Journal of King Saud University – Computer and Information Sciences 35 (2023) 101615

HOSTED BY

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

# PLPose: An efficient framework for detecting power lines via key points-based pose estimation

Rabeea Jaffari [a,b,*], Manzoor Ahmed Hashmani [b], Constantino Carlos Reyes-Aldasoro [c], Aisha Zahid Junejo [b], Hasmi Taib [d], M. Nasir B. Abdullah [e]

[a] Software Engineering Department, Mehran University of Engineering and Technology, Jamshoro, Sindh, Pakistan
[b] Department of Computer and Information Science, Universiti Teknologi PETRONAS, Seri Iskandar, Perak, Malaysia
[c] giCentre, Department of Computer Science, City, University of London, London EC1V 0HB, UK
[d] Floating Production Facilities, Civil and Structural Section, Engineering Department, Group Technical Solutions, Project Delivery & Technology Division, Petroliam Nasional Berhad (Petronas), Kuala Lumpur, Selangor, Malaysia
[e] Petroliam Nasional Berhad (Petronas), Kuala Lumpur, Selangor, Malaysia

ARTICLE INFO

*Article history:*
Received 8 February 2023
Revised 18 May 2023
Accepted 8 June 2023
Available online 14 June 2023

*Keywords:*
Power lines
Cable detection
Unmanned aerial vehicles
Key point pose estimation
Deep learning

ABSTRACT

The inspection and maintenance of electrical power lines (PL) via unmanned aerial vehicles (UAV) require fast and accurate PL detection to ensure smooth and secure electrical operations. However, the detection of PLs from aerial images is a highly challenging task due to the thin nature of PLs and the inherent noisy image backgrounds. Traditional line and edge detection methods do not detect the PLs accurately due to the cluttered backgrounds while the more recent deep learning (DL) CNNs are also not feasible for efficient PL detection due to the coarse bounding boxes and the computationally expensive pixel-based segmentations. Hence, in this study we propose PLPose, a novel framework for detecting the PLs via key points-based pose estimation technique and adapt the MobileNetV3 CNN for this task (kMobileNetV3), by adding a simple key point detection head to predict the PL key points. We also introduce a novel data-centric architecture (kMobileNetV3 + UDP), by adding the unbiased data processing (UDP) module to our kMobileNetV3, for faster and more accurate key point-based PL detection along with novel methods for data annotations and performance evaluation. Evaluations of PLPose on three benchmark PL datasets (PLDM, PLDU and the Mendeley Powerline Dataset) reveal that our proposed framework outperforms the state-of-the-art top-down pose estimation networks (HRNet-w32, HRNet-w32 + UDP and Resnet-50 Simple Baselines) in processing speed (~29 FPS) and model size (5.23 M) for PL detection. Thus, the comprehensive experimental results demonstrate the effectiveness of our proposed framework. Our code is available from Github (https://www.github.com/rubeea/pl_mmpose).


## 1. Introduction

Power lines (PL) or transmission cables are the backbones of electrical infrastructure in all urban and rural areas. Any damage or faults in the PLs can disrupt the continuous supply of electricity and can lead to power outages. There are many possible problems that can cause faults in PL, for instance, tree contact, equipment damage, extreme weather conditions and animal or human intervention. These have been reported to be the major causes of PL faults and the subsequent power outages (Wang, 2016). Power outages incur severe economic and societal problems, such as maintenance and restart costs, equipment damage, discomfort at homes or workplaces and among others (Amadi & Okafor, 2015). Hence, to avoid these problems, it is crucial to guarantee correct operation of PLs, whose reliability is ensured via frequent inspection and maintenance operations.

Traditional approaches to PL inspection comprise of visual inspection from the ground by utilities employees, via regular foot patrolling (Russell et al., 2007). Recently, unmanned aerial vehicles (UAV), equipped with vision systems, have replaced and

* Corresponding author at: Software Engineering Department, Mehran University of Engineering and Technology, Jamshoro, Sindh, Pakistan.
*E-mail addresses:* rabeea.jaffari@faculty.muet.edu.pk, rabeea_19000989@utp.edu.my (R. Jaffari), manzoor.hashmani@utp.edu.my (M.A. Hashmani), constantino-carlos.reyes-aldasoro@city.ac.uk (C.C. Reyes-Aldasoro), aisha_19001022@utp.edu.my (A.Z. Junejo), hasmi@petronas.com (H. Taib), mnasir_abdullah@petronas.com (M. Nasir B. Abdullah).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2023.101615
1319-1578/

automated the manual inspection approaches due to several advantages, namely, their effortless approach, consistent results and the ability to reach remote areas conveniently (Li et al., 2010; Miao et al., 2019). PL detection constitutes the core of such UAV-based inspection systems. Accurate and timely PL detection is required for the safe navigation and real time motion planning of UAVs. Besides UAV-based inspection, efficient PL detection is also necessary for safe navigation and obstacle avoidance in aircrafts operating at low altitudes.

The PL detection task is considered highly challenging due to the thin morphological structure, weak visual appearance and pervasive existence of PLs in aerial images (Yan et al., 2007). Moreover, the presence of background clutter and various lighting conditions (bright, dark, foggy etc.) further add to the difficulty of accurately detecting the PLs from aerial images (Li et al., 2010; Memon et al., 2021; Zhang et al., 2019a; Zhou et al., 2016). According to the reports of United States Army, more helicopters have been lost to PLs, in urban search and rescue (USAR) missions, than against enemies in actual combats due to the failure in the identification of these PLs in low-contrast and heavily cluttered backgrounds (Avizonis & Barron, 1999).

