with Focal loss (Lin, Goyal, Girshick, He, & Dollár, 2017) and Wing loss (Feng, Kittler, Awais, Huber, & Wu, 2017) respectively (more details are given in 3.2). The distance error, which is used for the Wing loss, is defined as the minimum L_1 loss between the ground truth pairs and the predicted pairs, where the predicted pairs are permuted to result in the minimum loss. We observe that this permutation can lead to training instabilities when training LS-Net, caused by the predictions being interdependent, resulting in sub-optimal results.

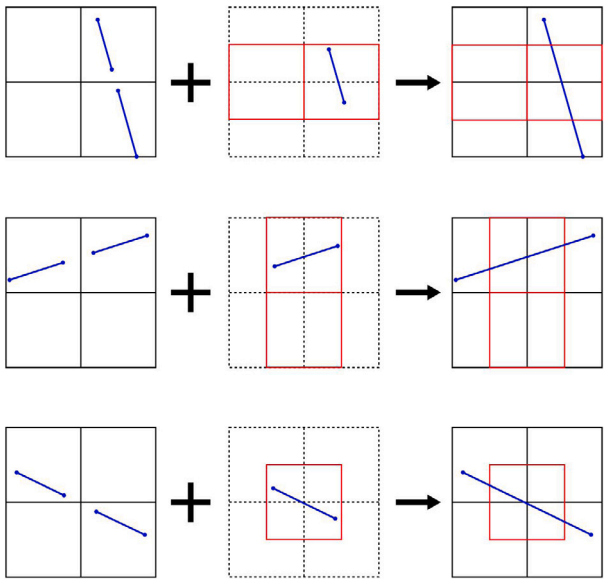


Fig. 2. Depiction of the four-grid approach. With one grid like the first column, LSNet faces the problem of disconnection at the border and corner of cells. By utilizing three additional grids (second column), the gaps can be closed and LSNet can provide more complete detections of power lines.

The feature extractor of the original LSNet was proposed to be a version of VGG-16 network (Simonyan & Zisserman, 2014) which was modified to include Group Normalization (Wu & He, 2018) before the activation functions and all max pooling layers were replaced by leveraging a stride of 2 in the convolutional layers. This architecture leads to a relatively small receptive field, only covering an area of about four times the size of a cell, resulting in inconsistent detections when context information is required, for instance when a power line is blending in with the background.

3.2. LSNetv2

In this section, we introduce our proposed LSNetv2, which aims to address the abovementioned limitations of LSNet. In practice, images captured for inspection commonly have power lines visually intersecting or being in close proximity to each other, which LSNet is unable to model.

We, therefore, design LSNetv2 to detect multiple lines per cell by performing a fixed number of inferences N, which are chosen to be larger than the maximum number of line segments that are believed to exist in the cell. The training pseudocode is shown in Algorithm 1. From our observations, $N = 10$ is a safe choice. This results in the output of the classification and regression being in the shape of $B \times 31 \times 31 \times N \times 2$ and $B \times 31 \times 31 \times N \times 4$, respectively (Fig. 3). For ease of notation, we focus the scope of our discussion on an individual cell. The ground truth y of each cell, which contains m actual line segments, is also perceived to contain N line segments $y = \{y_i\}_{i=1}^{N}$. However, the ground truth is now padded with $N-m$ negative classification. Inspired by DETR (Carion, Massa, Synnaeve, Usunier, Kirillov, & Zagoruyko, 2020), we frame the line segment detection within each cell as a direct set prediction problem. In our loss computation step, we include an optimal bipartite matching between the N predictions and N ground truths (see Fig. 4).

Let $\hat{y} = \{\hat{y}_i\}_{i=1}^{N}$ be the set of N predictions. The optimal bipartite matching between the N predictions and N ground truths, which is

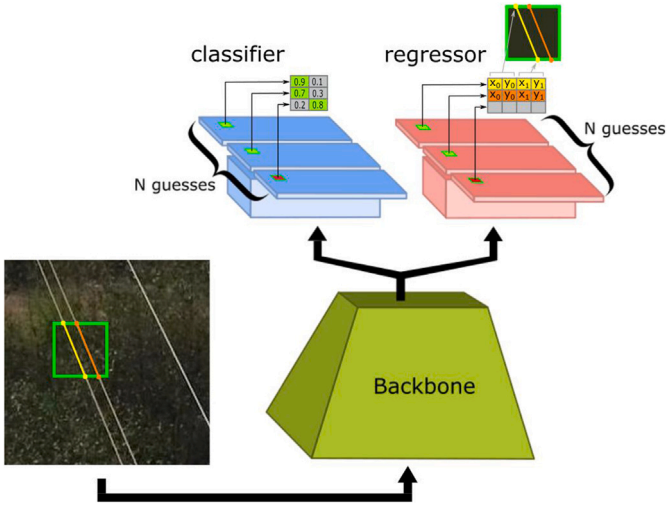


Fig. 3. LSNetv2 shares a similar overall design as LSNet with a fully convolutional feature extractor, a classifier module, and a regressor branch module. However, the two branches now perform multiple guesses so that LSNetv2 has the capability to detect multiple line segments per cell. This illustration shows that the model performs detection on the highlighted cell in the image and produces $N = 3$ guesses. Two of the guesses are classified as positive and the regressor module infers the endpoints of these positive line segments.

done using the Hungarian algorithm, produces a permutation $\hat{\sigma} \in \mathfrak{S}$ that satisfies:

$$\hat{\sigma} = \arg\min_{\sigma \in \mathfrak{S}} \sum_{i=1}^{N} \mathcal{L}_{match}(y_i, \hat{y}_{\sigma(i)}) \tag{1}$$

where $\mathcal{L}_{match}(y_i, \hat{y}_{\sigma(i)})$ is the loss when pairing $y_i = (c_i, b_i)$ and $\hat{y}_{\sigma(i)} = (\hat{p}_{\sigma(i)}, \hat{b}_{\sigma(i)})$, which is reordered via permutation $\sigma(i)$, which belongs to the set of all possible permutations $\mathfrak{S}$. This loss is defined as:

$$\mathcal{L}_{match}(y_i, \hat{y}_{\sigma(i)}) = \mathcal{L}_{match}\left[(c_i, b_i), (\hat{p}_{\sigma(i)}, \hat{b}_{\sigma(i)})\right] = -\hat{p}_{\sigma(i)}(c_i) + L_1(b_i, \hat{b}_{\sigma(i)}) \tag{2}$$

where b_i and $\hat{b}_{\sigma(i)}$ are line segment endpoint ground truth and the permutated prediction at index i respectively. $L_1(.)$ is the L_1 loss and $\hat{p}_{\sigma(i)}(c_i)$ is the predicted probability of class c_i. Here, c_i denotes the classification ground truth at index i, indicating if a line is present or not. The permutation $\hat{\sigma}$ is found by solving the linear sum assignment problem with a modified Jonker-Volgenant algorithm (Crouse, 2016). An illustration of the bipartite matching is shown in Fig. 4.

After bipartite matching has been done, the model can be trained with a multitask loss, $\mathcal{L}$, to simultaneously train the classifier and regressor module:

$$\mathcal{L} = \sum_{i=1}^{N} \mathcal{L}_{\text{cls}}(y_i, \hat{y}_i) + \lambda \sum_{i=1}^{N} \mathbb{1}_{\{c_i = \text{positive}\}} \mathcal{L}_{\text{reg}}(y_i, \hat{y}_i) \tag{3}$$

where $\hat{y}_i = \hat{y}_{\hat{\sigma}(i)}$ is one of N prediction at an arbitrary cell after bipartite matching. $\mathcal{L}_{\text{cls}}$ is a Focal loss (Lin et al., 2017):

$$\mathcal{L}_{\text{cls}} = \begin{cases} -\alpha(1-p)^{\gamma} \log(p), & \text{if } c_i = \text{positive} \\ -(1-\alpha)p^{\gamma} \log(1-p), & \text{otherwise} \end{cases} \tag{4}$$

where $p = \hat{p}_{\hat{\sigma}(i)}(\text{positive})$, $\alpha \in [0, 1]$ and $\gamma \geq 0$ are tunable hyperparameters that adjust the weight on uncommon class and misclassified examples respectively. This loss is used in the original LSNet to tackle the imbalance problem between cells with line segments and cells without. In LSNetv2, we continue to use this loss to train the classifier module.

The regressor module is trained using the average sum of Wing losses (Feng et al., 2017), which is applied on each coordinate value so that the training is more sensitive to small errors and robust against outliers. With $b_i = (b_i^{x_1}, b_i^{y_1}, b_i^{x_2}, b_i^{y_2})$ and $\hat{b}_i = \hat{b}_{\hat{\sigma}(i)} = (\hat{b}_i^{x_1}, \hat{b}_i^{y_1}, \hat{b}_i^{x_2}, \hat{b}_i^{y_2})$, the regression loss for LSNet is defined as:

$$\mathcal{L}_{\text{reg}}^{\text{LSNet}} = \frac{\min(W, W_{\text{swap}})}{4} \tag{5}$$

with

$$W = \frac{\mathcal{L}_W(b_i^{x_1}, \hat{b}_i^{x_1}) + \mathcal{L}_W(b_i^{y_1}, \hat{b}_i^{y_1}) + \mathcal{L}_W(b_i^{x_2}, \hat{b}_i^{x_2}) + \mathcal{L}_W(b_i^{y_2}, \hat{b}_i^{y_2})}{4}, \tag{6}$$

$$W_{\text{swap}} = \frac{\mathcal{L}_W(b_i^{x_1}, \hat{b}_i^{x_2}) + \mathcal{L}_W(b_i^{y_1}, \hat{b}_i^{y_2}) + \mathcal{L}_W(b_i^{x_2}, \hat{b}_i^{x_1}) + \mathcal{L}_W(b_i^{y_2}, \hat{b}_i^{y_1})}{4} \tag{7}$$

where

$$\mathcal{L}_W(m, n) = \begin{cases} w \ln(1 + |m-n|/\epsilon), & \text{if } |m-n| < w \\ |m-n| - C, & \text{otherwise,} \end{cases} \tag{8}$$

w is used to constrain the range of the nonlinear behavior of the loss to $(-w, w)$, ϵ controls the growth of loss as regression error increases and the curvature of the nonlinear part, and $C = w - w\ln(1 + w/\epsilon)$ is used to smoothen the connection between the linear and nonlinear parts of the loss.[1]

It can be seen that in the regression loss of LSNet, there is a swapping mechanism that assigns the two ground truth endpoints to the two predicted endpoints so that the regression loss will have the minimum value. We observe that this swapping mechanism, which allows a value pair in the regressor module to predict the location of an arbitrary endpoint based on its closeness to the ground truth introduces unwanted interdependence between the endpoints, resulting in training instabilities. Specifically, during training, the prediction of one endpoint relies on the position and closeness of the other point to either of the ground truth endpoints. Thus, potentially sudden and frequent changes in the ground-truth-prediction assignment can lead to inconsistent and unmeaningful weight updates. Thus, for LSNetv2, we propose the regression loss as follows:

$$\mathcal{L}_{\text{reg}}^{\text{LSNetv2}} = \begin{cases} \frac{\mathcal{L}_W(b_i^{x_1}, \hat{b}_i^{x_1}) + \mathcal{L}_W(b_i^{y_1}, \hat{b}_i^{y_1}) + \mathcal{L}_W(b_i^{x_2}, \hat{b}_i^{x_2}) + \mathcal{L}_W(b_i^{y_2}, \hat{b}_i^{y_2})}{4}, & \text{if } b_i^{x_1} < b_i^{x_2} \\ \frac{\mathcal{L}_W(b_i^{x_2}, \hat{b}_i^{x_1}) + \mathcal{L}_W(b_i^{y_2}, \hat{b}_i^{y_1}) + \mathcal{L}_W(b_i^{x_1}, \hat{b}_i^{x_2}) + \mathcal{L}_W(b_i^{y_1}, \hat{b}_i^{y_2})}{4}, & \text{otherwise} \end{cases} \tag{9}$$

This distance error ensures that the endpoints with the lower x-coordinate and the endpoints with the higher x-coordinate are detected by the same output elements in the 4-element vectors output from the regressor module. With this restriction, the training process is more robust and we empirically demonstrate the performance improvement in the experiment section.