Traditional methods of PL detection focus on hand-crafted edge and line detectors (Candamo, Kasturi, Goldgof, & Sarkar, 2006; Golightly & Jones, 2005; Golightly, 2006; Kasturi et al., 2002; Li et al., 2008; Li et al., 2010; Yan et al., 2007; Zhang et al., 2012). These methods work under the assumptions of a distinct contrast between the PLs and the background, and straight PL shapes (Candamo, Kasturi, Goldgof, & Sarkar, 2009; Kasturi et al., 2002). Hence, they fail to detect PLs accurately in complex image environments and noisy backgrounds (Dai et al., 2020). Few works also employ machine learning techniques for PL detection but are limited by their requirement to design hand-crafted features which do not generalize well for complex image scenarios (Yetgin & Gerek, 2018a, 2018b). The recent deep learning (DL) semantic segmentation CNNs have shown remarkable performances in a variety of object detection tasks (Huang et al., 2022; Wang et al., 2022; Yun et al., 2022) and hence, have been successfully applied to detect PLs at the pixel level with remarkable results (Choi et al., 2019; Madaan et al., 2017; Zhang et al., 2019). However, these methods come with a huge computational cost and low inference speed associated with pixel-level classification. Semantic segmentation detectors also suffer from insufficient spatial resolution (Hao et al., 2020; Zhao et al., 2019), lack of global context (Baker et al., 2018; Z. Huang et al., 2019) and missing inter-pixel relationships (Ghafoorian et al., 2018; Zhao et al., 2019), all of which are necessary to accurately detect the PLs. Moreover, these semantic segmentation PL detectors also require additional techniques to:

- Handle class imbalance issue as PL pixels only occupy a minimal portion 1–5% of the aerial images in comparison to the majority of background pixels 95–99% (Jaffari, Hashmani, & Reyes-Aldasoro, 2021a).
- Post-process and cluster the ungrouped detected PL pixels, resulting from insufficient contextual information, for a uniform and more accurate PL representation.

Lastly, the performance of the semantic segmentation detectors is directly proportional to the labelling cost. Pixel-level annotations are generally time-consuming and commercially expensive. According to the Microsoft COCO dataset annotation report, the workload of pixel level annotations is 15 times more than that of spotting object locations (Lin et al., 2014). For PL detection, this workload will increase even more as PLs are only a few pixels wide and therefore, must require precise and careful pixel level annotations (Choi, Koo, Kim, & Kim). Usually, bounding box based CNNs are considered viable alternatives to semantic segmentation detectors due to their high inference speed and minimal annotation effort. However, these detectors are also not suitable for detecting thin PLs as the coarse bounding box detections contain several meaningless pixels which do not correspond to the thin PL objects of interest (Jaffari et al., 2021b; Li et al., 2019a). Few works focus on weakly supervised learning (Choi et al.) and key points-based PL detection (Dai et al., 2020; Sumagayan et al., 2021) to bypass the pixel-level annotations and rectangular bounding boxes. However, these techniques require a post-processing module for PL curve fitting and line refinement. Another alternative category of detection methods is the generic DL-based line segment detectors (Huang et al., 2018; Li et al., 2021; Xue, Bai, Wang, Xia, Wu, & Zhang, 2019; Xue, Wu, Bai, Wang, Xia, Zhang, & Torr, 2020; Zhou et al., 2019) which may be utilized to detect PLs due to the geometrical similarities between generic lines and thin PLs. However, these line segment detectors either assume the lines to be straight and/or focus on joint inference of junctions (line segment end points) along with the saliency of lines which might not always hold true for PLs. In addition to this, these methods also suffer from insufficient spatial information which hinders accurate PL detection (Abdelfattah et al., 2022). Few miscellaneous works (Abdelfattah et al., 2022; Xu et al., 2021) propose to utilize GAN and transformer networks for PL and line detection respectively but suffer from painfully long training times. Hence, there is a need for an efficient PL detection framework that successfully tackles the aforementioned issues.

To overcome the problems of coarse bounding box detectors, computationally expensive pixel-level detectors and the associated pixel-level annotations, class imbalance and post-processing techniques, in this paper, we propose PLPose, an efficient end-to-end DL framework for processing and detecting PLs from aerial images directly via key points-based pose estimation. We treat PL detection as a key point-based top-down pose estimation task similar to human pose estimation (HPE). We empirically investigate various key point-based annotation protocols and introduce a novel data labeling method for PL detection based on only three representative key points. For PL detection via pose estimation, we modify the vanilla MobileNetV3 (small version) CNN model (Howard et al., 2019) by appending a simple key point head, with three deconvolutional layers (dconv), for predicting PL key point heatmaps (Xiao et al., 2018) and name it kMobileNetV3 ($k$ for key points). Being a pose estimation network, the proposed kMobileNetV3 rules out the necessity of a separate post-processing module to cluster and connect the detected key points. Instead, the connection between the key points is directly estimated as pose for PL representation. We also incorporate an additional data-centric unbiased data processing module (UDP) to our kMobileNetV3 (Huang et al., 2020), which is employed to improve the detection performance in HPE tasks. MobileNetV3 (small version) is chosen as the base network for our PLPose framework as it reduces the network parameters and speeds up the detection process with little to no effect in the detection performance. UDP is employed to improve the detection performance by compensating for the bias usually introduced, due to the flipping strategy and encoding decoding statistical error, in pose estimation tasks. To the best of authors' knowledge, this approach has not been introduced previously. The effectiveness of our proposed PLPose framework is demonstrated by validating it on three public PL datasets: namely, (1) the Mendeley power line dataset (Yetgin & Gerek, 2019), (2) the power line dataset of a mountain scene (PLDM) and (3) power line dataset of an urban scene (PLDU), published in (Zhang et al., 2019). The experimental results demonstrate that PLPose outperforms the state-of-the-art in terms of inference speed and model complexity while maintaining equivalent characteristic detection parameters