It has been shown that LSNet has been successful for cases where images of power lines are captured from a relatively close distance and the visibility of the power lines throughout the span of the image is clear (Nguyen et al., 2020). However, in most practical application settings, images such as those used for visual inspection are taken from further away leading to segments of power lines appearing thin and/or blending in with the background. Following the grid-based approach of LSNet, the detection performance of power lines in each cell, especially ones that are inconspicuous, may depend heavily on the global information of areas around it. However, the theoretic receptive field of LSNet is only about four times the cell size. This limits the capability of LSNet to aggregate global context for the cell inference. Thus, LSNetv2 is designed to have a larger receptive field. In particular, we leverage the ConvNeXt-Tiny backbone, which increases the receptive field by a

[1] Note, unlike in Nguyen et al. (2020), where the loss is computed per dimension and then aggregated. This is more faithful to the original Wing loss formulation (Feng et al., 2017) and we observe that it empirically performs on par or improves on the formulation in Nguyen et al. (2020). Results reported for LSNet in this work include this modification.

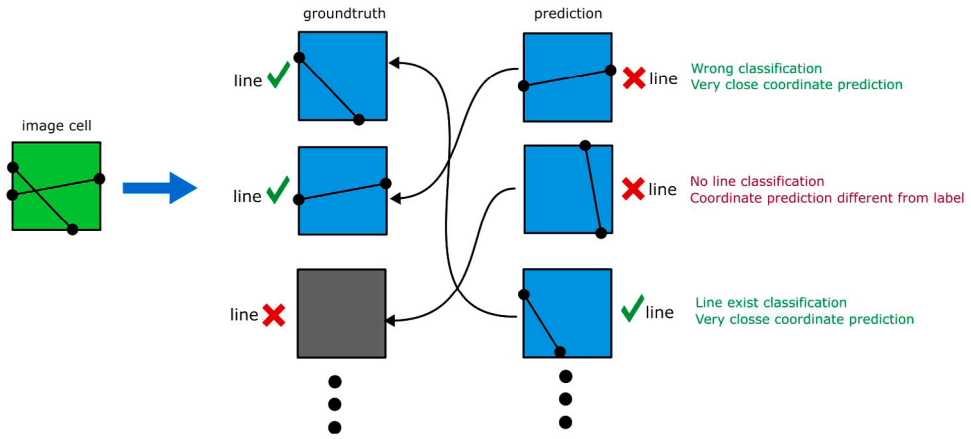


Fig. 4. Illustration of the bipartite matching mechanism that facilitates the multi-guess capability of LSNetv2. The multiple predictions of each cell are matched to the multiple ground truths based on its classification and localization of endpoints. Shown here is an image cell (green) with two line segments crossing each other. Thus, the ground truth (left column) includes N elements, two of which (marked with ✓) contain the positive classification label and the endpoint coordinates of the two line segments while the rest have negative classification labels (marked with ✗) and dummy coordinate values. The prediction (right column) also has N elements, each of which predicts if a line segment exists and the endpoint coordinates. The N elements from the prediction are matched with the N elements from the ground truth using the Hungarian algorithm based on L_{match}.

Algorithm 1: Training LSNetv2 pseudocode

Data: Training dataset: $D = \{(x_i, y_i)\}_{i=1}^{M}$
Result: Trained LSNetv2 model
Initialization:;
Initialize LSNetv2 weights θ;
Set number of epochs, learning rate and Adam hyperparameters;
Preprocess the dataset (adapt labels, resize, etc.);
for *epoch = 1 to num_epochs* **do**
  for *each minibatch* $\{(x_j, y_j)\}$ *in* D **do**
    /* Forward pass */
    $\hat{y}_j = (\hat{p}_j, \hat{b}_j) = LSNetv2\_Forward\_Pass(x_j)$;
    /* Compute Bipartite Matching with $\mathcal{L}_{match}$ and permutate */
    $\hat{y}_{\hat{\sigma}(j)} = \text{HungarianAlgorithm}(\hat{y}_j, y_j, \mathcal{L}_{match})$;
    /* Compute the loss */
    $\mathcal{L} = \sum_{i=1}^{N} \mathcal{L}_{\text{cls}}(y_i, \hat{y}_{\hat{\sigma}(j)}) + \lambda \sum_{i=1}^{N} \mathbb{1}_{\{c_i=\text{positive}\}} \mathcal{L}_{\text{reg}}(y_i, \hat{y}_{\hat{\sigma}(j)})$;
    /* Compute gradients w.r.t. the parameters */
    $\nabla_\theta \mathcal{L} = \frac{\partial \mathcal{L}}{\partial \theta}$;
    /* Update weights */
    Update parameters θ using Adam optimizer;

factor of 12. ConvNeXt-Tiny, which is the smallest in the ConvNeXt, was recently introduced as a modernized version of Resnet (He, Zhang, Ren, & Sun, 2015) which is equipped with recent properties based on the hierarchical vision transformer Swin (Liu, Lin, et al., 2021). We use an altered version of ConvNeXt-Tiny. Similarly to Resnet and other well-known CNNs, ConvNeXt-Tiny has a multi-stage design. Each stage results in the compression of the features with a ratio of 2. We use a truncated ConvNeXt-Tiny as the new backbone. The output of the truncated model reduces the input to a size of 32 × 32 × 384. The architecture is detailed in Table 5.

4. Experiments

In this section, we provide quantitative and qualitative evaluations of our proposed approach and illustrate its advantages over the current state-of-the-art approaches LSNet (Nguyen et al., 2020) and HAWPv2 (Xue et al., 2022). HAWPv2 is included in the comparison as it can be considered the state-of-the-art approach for wireframe parsing. It is an improved version of the highly-cited HAWP and, to the best of our knowledge, HAWPv2 achieved the highest performance in popular wireframe parsing benchmarks. As mentioned above, wireframe parsing is, in essence, very similar to the task of power line detection and HAWPv2 can be trained directly with the polyline annotations. Experiments are conducted on four different power line detection datasets of varying creation methods and difficulty. In addition, an analysis of different backbones was performed and ablation studies were done to highlight the benefit of the novel components that constitute LSNetv2, namely the multi-guess capability via bipartite matching, the ordered regression loss, and the new ConvNeXt-Tiny backbone.

4.1. Datasets

4.1.1. PLD-UAV

PLD-UAV (Snorker, 2019) contains two datasets of power lines: the power line dataset of urban scenes (PLDU) and the power line dataset of mountain scenes (PLDM). In these datasets, the backgrounds are urban and mountain scenes, respectively, and are relatively cluttered and complex. However, the power lines to be detected are still quite observable. In this dataset, the boundaries of power lines are annotated at the pixel level. To adapt to LSNetv2, we detected individual boundaries by dilating the pixel annotations and clustering the pixels via connected component analysis. From each cluster of pixels, a polygon is approximated and filled with white pixels (positive labels). Then, a skeletonization algorithm, introduced in Huang (2021), is used to produce polylines tracing the power lines. The polylines are simplified using the Ramer–Douglas–Peucker (RDP) algorithm (Douglas & Peucker, 1973) and imposed on the 31 × 31 grid to produce the annotation for LSNetv2. Manual inspection was done to ensure that the procedure produces accurate labels. PLDU contains 453 training data points and 120 testing data points. PLDM contains 237 training data points and 50 testing data points.

4.1.2. TTPLA

TTPLA (Abdelfattah, Wang, & Wang, 2020) is a newly introduced dataset. It consists of images taken from UAVs under a variety of conditions, such as different scenes, angles, zoom, and lighting conditions. Many instances in this dataset suffer from problems like occlusion and blending, which make the detection more challenging. TTPLA also possesses some data points that have power lines being close to each other and power line crossings. This makes it an ideal test bed to evaluate the effectiveness of our proposed method, which is designed to effectively segment multiple power lines/line segments. It should be

noted that the annotations of this dataset only include power transmission lines. Despite some other wire types also appearing in some images, these other wires are not annotated, which is different from the eSmart dataset described in the next subsection. The annotation of TTPLA is provided at an instance segmentation level via polygons that precisely wrap around the power lines. To adapt this dataset to LSNetv2, similarly to the datasets above, we performed an image processing procedure that includes skeletonizing the blobs that fill the polygon annotations to generate polylines. The polylines are simplified using the RDP algorithm and then imposed on the 31×31 overlapping grid to find intersections, which are the labels for LSNetv2. This dataset contains 1242 images and we use 992 for training and 250 for testing.

4.1.3. eSmart dataset

We also evaluated the method using a proprietary dataset of power lines aggregated by eSmart Systems. This dataset is made from UAV images taken by multiple clients of eSmart Systems, and thus, contains significant diversity in terms of scenes, angles, zooming levels, weather, and lighting conditions. This dataset has polyline annotations tracing the trajectories of the power lines. The polyline annotation can be imposed onto the 31×31 grid to produce the annotation for LSNetv2. In this dataset, the lines annotated include conductors, guy wires, and overhead ground wires. Conductors are normally in parallel. However, the other two types of wires can have different directions and, hence, there are many visual crossings. Included in the dataset are also some instances of line segments being in close proximity. The dataset contains 2961 images for training and 468 images for testing.

4.2. Implementation details

The proposed LSNetv2 is implemented in Tensorflow. Each model is trained on one NVIDIA RTX 3090 24 GB. The model was initialized using Xavier initialization (Glorot & Bengio, 2010) and trained with a batch size of 8 for 100 epochs. The Adam optimizer (Kingma & Ba, 2015) is used with a learning rate of 0.0001, a first momentum of 0.9, and a second momentum of 0.99. Empirical experiments indicate that results are robust with respect to the choice of the weight λ for the multitask loss. Following LSNet, we therefore also set λ to 1 for all datasets.. The input image size is 512×512. During training, augmentation techniques applied include randomized occurrences of sharpening, blur, color jittering, pixel dropouts, and additive noise. After that, randomized square crops of varying sizes between 360 and 512 are taken from the input and then scaled back to the input size of 512×512.

4.3. Evaluation metrics

For comparison purposes, we follow prior work (Nguyen et al., 2020) and evaluate LSNetv2 in the same manner as segmentation models. Similarly to Nguyen et al. (2020), we adopt pixel-level Averaged Recall Rate (ARR), Averaged Precision Rate (APR), and F_1 Scores. In addition, we also use averaged F_β:

$$F_\beta = \frac{1}{N}\sum_{i=1}^{N}\frac{(1+\beta^2)\text{Precision}\times\text{Recall}}{\beta^2\times\text{Precision}+\text{Recall}} \tag{10}$$

where N is the number of test images and $\beta^2 = 0.3$ in order to place more emphasis on precision compared to the conventional F_1 metric. According to Achanta, Hemami, Estrada, and Süsstrunk (2009), Cheng, Zhang, Mitra, Huang, and Hu (2011), recall is not as relevant as precision since the recall rate of 1.0 can be achieved by predicting all pixels in the image as positive. Also, higher recall rates can be an indication of imprecise predictions of line segment endpoints as illustrated in Fig. 5. Thus the F_β score can be considered a better measurement of the overall performance than the F_1 score.

To produce the segmentation map, for each cell within the 31×31 output grid that was classified to contain segments of power lines, we use OpenCV to generate visible white lines from the predicted pairs of endpoints. Specific line widths are chosen for each dataset based on the common width of the power lines for each dataset. Through inspection, we find that the common width of the power lines in the PLDU dataset is 9, 6 for the PDLM dataset, 2 for the TTPLA dataset, and 5 for the eSmart dataset.