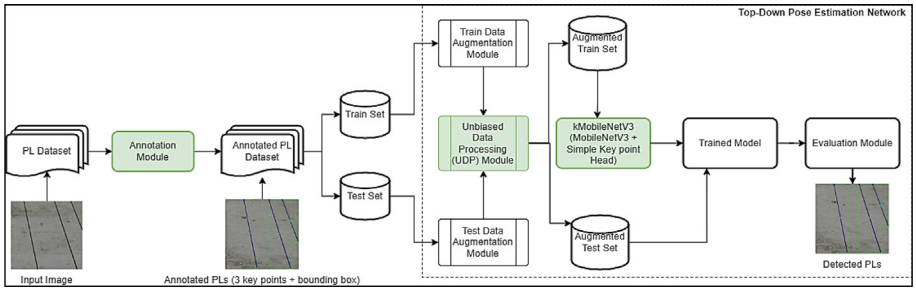


**Fig. 1.** Schematic Diagram of proposed PLPose for processing and detecting PLs via key points-based pose estimation. The main modules are highlighted in green color.

of precision and recall. The schematic diagram of PLPose is depicted in Fig. 1. The major contributions of this work are four-fold:

- A novel pipeline for processing and detecting PLs from aerial images via key points-based top-down pose estimation. At the core of this pipeline is the top-down pose estimator for PL detection.
- Adaptation of MobileNetV3 for the key points-based top-down pose estimation task (kMobileNetV3) via the addition of a simple key point head comprising of three dconv layers to predict the key point heatmaps.
- A novel top-down pose estimation model architecture (kMobileNetV3 + UDP), realized via the extension of kMobileNetV3 with a data-centric unbiased data processing module (UDP), for the efficient detection of PLs.
- Empirical analysis of the choice of annotation protocol and introduction of a new data labeling method for PL representation. We represent PLs via a group of three representative key points only. The proposed annotation format is more accurate and concise as compared to the existing annotation formats of rectangular bounding boxes or pixel level segmentation masks.

It is noteworthy that the scope of this study is limited to efficient PL detection which can aid in subsequent PL faults detection and even power grid faults detection (Gao et al., 2021) via constant monitoring mechanisms.

The remainder of this paper is organized as follows: Section 2 of this study briefly discusses the related works for PL detection. Section 3 proposes the novel PLPose framework for PL detection task via key points-based pose estimation. Section 4 presents the related new data labeling method. Section 5 presents the experimental results and discussions along with a detailed comparative analysis of PLPose with established baselines. Finally, conclusion and future work is presented in Section 6.

## 2. Related work

This section reviews various traditional and deep learning techniques for PL detection.

### 2.1. PL detection using traditional techniques

The traditional PL detection techniques can be divided into three main categories: (1) edge and line-based methods, (2) knowledge-based methods and (3) machine learning based methods.

#### 2.1.1. Line-based methods

Most of the traditional PL detection literature considers edge and line detectors on the assumption that PLs usually appear as straight and parallel lines in aerial images. The Hough Transform (HT) (Hough, 1962) and its variants were frequently used to detect PLs. The works in (Candamo, Kasturi, Goldgof, & Sarkar, 2006; Kasturi et al., 2002; Wu et al., 2010) first identified the edges of PLs using an edge detector such as Steger's method (Kasturi et al., 2002), SWIFTS algorithm (Wu et al., 2010) or Canny edge detector (Candamo et al., 2006) respectively, followed by the application of conventional HT in (Kasturi et al., 2002; Wu et al., 2010) and windowed HT in (Candamo et al., 2006) to detect PLs from the extracted edge map. Zhang et al. (Zhang et al., 2012) used HT to detect PLs followed by the clustering and tracking of PLs using K-means clustering algorithm and Kalman filter. The vision system developed by Golightly and Jones (Golightly & Jones, 2005; Golightly, 2006) used HT to identify overhead PLs. Instead of conventional edge detectors, Li et al. (Li et al., 2008; Li et al., 2010) used a single layered Pulse Couple Neural Network (PCNN) to filter out the background noise and generated an intriguing edge map of PLs which was then passed to HT for detection. Apart from HT, Radon transform (RT) has also been employed for PL detection by Yan et al. in (Yan et al., 2007).