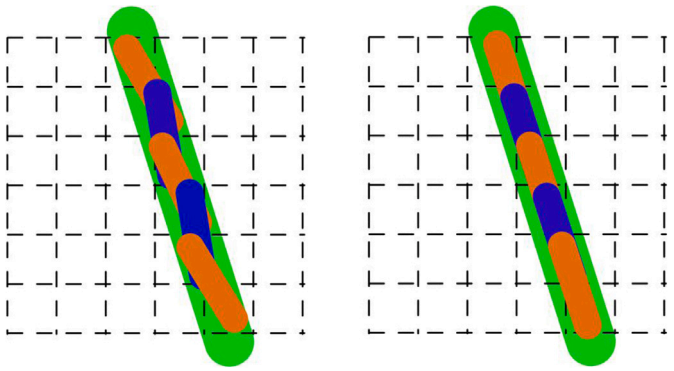


Fig. 5. Illustration of a case where imprecise inferences from LSNetv2 can leads to better recall. The orange and blue segments are individual inferences of the cells. The line segment inferences are rasterized with a width. The green lines are the ground truth rasterized with the actual line width. The left image shows that, especially in cases where the chosen line width is less than the actual one, imprecise inferences of line segments, when combined, can include more true positive pixels and less false negative than the more precise inferences shown on the right.

4.4. Evaluation methods

The comparison is done with the previously proposed LSNet across the four datasets. We also compare LsNetv2 with HAWPv2 by training the model, whose code is provided by the original author (Xue, 2021), the output of the HAWPv2 model is rasterized into segmentation maps, from which APR, ARR, F_1 and F_β score are calculated.

4.5. Main results

Results in Table 1 illustrate that LSNetv2 consistently outperforms LSNet across all datasets considered. The difference is most pronounced when considering the APR metric, leading to a performance improvement when considering the F_β metric. The performance gap is largest for the TTPLA and eSmart datasets, which can be attributed to them being arguably more complex datasets. From Fig. 6, we can see that the TTPLA dataset contains many images with several semi-parallel power lines in close proximity to each other, which showcases the effectiveness of LSNetv2 to a larger degree leading to a considerable performance gap. LSNetv2 is able to both detect more true positive line segments and semantic power lines. In the eSmart dataset, as aforementioned, due to the need to also detect guy wires, each image in this dataset usually contains at least one visual crossing. As shown in Fig. 7, unlike LSNetv2, LSNet, by design, cannot detect the two line segments at the crossings. Further, it can be observed that LSNetv2 is even better than LSNet in general one-line-segment-per-cell cases, especially when the visibility of the line segments is limited due to camera angle and distance, which makes the line segments thin, and/or background blending. This can partially be attributed to the ConvNeXt backbone by providing filters with a larger receptive field that, together with other modernized features, allows LSNetv2 to get global information across the images to make accurate inferences at each cell.

The performance gap is smaller for the PLDU and PLDM datasets since these datasets are subjectively less difficult than the other two. Similarly to the two previous cases, the improvement lies mainly in the APR metric. There are no explicit line crossings found in the test set and in cases where there are parallel power lines in close proximity, the

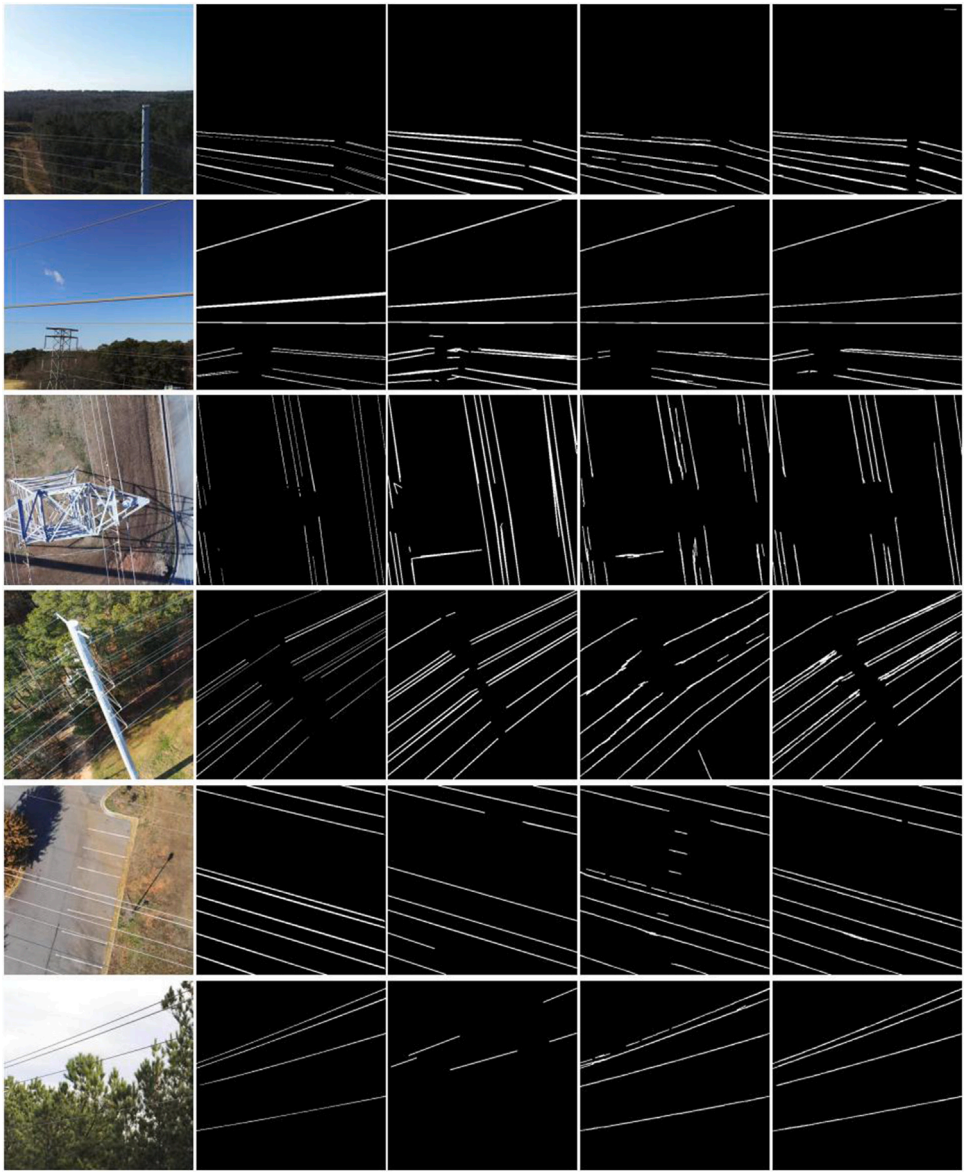


Fig. 6. The comparison of different methods on test images in TTPLA dataset.

Table 1
Performance of HAWPv2, LSNet and LSNetv2 across 4 datasets: PLDU, PLDM, TTPLA and eSmart dataset.

		HAWP (best F_1)	HAWP (best F_β)	LSNet	LSNetv2
PLDU	APR	0.830	0.924	0.914	0.938
	ARR	0.649	0.562	0.662	0.666
	F_1	0.728	0.700	0.767	0.779
	F_β	0.780	0.805	0.840	0.857
PLDM	APR	0.815	0.904	0.916	0.934
	ARR	0.664	0.578	0.726	0.724
	F_1	0.732	0.705	0.810	0.815
	F_β	0.775	0.800	0.863	0.875
TTPLA	APR	0.559	0.586	0.603	0.714
	ARR	0.639	0.567	0.618	0.560
	F_1	0.596	0.576	0.610	0.628
	F_β	0.575	0.582	0.606	0.671
eSmart	APR	0.751	0.775	0.726	0.845
	ARR	0.850	0.808	0.812	0.814
	F_1	0.797	0.791	0.766	0.829
	F_β	0.772	0.782	0.744	0.837

four-grid design of LSNet may help compensate for when a line segment is not detected by one of the four grids. However, as shown in Figs. 8–9, the missing line segment problem still exists for LSNet leading to the eventual miss-detection of the whole power line. Furthermore, close line segments may confuse LSNet resulting in imprecise localization of endpoints implied by the occasional fillings in the gaps between power lines, such as the examples shown in the first row of Fig. 8. This might cause further complications for potential downstream instance segmentation tasks. LSNetv2 is more robust to such close-proximity cases and can produce output masks with fewer missing segments and with clearer and more precise division between power lines. The last row of Fig. 8 shows that LSNet is susceptible to confusion by background lines, while LSNetv2 appears to be more resistant. This can be attributed to ConvNeXt being able to mimic the non-local self-attention mechanism of the Vision Transformer (Dosovitskiy et al., 2020), which helps gather more global information, helping LSNetv2 to better differentiate between semantic line segments.

Table 1 further shows the quantitative results of HAWPv2. We observed that the performance of HAWPv2, estimated by either F_1 or F_β, was highly dependent on different choices of line width and score threshold (used to filter out unconfident line segments). Thus,

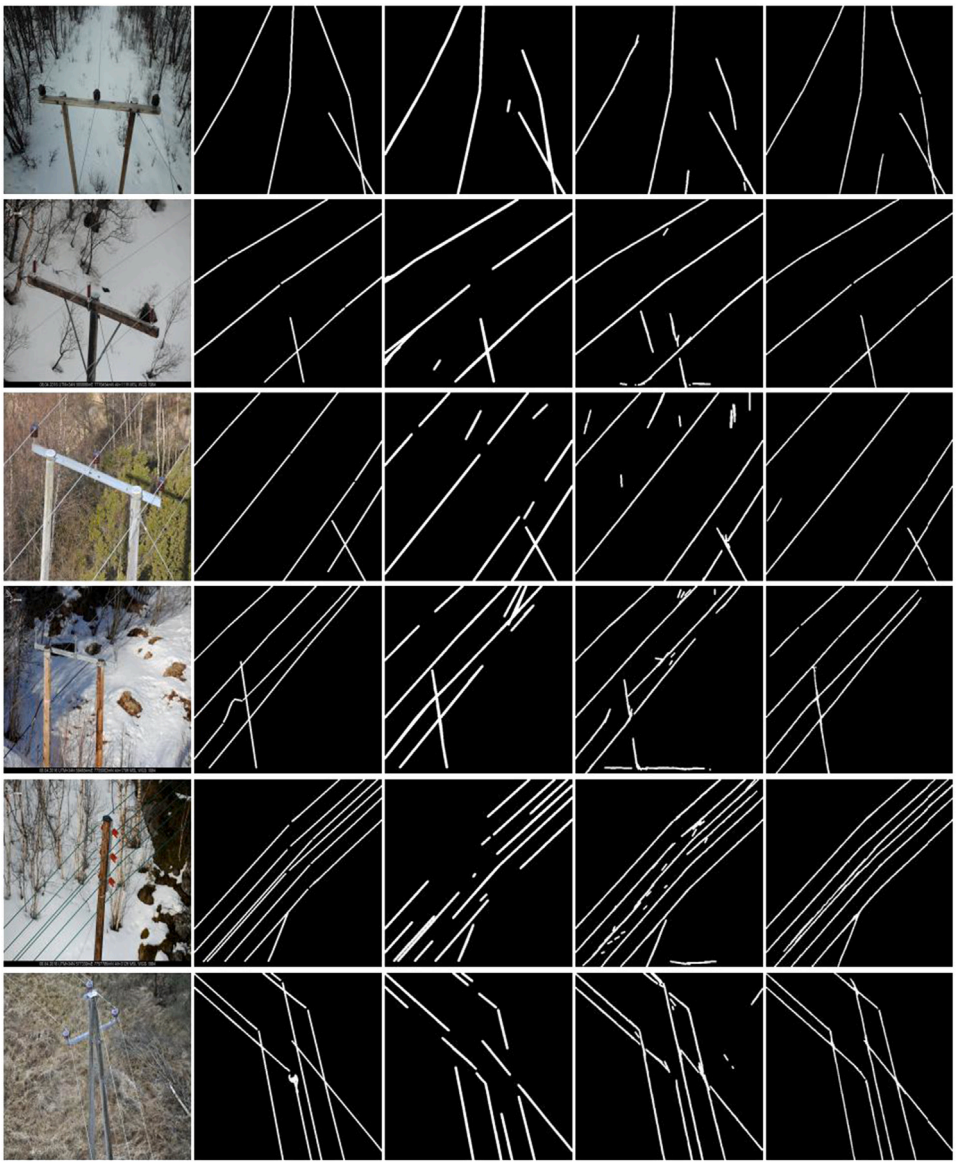


Fig. 7. The comparison of different methods on test images in eSmart dataset.

Table 2
Line widths (lw) and score thresholds (st) at which HAWPv2 achieves the best F_1 and F_β scores across all four datasets.

	PLDU		PLDM		TTPLA		eSmart	
	lw	st	lw	st	lw	st	lw	st
Best F_1	7	0.1	7	0.1	2	0.2	5	0.1
Best F_β	7	0.6	5	0.1	2	0.3	5	0.2

for HAWPv2, we considered two settings, the one that leads to the best F_1 score and the one yielding the best F_β score. Table 2 shows the configuration choices, at which these highest metric values were achieved for each dataset. Overall, the best values are achieved at line widths similar to those used when calculating the metrics for LSNet and LSNetv2. In addition, high score thresholds lead to higher APR and F_β at the expense of ARR and vice versa, which is expected. Overall, LSNetv2 is able to outperform the best F_1 score and F_β score across all datasets. Figs. 6–9 show that HAWPv2 is able to detect entire power lines consistently, however, false positive and false negative regions are often larger than those observed in LSNetv2. This can be attributed to the 4-overlapping-grid design, where missing line segments in each cell can be compensated by its neighbors and, since each cell is only responsible for a small region, false positives occurring in one cell do not have a significant impact. In addition, LSNet model outperforms HAWPv2 in three out of four datasets.

In the following, we briefly present some failure cases of LSNetv2. Besides common cases of false positive and false negative line detection. There are some interesting phenomenons as shown in Fig. 10. The first row shows that both LSNet and LSNetv2 predict a false positive line (constituted by various line segments) which is caused by the edge of the road. For LSNetv2, this false positive line cuts off the continuity of a nearby true positive line and extends itself using a part of that true positive line. Cases like this one indicate that LSNetv2 implicitly enforces the line segments forming continuous lines. However, LSNetv2 might be wrong in its assumption and output semantically incorrect continuous lines as power lines. Another failure case is presented in the second row. As mentioned previously, the TTPLA dataset does not involve guy wires as detection targets, and thus the trained LSNetv2 should exclude these wires. However, as can be observed in this example LSNetv2 detects them partially as positives. LSNetv2 detects guy wires often in the test set demonstrating that it struggles to differentiate between normal power lines and guy wires. This could potentially be

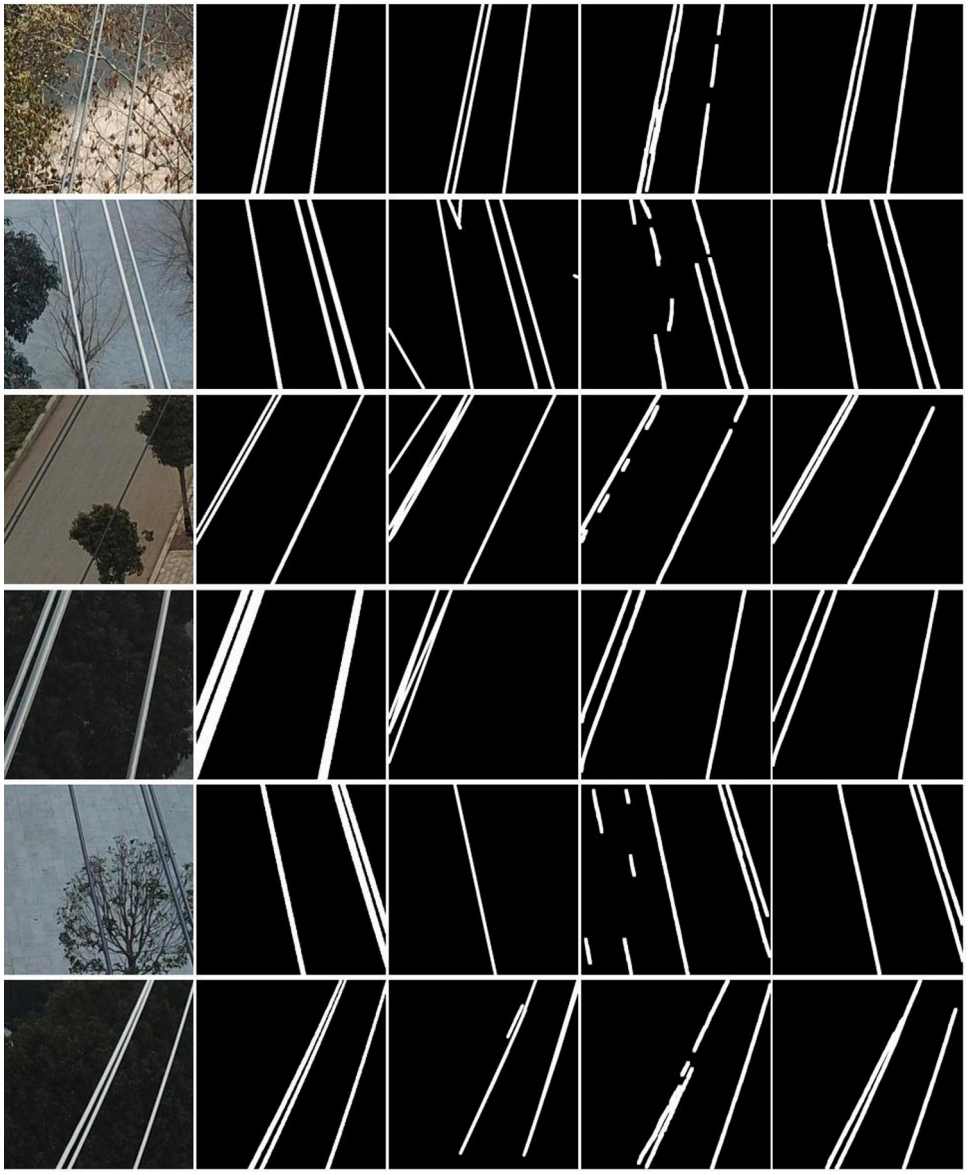


Fig. 8. The comparison of different methods on test images in PLDU dataset.

due to the input images being cropped, providing a limited view of the overall line structure. Further, while LSNetv2 generally works well for images with power lines in close proximity, it can still struggle with very thin lines that run closely in parallel, as illustrated in the example in row three. Finally, the overall performance of LSNetv2 is affected by outliers such as the image in row four. This image is captured with a bottom-up perspective, which is different from the rest of the training dataset.

4.6. Ablation study

We conduct ablation experiments on the TTPLA and eSmart datasets and present results in Table 3 and 4, respectively. We can see the following overall trend for both datasets: removing any of the three proposed components of LSNetv2 results in a degradation in terms of the F_β score, highlighting their importance to the overall model. Furthermore, we note that adding any of the components to the original LSNet (top row) improves the LSNet performance. Finally, the combination of the three achieves the most desirable performance, providing a considerable boost in both F_1 and F_β scores.

Table 3
Results from the ablation study performed on the TTPLA dataset. The study investigates the effect of the proposed components individually and in combination with each other.

ConvNeXt	Multi-guess	Ordered loss	APR	ARR	F_1	F_β
			0.603	0.618	0.610	0.606
✓			0.624	0.660	0.641	0.631
	✓		0.671	0.550	0.604	0.638
		✓	0.673	0.487	0.565	0.619
✓	✓		0.683	0.561	0.616	0.650
✓		✓	0.736	0.495	0.592	0.661
	✓	✓	0.696	0.566	0.624	0.660
✓	✓	✓	0.714	0.560	0.628	0.672

For the eSmart dataset, ConvNeXt is clearly a beneficial addition that gives LSNetv2 improvements in both F_1 and F_β scores. The benefits are also noticeable when concerning the addition of the multi-guess capability brought about by bipartite matching and the ordered loss. These two components individually improved the F_β score while maintaining a comparable F_1 score. Combining the ConvNeXt-Tiny backbone with either the multi-guess capability or the ordered loss, instead, achieves a noticeable performance increase when compared to just

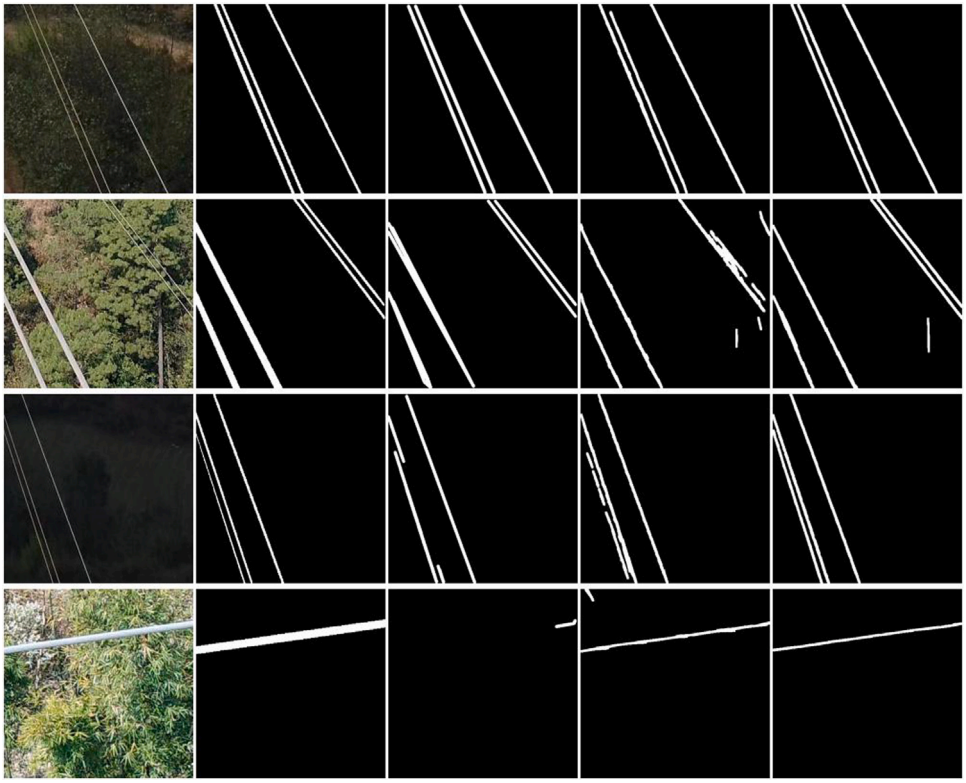


Fig. 9. The comparison of different methods on test images in PLDM dataset.

Table 4
Results from the ablation study performed on the eSmart dataset. The study looks into the effect of the proposed components individually and in combination with each other.

ConvNeXt	Multi-guess	Ordered loss	APR	ARR	F_1	F_β
			0.726	0.812	0.766	0.744
✓			0.844	0.770	0.806	0.825
	✓		0.743	0.787	0.764	0.752
		✓	0.769	0.749	0.759	0.764
✓	✓		0.825	0.820	0.823	0.824
✓		✓	0.839	0.765	0.800	0.820
	✓	✓	0.759	0.813	0.785	0.770
✓	✓	✓	0.845	0.814	0.829	0.837

using the multi-guess capability or the ordered loss alone. This further validates the benefit of having ConvNeXt-Tiny with its increased receptive field as the new backbone. For the TTPLA dataset, we observe that the addition of each of the components individually provides noticeable improvements in F_β scores while maintaining comparable F_1 scores. In particular, the addition of the ConvNeXt-Tiny backbone provides a significant performance boost, indicated in the jumps in both F_1 and F_β score. Pairings ConvNeXt-Tiny with one of the other contributions provides models with higher APR and F_β but with lowered ARR and F_1 when compared to just using ConvNeXt-Tiny alone. This is acceptable since the F_β metric is more preferable as explained in 4.3 and the ARR values are still comparable to the base case where no components are added. In addition, this is understandable since the bipartite matching task provides LSNetv2 with the ability to detect multi line segments per cell but might pose a harder challenge for the model to train. Also, the ordered loss is observed to be particularly relevant when paired with the harder training task brought about by bipartite matching. In fact, for both datasets, the combinations of bipartite matching and ordered loss generated models with better performance than if the two components are used alone.

Fig. 11 also supports these observations. Across different line widths, LSNetv2, with the combination of all three components, consistently achieves superior performance, especially on the eSmart dataset. An addition of either of the three proposed components alone is enough to provide the model with consistent performance improvements, with ConvNeXt having the most impact. Further combination of any two of the proposed components leads to additional improvements.