#### 2.1.2. Knowledge-based Methods

Knowledge-based methods tend to leverage the contextual PL characteristics like structure, color, and shape for efficient detection. Thin line structure, parallel lines, flat color, and co-existence with pylon poles are some of the auxiliaries employed by these methods for PL detection. Zhang *et al.* (Zhang et al., 2014) used the spatial correlation between the PLs and pylon poles to detect PLs. The authors in (Ceron & Prieto, 2014) employed a circle-based search technique to detect PLs based on the fact that PLs are straight parallel lines and two equidistant points on the straight line always lie at the circumference of the circle. An edge drawing method was reported by authors in (Santos et al., 2017) to detect PLs based on their form and parallelism auxiliary. Most of these auxiliaries were determined manually (Pan et al., 2017). However, the technique introduced in (Shan et al., 2017) tried to automate the auxiliary determination process via a local optimization approach.

#### 2.1.3. Machine Learning based Methods

Yetgin (Yetgin & Gerek, 2018a, 2018b) introduced a new feature selection mechanism based on Discrete Cosine Transform (DCT) for the binary classification of PL scene recognition. Spectral clustering approach was utilized by Rishav Bhola *et al.* (Bhola et al., 2018) to segment PL pixels from aerial images. Apart from these, many works (Li et al., 2008; Li et al., 2010; Wu et al., 2010; Zhang et al., 2012) utilized the k-means and nearest neighbor machine learning algorithms for clustering the detected lines.

Despite the development and improvement of the traditional approaches, the accurate detection of PLs in diverse and complex

image environments remains challenging due to noisy backgrounds and the need of hand-crafted auxiliaries.

### 2.2. PL detection using deep learning

The recent approaches to PL detection rely on DL techniques due to their ability to automatically extract the PL features from aerial images and work well for complex image environments. The works in (Yetgin et al., 2018; Zhang et al., 2018b) tend to classify the PL objects, against the background, from aerial images without localization. For detection and localization, pixel-based semantic segmentation is the most commonly used DL technique. Madaan *et al.* (Madaan et al., 2017) first utilized a CNN with dilated convolution operation for segmenting PL pixels using synthetically generated PL images. Heng Zhang et al. (Zhang et al., 2019) fused the convolutional and structural features of PLs for accurate detection. Li et al. (Li et al., 2019b) proposed attentional fusion network for detecting PL pixels. Nguyen et al. (Nguyen et al., 2019) proposed LS-Net, a fast single-shot line segment detector for detecting PLs generated synthetically via a physical based rendering technique. U-Net segmentation architecture and its variants were deployed for segmenting PL pixels in (Saurav et al., 2019). PLGAN (Abdelfattah et al., 2022) uses a GAN network for pixel-wise PL detections. The greatest disadvantages of pixel-level detection methods are the computational complexity and lack of large, publicly available pixel-wise annotated datasets. The only publicly available pixel-wise PL datasets are: Mendeley power line dataset (Yetgin & Gerek, 2019) and, Power line dataset of urban (PLDU) (Zhang et al., 2019b) and mountain scenes (PLDM) (Zhang et al., 2019a) introduced in (Ö. E. Yetgin et al., 2018) and (Zhang et al., 2019b) respectively. These datasets are not large and contain approximately 500 images or less. The works in (Madaan et al., 2017; Nguyen et al., 2019) utilized PL datasets which are not publicly available.

Several approaches have been introduced to cope with the computational complexity and time-consuming annotations associated with the pixel level methods. For instance, the works in (Choi et al.; Li et al., 2018) utilized patch level labels instead of pixel annotations for detecting PLs. These techniques usually required line refinement and clustering methods to construct the final PLs.

Apart from the pixel-level methods, generic DL line-based methods can also be used for thin PLs detection. Most of these methods rely on junction information for efficient line detection, for instance, HAWP (Xue et al., 2020) and DWP (Huang et al., 2018) use a stacked Hourglass backbone network to collectively detect the line segments and the junctions. L-CNN only detects the junctions using a stacked Hourglass network and then infers the corresponding line segments from it. AFM (Xue et al., 2019) circumvents the heuristic-based junction detection and constructs an attraction field map using the U-Net backbone network for lines detection but demonstrates an inferior performance as compared to the junction-based methods. ULSD (H. Li et al., 2021) represents the line segments using a Bezier curve for efficient detection from both distorted and undistorted images. It is notable that these methods cannot handle the curved PLs due to their reliance on saliency of straight lines and/or junction information which might not be present for PLs in aerial images. Moreover, the quality of these line segment detectors may not be very good due to the inherent spatial region partitioning which causes dislocations between the detected PLs and the corresponding GTs (Abdelfattah et al., 2022). All these techniques rely on heuristics-based modules (edge, region, junction), line grouping and post processing mechanisms which are refuted by LETR (Xu et al., 2021) to detect lines via a transformer network. However, being a computationally expensive transformer network LETR is limited by its ability to efficiently work on target embedded environments.

Key points have also been used to represent and detect PL objects in (Dai et al., 2020; Sumagayan et al., 2021) due to their concise format. The authors in (Dai et al., 2020) used a group of five key points to represent a PL while the authors in (Sumagayan et al., 2021) used dense key points similar to pixel annotations. However, the problem with key points-based approaches is that they require additional post-processing techniques such as curve fitting to construct PLs from the detected key points.