Figs. 12 and 13 show the example outputs of the model variants in the ablation study. In general, the detection outputs of LSNet are susceptible to random false positives and lines in the background that are not power lines (e.g. road markings). Naturally, LSNet also suffers in cases of line crossings and close proximity. With the ConvNeXt backbone and its increased receptive fields, the model is less affected by semantically incorrect line segments and general false positives. The outputs from LSNet with ConvNeXt backbone are also visually 'cleaner' where the detected neighboring line segments constituting the same power lines are in stronger agreement leading to smoother detected overall power lines. However, leveraging ConvNeXt does not relieve the shortcomings in cases of line crossings and lines that are in close proximity, leading to missing detection and disrupted power line continuity. It can be seen that the addition of the multi-guess capability effectively tackles this problem. Although less apparent, we observed that the inclusion of the ordered loss alone helps LSNet remedy the confusion problem taking place between parallel power lines that are closely spaced. By using the ordered loss, there are fewer false positives that bridge between these power lines and better separation of detected power lines can be achieved.

4.7. Backbone analysis

To further validate the benefit of the ConvNeXt-Tiny backbone, we compare ConvNeXt-Tiny with the original modified VGG-16 architecture, as well as a Resnet-50 and Efficientnetv2-M backbone. The detailed architecture is shown in Table 5.

Table 6 shows the results of our experiment with other backbones. In general, the ConvNeXt-Tiny backbone is superior to the other alternatives, helping LSNetv2 achieve the highest F_1 and F_β scores across

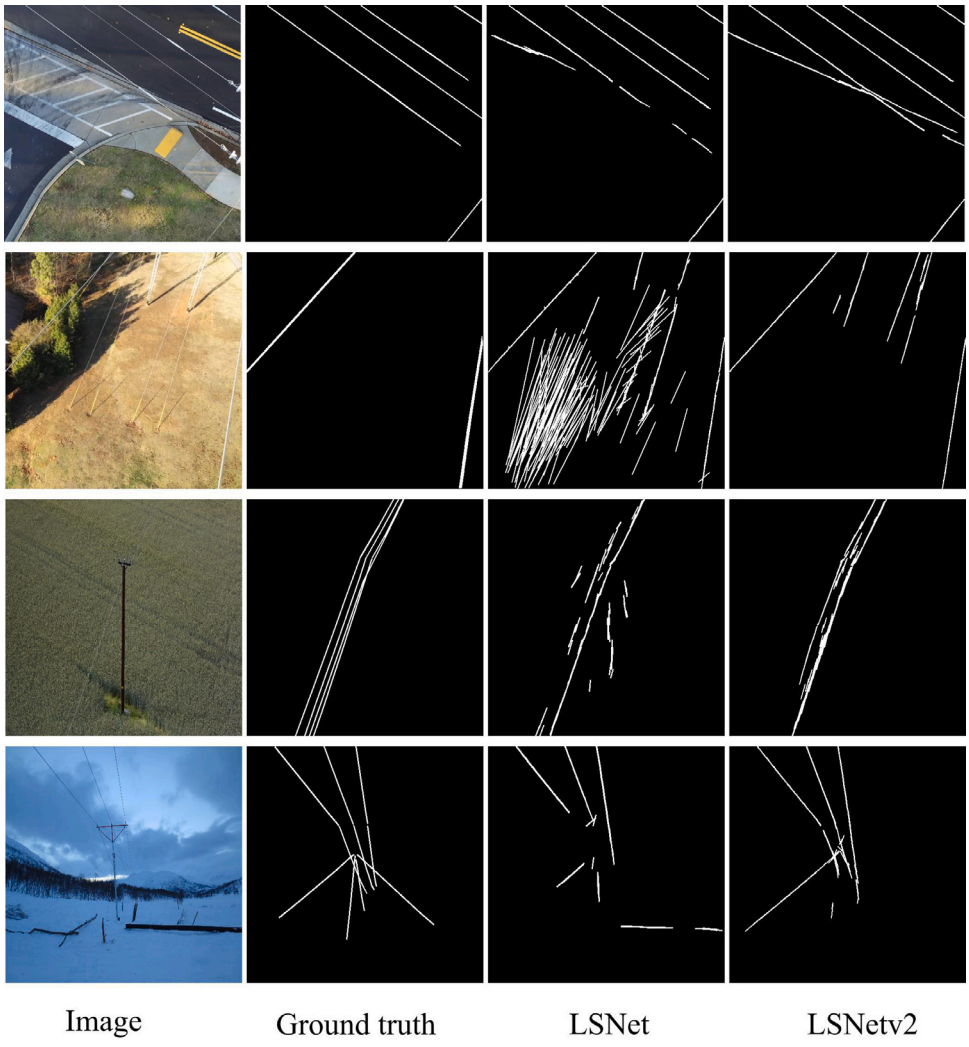


Fig. 10. Example failed outputs of LSNetv2. The first two rows show images from the TTPLA dataset. The remaining two rows show images from the eSmart dataset.

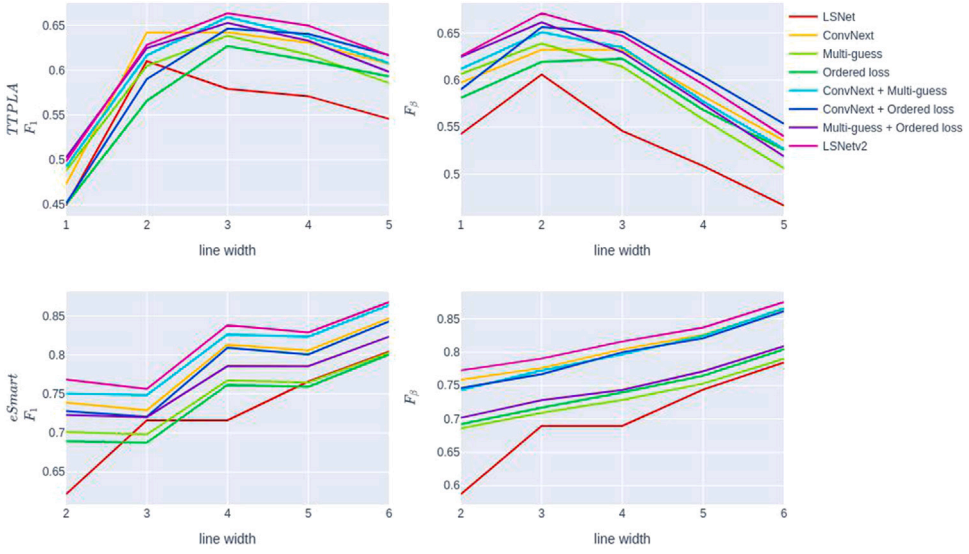


Fig. 11. Line plots of F_1 and F_β scores obtained from the ablation study across multiple line pixel widths.

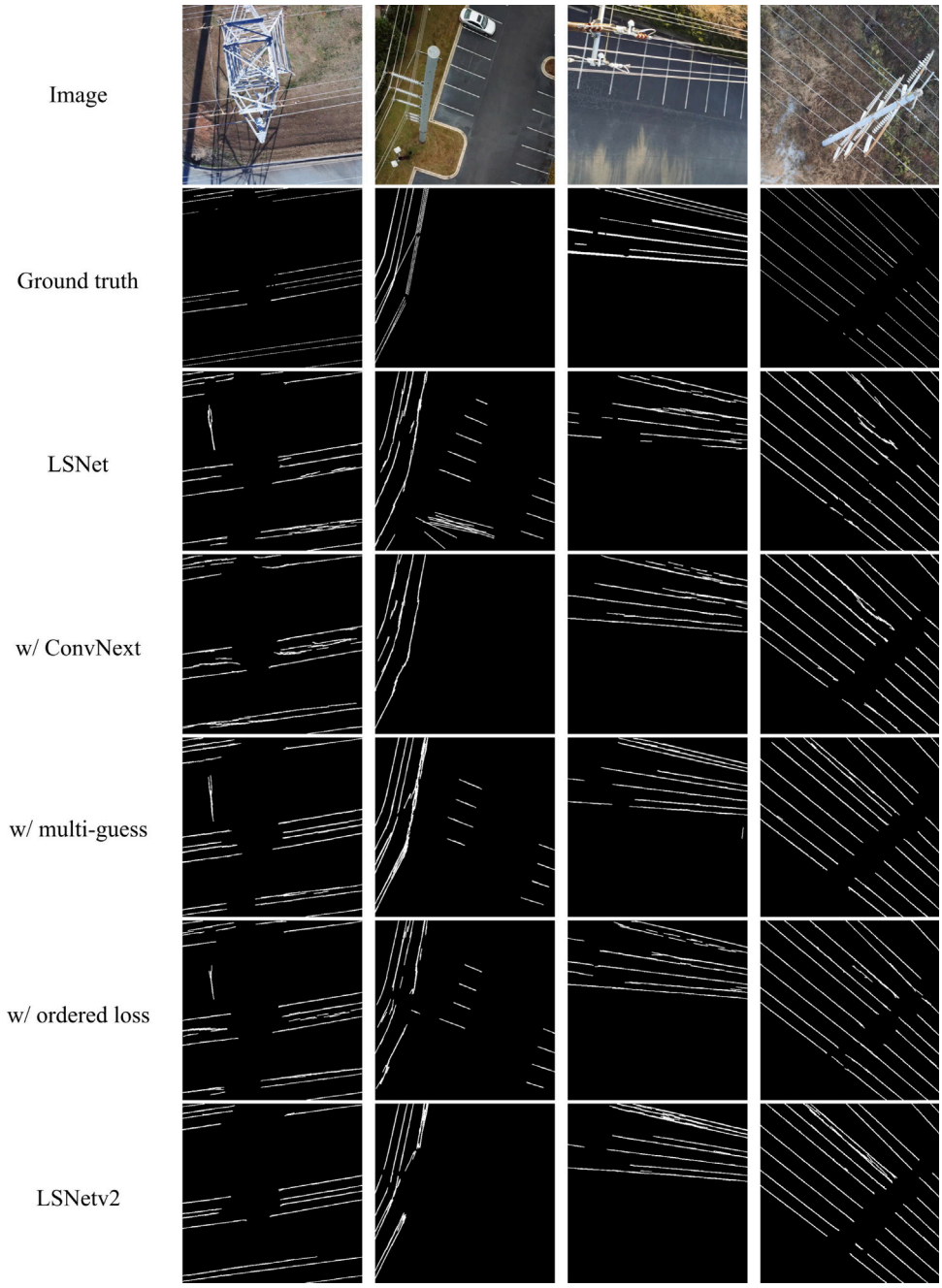


Fig. 12. Example outputs on the TTPLA dataset of LSNet, LSNetv2, and other variants in the ablation study.

almost all scenarios. In addition, with ConvNeXt-Tiny being in one of the state-of-the-art families of CNNs, the performance of this backbone can be attributed to it having a big theoretical receptive field, more than double the size of the input image. This means that each cell inference produced by this network can hypothetically gain information from the entire image. LSNetv2 with the modified VGG-16 backbone has competitive performance versus the first version of LSNet for all datasets except for PLDM, where LSNetv2 has a comparable F_1 score but noticeably lower APR and F_β score. This can be attributed to the limited need for multi-line-segment-per-cell detection when considering the PLDU and PLDM datasets. Additionally, in these scenarios, the introduction of bipartite matching might result in more challenging optimization due to the additional degree of freedom when training LSNetv2. Nevertheless, for the more challenging TTPLA and eSmart datasets, which benefit from the multi-line-segment detection capability, LSNetv2 with a modified-VGG backbone achieves considerably better performance than the original LSNet. LSNetv2 with Resnet-50 backbone is consistently the second-best across all cases despite having a relatively smaller theoretical receptive field and the lowest number of trainable parameters. This is expected since Resnet is the direct predecessor of ConvNeXt and both are equipped with skip connections, which have been demonstrated to smoothen the optimization landscape (Li, Xu, Taylor, Studer, & Goldstein, 2018). The VGG-16 model, on the other hand, does not have these features, making it harder to optimize. Efficientnetv2-M is another relatively recent architecture that is competitive at image classification (Tan & Le, 2021) and object detection (Tan, Pang, & Le, 2019). It also has a high theoretical receptive field (technically covers the entire image due to the squeeze-and-excitation blocks) as well as skip connections. However, this backbone is consistently inferior to the rest, including the original modified-VGG backbone. This might be because EfficientnetV2 was designed