## 3. Materials and methods

### 3.1. Materials

#### 3.1.1. Dataset

We utilized three benchmark PL datasets in this study namely, the Mendeley PL Dataset (Yetgin & Gerek, 2019), the Power line Dataset of Urban Scenes (PLDU) (Zhang et al., 2019b) and the Power line Dataset of Mountain Scenes (PLDM) (Zhang et al., 2019a). These datasets contain the visual light images of electrical PLs captured by a UAV along with their corresponding pixel-wise ground truth (GT) annotations. For PLPose, we generated the key point and bounding box annotations for these datasets using the annotation module. The Mendeley PL dataset is composed of 400 images of size 512 $\times$ 512, out of which 200 images contain the PLs while the remaining 200 images only contain the background with no PLs in them. We only considered the 200 images containing the PLs for our experiment and split this dataset according to the 80:20 train-test split so that the training dataset contains 160 images while the test dataset contains the remaining 40 images. We employed stratified sampling method to split the dataset into train-test sets. The PLDU and PLDM datasets are already split into training and test sets. The images in these datasets are either of size 560 $\times$ 360 or of size 360 $\times$ 560. We resized all the images to a standard size of 256 $\times$ 256 for our experiments. The details of these PL benchmark datasets are summarized in Table 1. Some of the images from these datasets are illustrated in Fig. 2.

### 3.2. Methods

PLPose takes on a top-down pose estimation approach to detect the PLs represented via key points. The details are discussed in the succeeding subsections, but first, since the key points were inspired by Human Pose Estimation, this will be described first.

#### 3.2.1. Human pose estimation (HPE)

Human pose estimation (HPE) focuses on estimating the configuration of human body parts from images or videos. The human body parts are most commonly modeled via a kinematic or skeleton-based model (Chen et al., 2020) in which the body joints are represented using key points and the connections between these key points represent the limb orientations as depicted in Fig. 3.

HPE is generally employed for visual understanding tasks such as action recognition and tracking. Depending on the number of people, HPE methods can be categorized into single-person and multi-person methods. Since multi-person HPE is more appropriate for real life scenarios it is achieved via two major techniques, namely, the top-down pose estimation and the bottom-up pose estimation.

The top-down pose estimation pipeline first detects the human objects in the image via rectangular bounding boxes and then predicts location of key points within these boxes. On the other hand, the bottom-up pose estimators do not utilize any bounding boxes and directly start with regressing the key points followed by the

**Table 1**
Detailed Information of the PL Datasets used in this Study.

| S.# | Dataset | Train | Test | Image Size |
|---|---|---|---|---|
| 1 | Mendeley PL Dataset (Yetgin & Gerek, 2019) | 160 | 40 | 512 × 512 |
| 2 | PLDU (Zhang et al., 2019b) | 453 | 120 | 560 × 360 or 360 × 560 |
| 3 | PLDM (Zhang et al., 2019a) | 237 | 50 | 560 × 360 or 360 × 560 |

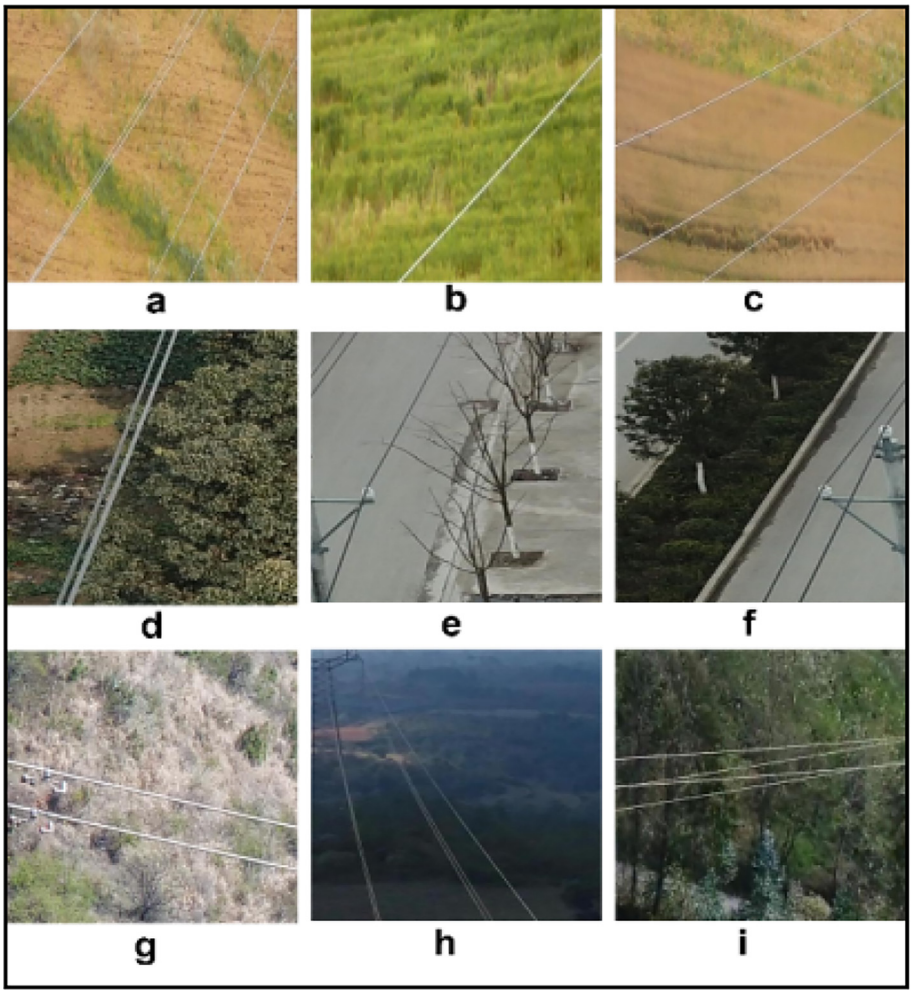


**Fig. 2.** Example images of PLs. Top Row: Mendeley PL dataset (Yetgin & Gerek, 2019), middle row: PLDU (Zhang, 2019b) and bottom row: PLDM (Zhang, 2019a). It should be noted that lines are not always straight lines and that the contrast can be very different like the Fig. 2 (e) and Fig. 2 (h).