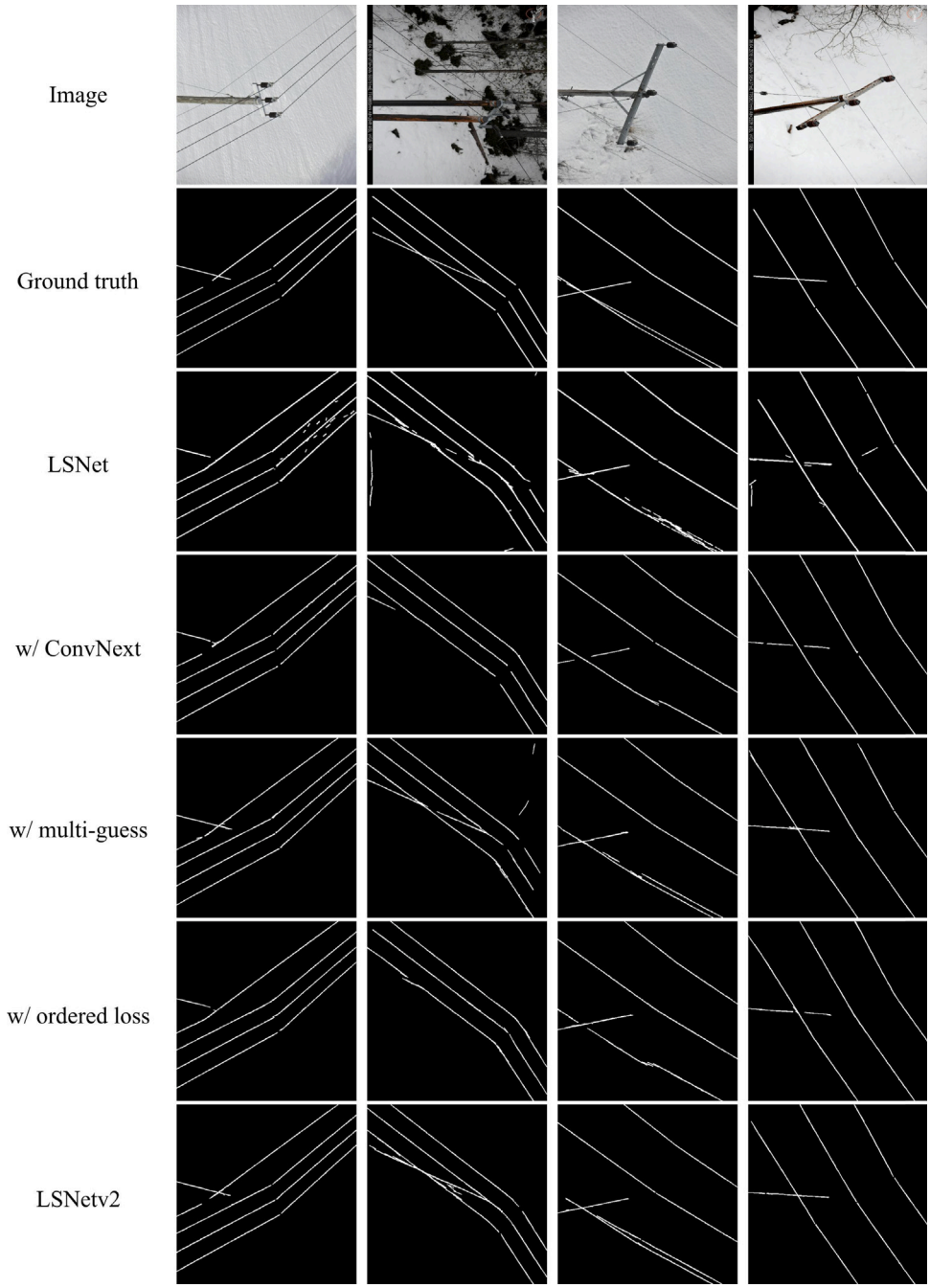


Fig. 13. Example outputs on the eSmart dataset of LSNet, LSNetv2, and other variants in the ablation study.

specifically to achieve high accuracy for image classification and, even though this architecture family also performs well for object detection and semantic segmentation (Tan & Le, 2021), EfficientnetV2 is simply not suitable for the tasks of LSNetv2. Also, since EfficientnetV2 was designed with a high level of specificity via a compound scaling method, therefore, any modification, such as truncation in this case, can have a greater negative effect on performance.

4.8. Robustness analysis

In order to ensure consistent performance of LSNetv2, we performed multiple training trials of LSNet and LSNetv2 and observed the variations in the metrics. The result is shown in Table 7. As can be seen, all the scores indicate that LSNetv2 is robust with small standard deviation in the performance. LSNet is also robust for the eSmart dataset but its metrics vary more across runs. T-tests illustrate that the improvements of LSNetv2 are statistically significant for most metrics ($p< 0.05$). In addition, we can see that, through T-tests, LSNetv2 is assuredly better than HAWPv2 in terms of the F_1 scores. For the eSmart dataset, F_β scores also ascertain the performance gap between LSNetv2 and HAWPv2.

4.9. Computational analysis

The increased capability of LSNetv2 comes at the cost of efficiency. LSNetv2 has a lower amount of FLOPS (floating point operations) (41.9 GFLOPs vs. 130.8 GFLOPs) due to the use of depthwise and 1×1 convolutions. However it has slightly more parameters (13.9 million vs. 10.0 million), and, in reality, a higher inference time (0.17 s vs. 0.10 s on an RTX 3090). Future work should investigate directions to

Table 5
The architecture of LSNetv2 using different CNNs as the backbone. $a \times a, b, s = c$ indicates a convolutional layer with kernel size a, number of filters b and stride c. *dw* indicates a depthwise convolutional layer. The curl brackets signal groups of layers with skip connections. This table excludes more detailed information such as normalization and activation layers. The receptive fields are calculated by `receptive_field_analysis_toolbox` (Richter, 2021).

Stage	Output size	Modified VGG	Resnet-50	EfficientnetV2-M	ConvNeXt-Tiny
1	256×256	$3 \times 3, 64$ $3 \times 3, 64$ $1 \times 3, 64, , s{=}2$	7×7, s=2	$3 \times 3, 24$, s=2 $3\times \{ 3 \times 3, 24$	
2	128×128	$3 \times 3, 128$ $3 \times 3, 128$ $1 \times 3, 128, , s{=}2$	3×3 maxpool, s=2 $3\times \{ 1 \times 1, 64$ $3 \times 3, 64$ $1 \times 3, 256$	$3 \times 3, 96$, s=2 $1 \times 1, 48$ $4\times \{ 3 \times 3, 192$ $1 \times 1, 48$	$4 \times 4, 96$, s=4
3	64×64	$3 \times 3, 256$ $3 \times 3, 256$ $1 \times 3, 256, , s{=}2$	$1\times \{ 1 \times 1, 128$, s=2 $3 \times 3, 128$ $1 \times 3, 512$ $3\times \{ 1 \times 1, 128$ $3 \times 3, 128$ $1 \times 3, 512$	$3 \times 3, 192$, s=2 $1 \times 1, 80$ $4\times \{ 3 \times 3, 320$ $1 \times 1, 80$	$3\times \{$ dw $7 \times 7, 96$ $1 \times 1, 384$ $1 \times 1, 96$ $2 \times 2, 192$, s=2
4	32×32	$3 \times 3, 512$ $3 \times 3, 512$ $1 \times 3, 512, , s{=}2$	$1\times \{ 1 \times 1, 256$, s=2 $3 \times 3, 256$ $1 \times 3, 1024$ $5\times \{ 1 \times 1, 256$ $3 \times 3, 256$ $1 \times 3, 1024$	$1 \times 1, 320$ dw $3 \times 3, 320, s = 2$ SE-Block(ratio=0.25) $1 \times 1, 160$ $6\times \{ 1 \times 1, 640$ dw $3 \times 3, 640$ SE-Block(ratio=0.25) $1 \times 1, 160$	$3\times \{$ dw $7 \times 7, 192$ $1 \times 1, 768$ $1 \times 1, 192$ $2 \times 2, 384$, s=2
5	32×32			$1 \times 1, 960$ dw $3 \times 3, 960$ SE-Block(ratio=0.25) $1 \times 1, 176$ $13\times \{ 1 \times 1, 1056$ dw $3 \times 3, 1056$ SE-Block(ratio=0.25) $1 \times 1, 176$ $1 \times 1, 1056$	$9\times \{$ dw $7 \times 7, 384$ $1 \times 1, 1536$ $1 \times 1, 384$
Classifier or Regressor module	31×31	$2 \times 2, 512$ $1 \times 1, 2$ * 10 or 4 * 10			
Receptive field		91	291	779 (inf)	1096
Parameters (M)		10.0	12.8	14.5	13.9

Table 6
Performance of LSNetv2 using different CNNs as the backbone.

		Modified VGG	Resnet-50	Efficientnetv2-M	ConvNeXt-Tiny
PLDU	APR	0.899	0.920	0.907	0.938
	ARR	0.685	0.673	0.645	0.666
	F_1	0.778	0.778	0.764	0.779
	F_β	0.838	0.848	0.829	0.857
PLDM	APR	0.818	0.918	0.923	0.934
	ARR	0.793	0.731	0.674	0.724
	F_1	0.805	0.814	0.779	0.815
	F_β	0.812	0.866	0.850	0.875
TTPLA	APR	0.696	0.643	0.573	0.714
	ARR	0.566	0.635	0.355	0.560
	F_1	0.624	0.639	0.438	0.628
	F_β	0.660	0.641	0.501	0.671
eSmart	APR	0.759	0.794	0.747	0.845
	ARR	0.813	0.809	0.796	0.814
	F_1	0.785	0.802	0.771	0.829
	F_β	0.770	0.797	0.757	0.837

Table 7
Table showing the robustness analysis of LSNet and LSNetv2 across 5 runs. The superscript symbols $*$, $+$, and $-$ indicates that the performance gaps of pairs (LSNetv2, LSNet), (LSNetv2, HAWPv2 at best F_1 score), and (LSNetv2, HAWPv2 at best F_β score) are statistically significant ($p < 0.05$).

		HAWPv2 (F_1)	HAWPv2 (F_β)	LSNet	LSNetv2
TTPLA	APR	0.677 ± 0.059	0.682 ± 0.049	0.607 ± 0.021	$0.703 \pm 0.008^{*}$
	ARR	0.426 ± 0.107	0.411 ± 0.078	0.593 ± 0.036	$0.566 \pm 0.011^{+-}$
	F_1	0.509 ± 0.045	0.505 ± 0.038	0.600 ± 0.024	$0.626 \pm 0.004^{*+-}$
	F_β	0.641 ± 0.035	0.642 ± 0.032	0.603 ± 0.021	$0.665 \pm 0.004^{*}$
eSmart	APR	0.823 ± 0.036	0.828 ± 0.027	0.713 ± 0.008	$0.844 \pm 0.004^{*+-}$
	ARR	0.654 ± 0.098	0.654 ± 0.098	0.813 ± 0.003	$0.818 \pm 0.002^{*}$
	F_1	0.722 ± 0.038	0.721 ± 0.035	0.760 ± 0.005	$0.831 \pm 0.002^{*+-}$
	F_β	0.806 ± 0.017	0.809 ± 0.013	0.734 ± 0.007	$0.837 \pm 0.003^{*+-}$

improve the complexity of LSNetv2 while maintaining its performance to support real-time applications. Regarding HAWPv2, it possesses an amount of parameters of about 11.1 million, 121.5 GFLOPs and an average inference time without post-processing of 0.03 s.