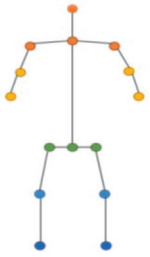

**Fig. 3.** Kinematic/Skeleton-based model for human body modeling in HPE. Various human joints are represented via 15 different colored key points. The representations are color-coded as: orange: head, light orange: neck and shoulders, yellow: elbow and hands, green: hips and groin, light blue: knees, dark blue: feet. (). Adapted from Zheng et al., 2020

assembling of the detected key points for final pose representations (Zheng et al., 2020). The top-down and bottom-up pose estimation pipelines are depicted in Fig. 4. The top-down pose estimation methods dominate the HPE performances due to their fast inference speeds and ability to work with small datasets.

*3.2.2. PLPose: An efficient framework for detecting PLs via key points-based pose estimation*

The schematic diagram for PLPose is presented in Fig. 1. For PLPose, we take inspiration from works (Dai et al., 2020; Sumagayan et al., 2021), which treat PL detection as a key points estimation task. To avoid the rigorous approaches of curve fitting or clustering for post-processing the detected key points, to cope with the available small scale PL datasets and to detect multiple PLs per image, PLPose frames PL detection as a top-down pose estimation task that is it first detects the PLs objects in the image via rectangular bounding boxes and then predicts the location of PL key points within these boxes. Hence, there is no limit on the number of PLs detected by PLPose. As shown in Fig. 1, PLPose consists of three main modules highlighted in green: (1) the annotation module (2) the kMobileNet V3 module (3) and the UDP module.

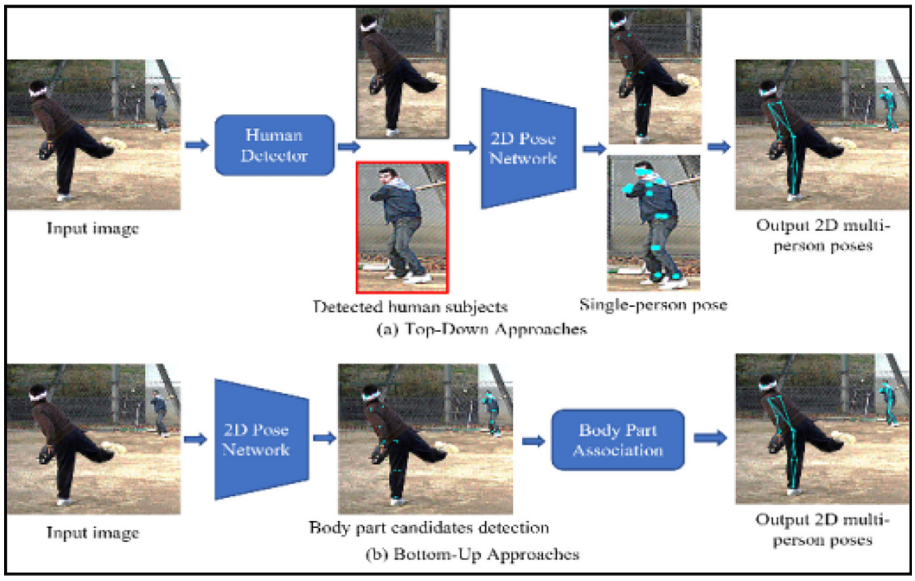


**Fig. 4.** Illustration of top-down and bottom-up HPE pipelines (a) Top-down approaches have two sub-tasks: (1) human detection and (2) pose estimation in the region of a singe human; (b) Bottom-up approaches also have two sub-tasks: (1) detect all keypoints candidates of body parts and (2) associate body parts in different human bodies and assemble them into individual pose representations. (). Adapted from Zheng et al., 2020

### 3.3. Annotation module

This module is responsible for determining the appropriate key points for the PL annotation. The PLs are then annotated accordingly via manual annotators. Each PL in the aerial image is represented via a set of three representative key points: start, center, and end, which were selected after the empirical investigation of two key point annotation protocols (discussed in Section 4) by the annotation module. The connections between these key points constitute the pose of a PL for final PL representation. Since top-down pipeline first identifies the objects of interest via bounding boxes, hence bounding box annotations are also required for training a top-down pose estimator. The annotation module, however, extracts the bounding box representations of PLs automatically from the annotated key points. A sample of the PL annotation obtained via the annotation module is depicted in Fig. 5. The final annotated PL dataset is employed for the training and performance evaluation of the top-down pose estimation network.

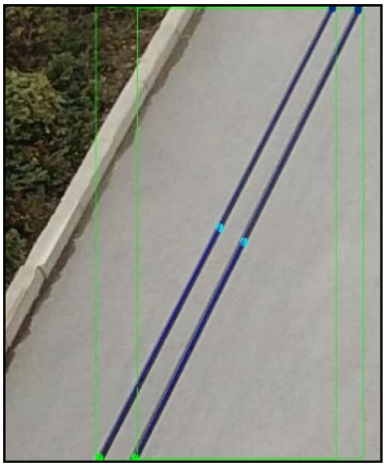

**Fig. 5.** Sample of PL annotation from the PLPose annotation module. In this image, two PLs are labelled with three points each, the start and end extremes are labelled with dark blue and green dots respectively and the centre with a cyan dot. A bounding box shows the region of interest. It should be noticed that if an edge detection method would be used, the left edge of the surface would generate false positive lines.

### 3.4. kMobileNetV3 module

High resolution network (HRNet) (Sun et al., 2019) is considered the state-of-the-art in top-down HPE task due to its stronger feature representation capability achieved via parallel multi-scale feature fusion instead of a series of low-to-high down-sampling and up-sampling feature extraction process. However, HRNet is computationally expensive and results in a model with increased network parameters. MobileNetV3 is a light-weight model designed for achieving acceptable performance on various computer vision tasks in resource constrained devices (Howard et al., 2019). It extends MobileNetV2 (Sandler et al., 2018) model by using AutoML (Gupta & Tan, 2019) to find the optimal architecture best suited to mobile computer vision tasks. The basic building block of MobileNetV3 comprises of inverted residual bottlenecks from MobileNetV2 along with the Squeeze and Excitation (SE) blocks which enhance the features that contribute more to the detection task and suppress the non-contributing features. Hard-swish (H-swish) non-linearity was introduced, to work efficiently in embedded environments, as an alternative to the swish linearity function. MobileNetV3 is available in small and large versions for different resource platforms. The MobileNetV3 block is depicted in Fig. 6.

For PLPose, we adopted the vanilla MobileNetV3 as the backbone network for efficient processing and simply added a head network over the last convolution stage for key points pose estimation following the work in (Xiao et al., 2018). The head network consisted of three deconvolution (dconv) layers each with 256 filters, 4x4 convolution kernel, stride 2 and padding 1. Batch Normalization and ReLu activation were used in each of the three dconv layers. A $1 \times 1$ convolutional layer was added at last to generate predicted heatmaps for all k (k = 3 for PLPose) key points. We employed an image input of size $256 \times 256 \times 3$ and predicted heatmaps of size $64 \times 64 \times k$ where k is the number of key points.

It should be noted that either of the vanilla MobileNetV3 versions (small or large) may be employed as the backbone in our kMobileNetV3 network. The proposed model is depicted in Fig. 7 and the detailed model specification along with its layers is presented in Table 2 for the small version and in Table 3 for the large version.

### 3.5. Unbiased data processing (UDP) module

The unbiased data processing (UDP) is a model-agnostic data processing technique to improve the key points detection performance in HPE by handling the bias and statistical error introduced in top-down pose estimators due to flipping augmentation and encoding–decoding process (J. Huang et al., 2020). We investigated the application of UDP to our proposed PLPose top-down pose estimation network (kMobileNetV3) to understand its effect on the PL detection performance.

UDP was applied to PLPose during the affine data augmentation operation in the train and test phases as depicted in Fig. 1. The final datasets from the UDP module were utilized during the model training and evaluation.

Our proposed top-down pose estimation models kMobileNetV3 and kMobileNetV3 + UDP achieved state-of-the-art performance on benchmark PL datasets in terms of model size and inference speed as discussed in Section 5.

## 4. Manual data labeling

The public PL datasets (Mendeley PL dataset (Yetgin & Gerek, 2019), PLDU (Zhang et al., 2019b) and PLDM (Zhang et al., 2019a)) comprise of pixel-level annotations. Hence, due to the unavailability of large-scale public datasets with key points annotation for PL detection, we proposed a data labeling method to represent PLs via key points. We identified and investigated two annotation protocols for labeling the PLDU and PLDM datasets and evaluated four top-down key point pose estimators on the labelled datasets to observe the detection performance (results in Section 5). The annotation protocol with the best results was chosen to annotate all the three PL datasets. The annotation protocols which were investigated are discussed as follows.

### 4.1. Annotation protocol 1: Six key points (k = 6)

Each PL was represented via six unique key points (k = 6):

start$_1$ ($S_1$), center$_1$ ($C_1$), end$_1$ ($E_1$), start$_2$ ($S_2$), center$_2$ ($C_2$) and end$_2$ ($E_2$), constituting the two distinct boundaries of the PL as shown in Fig. 8. The start ($S_1$, $S_2$) and end ($E_1$, $E_2$) points represented the two ends of the PL while the center points ($C_1$, $C_2$) are additional points that were used to label any bend or curve in the PL. This allowed us to even annotate PLs which were not straight lines. In the case of straight PLs, $C_1$ and $C_2$ represented the actual centers of the PLs. This annotation scheme can be extended to annotate PLs with more than one curve, or more complex geometry, by introducing additional center points ($C_{11}$, $C_{12}$ etc.). However, since the PLs in PLDU and PLDM datasets are mostly straight with maximum one curve (if any), we only used one center point for each boundary of the PL. We defined $(x_1, y_1)$, $(x_2, y_2)$, $(x_3, y_3)$, $(x_4, y_4)$, $(x_5, y_5)$, $(x_6, y_6)$ as pairs of $\times$ and y coordinates of $S_1$, $C_1$, $E_1$, $S_2$, $C_2$ and $E_2$ respectively. The key points were connected sequentially to form the skeletal pose for each PL. For k = 6, the key points were connected as follows: $S_1 \rightarrow C_1 \rightarrow E_1$ and $S_2 \rightarrow C_2 \rightarrow E_2$. The bounding box for each PL was then represented via (x, y, w, h) where (x, y), w and h are calculated according to Eq. (1), Eqs. (2) and (3) respectively.