5. Future works

LSNetv2 opens up new possibilities in efficient monitoring of power lines and there are several directions that can be explored to further improve on this. These directions can be categorized into two main directions, one focusing on improving the detection capabilities of LSNetv2 by improving algorithmic aspects, while the other aims to directly incorporate and leverage UAV properties. These are two orthogonal directions, and we will briefly reflect on potential directions within both.

5.1. Algorithmic aspects

Future work on LSNetv2 should focus on exploring new directions to further improve its capabilities. One area of interest for future research is the multi-guess capability of LSNetv2, which has shown significant improvement over the predecessor model, LSNet. However, this feature potentially further exacerbates the imbalance problem between the number of positive and negative labels already present in LSNet. Both LSNet and LSNetv2 leverage the Focal loss to partly mitigate this problem, however, future work on losses and sampling techniques is needed to further address the issue and enhance the performance of LSNetv2.

Another promising direction is to further model the width of the power lines. While it is sufficient for most downstream inspection tasks to only obtain the trajectories of the power lines, it could be beneficial if width information across the power lines could be derived from LSNetv2 as well. For example, changes in the power line circumference could help detect bird caging defects and joint components. Thus, future work should look into ways to extract this width information while maintaining the use of polyline annotations.

Furthermore, future work should also involve an investigation of how to better model curvature in power lines. While LSNet and LSNetv2 can somewhat account for curves via detecting power lines in a piecewise manner, they are still limited for severely curved power lines when the assumption of straight power line segments in a cell is less applicable.

The ablation study has validated the contribution of each proposed components that advance LSNet into LSNetv2. However, the improvements can be further emphasized if they can be shown through more intuitive means such as GradCAM heatmaps. Hence, it can be a good idea to investigate into the discovery and application of interpretability methods on LSNetv2 to gain more insights into the inner workings of the model and identify possible future work directions.

Finally, another promising direction of future work is the study of domain adaptation in the context of LSNetv2 to generalize its performance to changing conditions. This would facilitate the use of LSNetv2 in diverse real-world scenarios, ensuring a better foundation for further downstream inspection tasks.

5.2. Leverage UAV properties

While we in this work have focused on algorithmic development, numerous hardware constraints need to be considered when working with UAVs. Hence, a promising future direction is to explicitly incorporate the UAV design into the model to tailor it to specific designs.

First is the choice of imaging sensor. Given the ease of implementation and applicability in most types of inspections, images by conventional visual light cameras have been used in this work and are currently the most popular approach (Abdelfattah et al., 2020; Martinez, Sampedro, Chauhan, Collumeau, & Campoy, 2018; Wang, Gao, Xu, & Li, 2022; Yang et al., 2020). However, LSNetv2 could potentially be improved by including other modalities such as LiDAR (Chen et al., 2017), ultraviolet (Zhao & Guo, 2019), infrared (Zhang et al., 2016) and/or thermal cameras (Demkiv, Ruffo, Silano, Bednar, & Saska, 2021), which have achieved promising results in detecting specific defects such as overheating or corona discharge.

Further, there are two types of UAVs commonly used for power line inspection: fixed-wing and rotary-wing UAVs. Each has its own characteristics, strengths, and weaknesses making it suitable for specific inspection scenarios (Nekovář, Faigl, & Saska, 2021; Wang et al., 2022; Xu, Zhao, Wang, & Chen, 2023b). In addition, they are inevitably affected by challenges such as innate vibrations or external disturbances, which can reduce the quality of the camera observations greatly (with motion blur, inconsistent angle, etc.) (Gurtner et al., 2009). These problems are attributed to different causes (Li et al., 2017; Ma & Wu, 2012; Verbeke & Debruyne, 2016) and, thus, need to be accounted for in different manners (Mizui, Yamamoto, & Ohsawa, 2012; Rodin, 2019). Currently, LSNetv2 does not take these challenges into consideration. Thus, future work should look at either finding efficient procedures to account for these problems and the UAV design shift, or making LSNetv2 robust against these variables, and thus widely applicable to many inspection systems configurations.

6. Conclusion

We presented LSNetv2, an improved version of LSNet with the capability to detect multiple line segments per divided grid cell, thus enabling it to detect line-crossing as well as power lines that are in close proximity to each other. This capability is brought about by using multiple outputs trained with bipartite matching. Furthermore, we proposed a new regression loss, where the order of the detected endpoints is fixed, to remove interdependencies between the predicted endpoints and thus improve the performance, especially when in conjunction with the multi-guess capability. In this new loss, the endpoint with a smaller x-coordinate is always inferred by the same element pair in the output matrix, and the remaining endpoint is inferred with the same remaining element pair. In addition, we updated the backbone to the state-of-the-art family of ConvNeXt, which inherits well-proven designed elements from previous state-of-the-art approaches, in order to increase the overall receptive field of the model. We empirically demonstrate that this leads to an overall increase in performance achieving F_β scores of over of 0.857, 0.875, and 0.671 on the public datasets PLDU, PLDM and TTPLA, respectively, while using only modified weak polyline annotation. Overall, LSNetv2 consistently outperforms its predecessor, LSNet, and its competitor, HAWPv2, across all datasets evaluated.

CRediT authorship contribution statement

Duy Khoi Tran: Conceptualization, Software, Methodology, Formal analysis, Visualization, Writing – review & editing, Writing – original draft. **Van Nhan Nguyen:** Conceptualization, Methodology, Supervision, Writing – review & editing, Writing – original draft. **Davide Roverso:** Conceptualization, Methodology, Supervision, Writing – review & editing, Writing – original draft. **Robert Jenssen:** Conceptualization, Methodology, Supervision, Writing – review & editing, Writing – original draft. **Michael Kampffmeyer:** Conceptualization, Methodology, Supervision, Writing – review & editing, Writing – original draft.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Duy Khoi Tran reports financial support was provided by Research Council of Norway.

Data availability

The method was compared on three openly available datasets and one proprietary dataset.

Acknowledgments

The authors thank the associate editor and the anonymous reviewers for their valuable comments. Their suggestions have improved the final version of this paper and pointed to promising directions for future work. For this, we are grateful.

This work was supported by the Research Council of Norway under grant no. 321261.

References

Abdelfattah, R., Wang, X., & Wang, S. (2020). TTPLA: An aerial-image dataset for detection and segmentation of transmission towers and power lines. In *Proceedings of the Asian conference on computer vision*.

Abdelfattah, R., Wang, X., & Wang, S. (2023). PLGAN: Generative adversarial networks for power-line segmentation in aerial images. *PP*, In *IEEE Transactions on Image Processing*. http://dx.doi.org/10.1109/TIP.2023.3321465.

Achanta, R., Hemami, S. S., Estrada, F. J., & Süsstrunk, S. (2009). Frequency-tuned salient region detection. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 1597–1604).

Alpatov, B., Babayan, P., & Shubin, N. (2016). Power line detection using Integrated Vector Radon Transform. In *2016 5th Mediterranean conference on embedded computing* (pp. 162–164). http://dx.doi.org/10.1109/MECO.2016.7525729.

Antwi-Bekoe, E., Zhan, Q., Xie, X., & Liu, G. (2020). Insulator recognition and fault detection using deep learning approach. *Journal of Physics: Conference Series*, *1454*(1), Article 012011. http://dx.doi.org/10.1088/1742-6596/1454/1/012011.

Bojarski, M., Choromanska, A., Choromanski, K., Firner, B., Ackel, L. J., Muller, U., et al. (2018). VisualBackProp: Efficient visualization of CNNs for autonomous driving. In *2018 IEEE international conference on robotics and automation* (pp. 4701–4708). http://dx.doi.org/10.1109/ICRA.2018.8461053.

Candamo, J., Kasturi, R., Goldgof, D., & Sarkar, S. (2009). Detection of thin lines using low-quality video from low-altitude aircraft in urban settings. *IEEE Transactions on Aerospace and Electronic Systems*, *45*(3), 937–949. http://dx.doi.org/10.1109/TAES.2009.5259175.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European Conference on Computer Vision* (pp. 213–229). http://dx.doi.org/10.1007/978-3-030-58452-8_13.

Chen, C., Peng, X., Song, S., Wang, K., Qian, J., & Yang, B. (2017). Safety distance diagnosis of large scale transmission line corridor inspection based on LiDAR point cloud collected with UAV. *Dianwang Jishu/Power System Technology*, *41*, 2723–2730. http://dx.doi.org/10.13335/j.1000-3673.pst.2016.3194.

Cheng, M.-M., Zhang, G.-X., Mitra, N. J., Huang, X., & Hu, S. (2011). Global contrast based salient region detection. In *CVPR 2011* (pp. 409–416).

Choi, H., Koo, G., Kim, B. J., & Kim, S. W. (2021). Weakly supervised power line detection algorithm using a recursive noisy label update with refined broken line segments. *Expert Systems with Applications*, *165*, Article 113895. http://dx.doi.org/10.1016/j.eswa.2020.113895, URL: https://www.sciencedirect.com/science/article/pii/S0957417420306953.

Crouse, D. F. (2016). On implementing 2D rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, *52*(4), 1679–1696. http://dx.doi.org/10.1109/TAES.2016.140952.

Demkiv, L., Ruffo, M., Silano, G., Bednar, J., & Saska, M. (2021). An application of stereo thermal vision for preliminary inspection of electrical power lines by MAVs. In *2021 aerial robotic systems physically interacting with the environment* (pp. 1–8). http://dx.doi.org/10.1109/AIRPHARO52252.2021.9571025.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248–255). Ieee.

Deng, C., Wang, S., Huang, Z., Tan, Z., & Liu, J. (2014). Unmanned aerial vehicles for power line inspection: A cooperative way in platforms and communications. *Journal of Communication*, *9*, 687–692.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Douglas, D. H., & Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, *10*, 112–122.

Feng, Z., Kittler, J., Awais, M., Huber, P., & Wu, X. (2017). Wing loss for robust facial landmark localisation with convolutional neural networks. In *2018 IEEE/CVF conference on computer vision and pattern recognition* (pp. 2235–2245).

Grompone von Gioi, R., Jakubowicz, J., Morel, J.-M., & Randall, G. (2010). LSD: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(4), 722–732. http://dx.doi.org/10.1109/TPAMI.2008.300.

Girshick, R. B. (2015). Fast R-CNN. In *2015 IEEE international conference on computer vision* (pp. 1440–1448).

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In Y. W. Teh, & M. Titterington (Eds.), *Proceedings of machine learning research*: *vol. 9*, *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249–256). Chia Laguna Resort, Sardinia, Italy: PMLR, URL: https://proceedings.mlr.press/v9/glorot10a.html.

Golightly, I., & Jones, D. (2005). Visual control of an unmanned aerial vehicle for power line inspection. In *ICAR '05. proceedings., 12th international conference on advanced robotics, 2005* (pp. 288–295). http://dx.doi.org/10.1109/ICAR.2005.1507426.

Gubbi, J., Varghese, A., & Balamuralidhar, P. (2017). A new deep learning architecture for detection of long linear infrastructure. In *2017 fifteenth IAPR international conference on machine vision applications* (pp. 207–210). http://dx.doi.org/10.23919/MVA.2017.7986837.

Gurtner, A., Greer, D. G., Glassock, R., Mejias, L., Walker, R. A., & Boles, W. W. (2009). Investigation of fish-eye lenses for small-UAV aerial photography. *IEEE Transactions on Geoscience and Remote Sensing*, *47*(3), 709–721. http://dx.doi.org/10.1109/TGRS.2008.2009763.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. B. (2017). Mask R-CNN. In *2017 IEEE international conference on computer vision* (pp. 2980–2988).

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. In *2016 IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Huang, L. (2021). Skeleton tracing. GitHub repository, GitHub, https://github.com/LingDong-/skeleton-tracing.

Jaffari, R., Hashmani, M. A., & Reyes-Aldasoro, C. C. (2021). A novel focal phi loss for power line segmentation with auxiliary classifier U-Net. *Sensors*, *21*(8), http://dx.doi.org/10.3390/s21082803, URL: https://www.mdpi.com/1424-8220/21/8/2803.

Kasturi, R., & Camps, O. I. (2002). Wire detection algorithms for navigation.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Y. Bengio, & Y. LeCun (Eds.), *3rd international conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, conference track proceedings*. URL: http://arxiv.org/abs/1412.6980.

Lee, S. J., Yun, J. P., Choi, H., Kwon, W., Koo, G., & Kim, S. W. (2017). Weakly supervised learning with convolutional neural networks for power line localization. In *2017 IEEE symposium series on computational intelligence* (pp. 1–8). http://dx.doi.org/10.1109/SSCI.2017.8285410.

Li, Z., Lao, M., Phang, S. K., Hamid, M. R. A., Tang, K. Z., & Lin, F. (2017). Development and design methodology of an anti-vibration system on micro-UAVs. In *International micro air vehicle conference and flight competition*.

Li, Z., Liu, Y., Hayward, R., Zhang, J., & Cai, J. (2008). Knowledge-based power line detection for UAV surveillance and inspection systems. In *2008 23rd international conference image and vision computing New Zealand* (pp. 1–6). http://dx.doi.org/10.1109/IVCNZ.2008.4762118.

Li, Z., Liu, Y., Walker, R., Hayward, R., & Zhang, J. (2010). Towards automatic power line detection for a UAV surveillance system using pulse coupled neural filter and an improved hough transform. *Machine Vision and Applications*, *21*(5), 677–686. http://dx.doi.org/10.1007/s00138-009-0206-y.

Li, Y., Xiao, Z., Zhen, X., & Cao, X. (2019). Attentional information fusion networks for cross-scene power line detection. *IEEE Geoscience and Remote Sensing Letters*, *16*(10), 1635–1639. http://dx.doi.org/10.1109/LGRS.2019.2903217.

Li, H., Xu, Z., Taylor, G., Studer, C., & Goldstein, T. (2018). Visualizing the loss landscape of neural nets. *Advances in Neural Information Processing Systems*, *31*.

Lin, T.-Y., Goyal, P., Girshick, R. B., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *2017 IEEE international conference on computer vision* (pp. 2999–3007).

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E., Fu, C.-Y., et al. (2015). SSD: Single shot MultiBox detector. In *European conference on computer vision*.

Liu, Y., Lai, T., Liu, J., Li, Y., Pei, S., & Yang, J. (2021). Insulator contamination diagnosis method based on deep learning convolutional neural network. In *2021 3rd Asia energy and electrical engineering symposium* (pp. 184–188). http://dx.doi.org/10.1109/AEEES51875.2021.9402970.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF international conference on computer vision* (pp. 9992–10002).

Liu, Z., Mao, H., Wu, C., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A ConvNet for the 2020s. In *2022 IEEE/CVF conference on computer vision and pattern recognition* (pp. 11966–11976).

Ma, H., & Wu, J. (2012). Analysis of positioning errors caused by platform vibration of airborne LiDAR system. In *2012 8th IEEE international symposium on instrumentation and control technology (ISICT) proceedings* (pp. 257–261). http://dx.doi.org/10.1109/ISICT.2012.6291650.

Madaan, R., Maturana, D., & Scherer, S. (2017). Wire detection using synthetic data and dilated convolutional networks for unmanned aerial vehicles. In *2017 IEEE/RSJ international conference on intelligent robots and systems* (pp. 3487–3494). http://dx.doi.org/10.1109/IROS.2017.8206190.

Major, J., Alvarez, J., Franke, E., Light, G., Allen, P., & Edwards, S. (2008). *Future inspection of overhead transmission lines: Technical report 1016921*, Electric Power Research Institute, [Online]. Available: https://www.epri.com/research/products/1016921.

Major, J., et al. (2011). *Emerging and future inspection of overhead transmission lines: Technical report 1021876*, Electric Power Research Institute, [Online]. Available: https://www.epri.com/research/products/000000000001021876.

Martinez, C., Sampedro, C., Chauhan, A., Collumeau, J. F., & Campoy, P. (2018). The Power Line Inspection Software (PoLIS): A versatile system for automating power line inspection. *Engineering Applications of Artificial Intelligence*, *71*, 293–314. http://dx.doi.org/10.1016/j.engappai.2018.02.008, URL: https://www.sciencedirect.com/science/article/pii/S0952197618300290.

Mizui, M., Yamamoto, I., & Ohsawa, R. (2012). Effects of propeller-balance on sensors in small-scale unmanned aerial vehicle. *IOSR Journal of Engineering*, *02*, 23–27, URL: https://api.semanticscholar.org/CorpusID:54035792.

Nekovář, F., Faigl, J., & Saska, M. (2021). Multi-tour set traveling salesman problem in planning power transmission line inspection. *IEEE Robotics and Automation Letters*, *6*(4), 6196–6203. http://dx.doi.org/10.1109/LRA.2021.3091695.

Nguyen, V. N., Jenssen, R., & Roverso, D. (2018). Automatic autonomous vision-based power line inspection: A review of current status and the potential role of deep learning. *International Journal of Electrical Power & Energy Systems*.

Nguyen, V. N., Jenssen, R., & Roverso, D. (2020). LS-Net: fast single-shot line-segment detector. *Machine Vision and Applications*, *32*(1), 12. http://dx.doi.org/10.1007/s00138-020-01138-6.

Pan, C., Cao, X., & Wu, D. (2016). Power line detection via background noise removal. In *2016 IEEE global conference on signal and information processing* (pp. 871–875). http://dx.doi.org/10.1109/GlobalSIP.2016.7905967.

Patnaik, S. (2019). Bankrupted by deadly wildfires, PG&E vows to keep the lights on. *Reuters*, URL: https://www.reuters.com/article/us-pg-e-us-bankruptcy/bankrupted-by-deadly-wildfires-pge-vows-to-keep-the-lights-on-idUSKCN1PN0PX.

Redmon, J., Divvala, S. K., Girshick, R. B., & Farhadi, A. (2015). You only look once: Unified, real-time object detection. In *2016 IEEE conference on computer vision and pattern recognition* (pp. 779–788).

Richter, M. (2021). ReceptiveFieldAnalysisToolbox. GitHub repository, GitHub, https://github.com/MLRichter/receptive_field_analysis_toolbox.

Rodin, C. D. (2019). *Applications of high-precision optical imaging systems for small unmanned aerial systems in maritime environments* (Ph.D. thesis), Trondheim, Norway: Norwegian University of Science and Technology, Available at https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2645782.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), *Medical image computing and computer-assisted intervention* (pp. 234–241). Cham: Springer International Publishing.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556.

Snorker, H. (2019). PLD-UAV. GitHub repository, GitHub, https://github.com/SnorkerHeng/PLD-UAV.

Song, B., & Li, X. (2014). Power line detection from optical images. *Neurocomputing*, *129*, 350–361. http://dx.doi.org/10.1016/j.neucom.2013.09.023, URL: https://www.sciencedirect.com/science/article/pii/S0925231213009429.

Steger, C. (1998). An unbiased detector of curvilinear structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(2), 113–125. http://dx.doi.org/10.1109/34.659930.

Tan, M., & Le, Q. V. (2021). EfficientNetV2: Smaller models and faster training. In M. Meila, & T. Zhang (Eds.), *Proceedings of machine learning research: vol. 139, Proceedings of the 38th international conference on machine learning, ICML 2021, 18-24 July 2021, virtual event* (pp. 10096–10106). PMLR, URL: http://proceedings.mlr.press/v139/tan21a.html.

Tan, M., Pang, R., & Le, Q. V. (2019). EfficientDet: Scalable and efficient object detection. In *2020 IEEE/CVF conference on computer vision and pattern recognition* (pp. 10778–10787).

Verbeke, J., & Debruyne, S. (2016). Vibration analysis of a uav multirotor frame. In *Proceedings of ISMA 2016 international conference on noise and vibration engineering* (pp. 2401–2409).

Wang, Z., Gao, Q., Xu, J., & Li, D. (2022). A review of UAV power line inspection. In L. Yan, H. Duan, & X. Yu (Eds.), *Advances in guidance, navigation and control* (pp. 3147–3159). Singapore: Springer Singapore.

Wang, L., Wang, C., Sun, Z., & Chen, S. (2020). An improved dice loss for pneumothorax segmentation by mining the information of negative areas. *IEEE Access*, *8*, 167939–167949. http://dx.doi.org/10.1109/ACCESS.2020.3020475.

Wu, Y., & He, K. (2018). Group normalization. *International Journal of Computer Vision*, *128*, 742–755.

Xu, B., Zhao, Y., Wang, T., & Chen, Q. (2023a). Development of power transmission line detection technology based on unmanned aerial vehicle image vision. *SN Applied Sciences*, *5*(3), 72. http://dx.doi.org/10.1007/s42452-023-05299-7.

Xu, B., Zhao, Y., Wang, T., & Chen, Q. (2023b). Development of power transmission line detection technology based on unmanned aerial vehicle image vision. *SN Applied Sciences*, *5*(3), 72. http://dx.doi.org/10.1007/s42452-023-05299-7.

Xue, N. (2021). Holistically-attracted wireframe parsing. GitHub repository, GitHub, https://github.com/cherubicXN/hawp.

Xue, N., Wu, T., Bai, S., Wang, F., Xia, G., Zhang, L., et al. (2020). Holistically-attracted wireframe parsing. In *2020 IEEE/CVF conference on computer vision and pattern recognition* (pp. 2785–2794).

Xue, N., Wu, T., Bai, S., Wang, F., Xia, G.-S., Zhang, L., et al. (2022). Holistically-attracted wireframe parsing: From supervised to self-supervised learning. ArXiv, abs/2210.12971.

Yan, G., Li, C., Zhou, G., Zhang, W., & Li, X. (2007). Automatic extraction of power lines from aerial images. *IEEE Geoscience and Remote Sensing Letters*, *4*(3), 387–391. http://dx.doi.org/10.1109/LGRS.2007.895714.

Yang, L., Fan, J., Huo, B., Li, E., & Liu, Y. (2022). PLE-Net: Automatic power line extraction method using deep learning from aerial images. *Expert Systems with Applications*, *198*, Article 116771. http://dx.doi.org/10.1016/j.eswa.2022.116771, URL: https://www.sciencedirect.com/science/article/pii/S0957417422002342.

Yang, L., Fan, J., Liu, Y., Li, E., Peng, J., & Liang, Z. (2020). A review on state-of-the-art power line inspection techniques. *IEEE Transactions on Instrumentation and Measurement*, *69*(12), 9350–9365. http://dx.doi.org/10.1109/TIM.2020.3031194.

Yang, L., Kong, S., Deng, J., Li, H., & Liu, Y. (2023). DRA-Net: A dual-branch residual attention network for pixelwise power line detection. *IEEE Transactions on Instrumentation and Measurement*, *72*, 1–13. http://dx.doi.org/10.1109/TIM.2023.3259047.

Yetgin, Ö. E., Benligiray, B., & Gerek, Ö. N. (2019). Power line recognition from aerial images with deep learning. *IEEE Transactions on Aerospace and Electronic Systems*, *55*(5), 2241–2252. http://dx.doi.org/10.1109/TAES.2018.2883879.

Zhang, H., Yang, W., Yu, H., Zhang, H., & Xia, G.-S. (2019). Detecting power lines in UAV images with convolutional features and structured constraints. *Remote Sensing*, *11*(11), http://dx.doi.org/10.3390/rs11111342, URL: https://www.mdpi.com/2072-4292/11/11/1342.

Zhang, J., meng Yang, H., ning Zhang, Z., Zhao, K., fang Chen, Y., & Wu, X. (2016). An automatic diagnostic method of abnormal heat defect in transmission lines based on infrared video. In *2016 4th international conference on applied robotics for the power industry* (pp. 1–4). http://dx.doi.org/10.1109/CARPI.2016.7745629.

Zhao, T., & Guo, J. (2019). Ultraviolet detection and location of power line corona in UAV track. *Optics Precision Engineering*, *27*(02), 309–315.

Zhou, Y., Qi, H., & Ma, Y. (2019). End-to-end wireframe parsing. In *2019 IEEE/CVF international conference on computer vision* (pp. 962–971).

Zu-jian, X. (2008). Research on transmission-lines-cruising technology with the unmanned aerial vehicle. *Southern Power System Technology*, URL: https://api.semanticscholar.org/CorpusID:113206298.