$$(x, y) = (x_1, y_1) \tag{1}$$

$$w = (x_6 - x_1 + 1) \tag{2}$$

$$h = (y_6 - y_1 + 1) \tag{3}$$

Usually the start ($S_1$, $S_2$) and end points ($E_1$, $E_2$) are interchangeable, but we annotate $S_1$ and $S_2$ starting from the image top and $E_1$ and $E_2$ at the image bottom. Fig. 8 depicts these annotations on PL mask images, rather than original images, for better visualization. It takes approximately 6 s to annotate each PL with six key points.

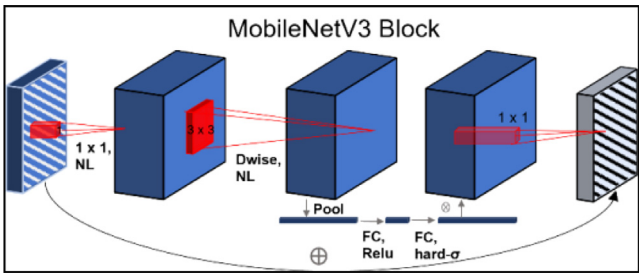


**Fig. 6.** MobileNetV3 Block: MobileNetV2 + Squeeze-and-Excite. The squeeze and excite is applied in the residual layer. Different nonlinearities (NL) (Relu and hard-σ) are applied depending on the layer. Dwise represents the depth-wise convolutions and FC represents the fully connected layers. (). Adapted from Howard et al., 2019

### 4.2. Annotation protocol 2: Three key points (k = 3)

Instead of six distinct key points, each PL was represented via three distinct key points (k = 3) only, namely: start (S), center (C) and end (E). The key points were labelled in the middle of the two PL boundaries as shown in Fig. 9.

The objective of reducing the annotation key points to half is to determine the optimal PL representation for the detection process, that is two representative lines for each PL curve (k = 6) versus only one representative line (k = 3). It took approximately 3 s to annotate the PLs with three key points. We defined $(x'_1, y'_1)$, $(x'_2, y'_2)$, $(x'_3, y'_3)$ as pairs of $\times$ and y coordinates of S, C and E respectively. For k = 3, the key points were connected as follows: S → C → E. The bounding box for each PL was then represented via (x, y, w, h) where (x,y), w and h are calculated according to (4), (5) and (6) respectively.

$$(x, y) = (x'_1, y'_1) \tag{4}$$

$$w = (x'_3 - x'_1 + 1) \tag{5}$$

$$h = (y'_3 - y'_1 + 1) \tag{6}$$

The key points for both the annotation protocols were initially labelled by a group of three researchers and then cross-checked by another group of three researchers. The resultant key point annotations are compatible with the Microsoft COCO Keypoint Dataset (Lin et al., 2014). It should be noted that our annotation module (Section 3) automatically calculated the bounding boxes from the labelled key points, using the calculations in Eqs (1)–(6), for both the annotation protocols. This makes our annotation scheme more efficient in contrast to the manual bounding box annotations which usually require 42 s for labelling each box (Su et al., 2012). The performance of four state-of-the art top-down pose estimation networks on the PLDU and PLDM datasets annotated via annotation protocols 1 and 2 are discussed in Section 5. We provide complete annotations using both the annotation protocols for all the mentioned datasets in our Github repository for public use and future research.

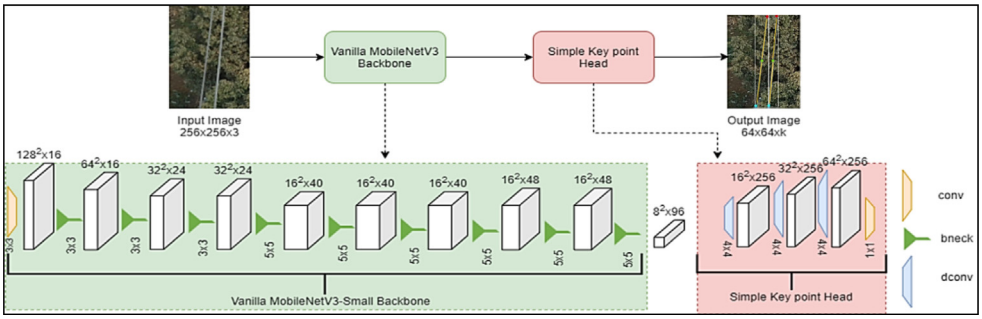


**Fig. 7.** The proposed kMobileNetV3 (MobileNetV3 + Simple Key Point Head) Network for PL detection via top-down pose estimation. MobileNetV3 small or large version can be used as the backbone in the network. In this image, MobileNetV3-Small constitutes the backbone for extracting the feature maps (8x8x96) from the image via downsampling while the simple key point head uses three dconv layers to generate the high resolution key point heatmaps (64x64xk) via upsampling process, where k is the number of key points. It should be noted that k = 3 (start,center and end) for PL detection in this study. PL start, center and end are detected via red, green and cyan key points and the PL is detected as the pose with a yellow line along with a white bbox.

**Table 2**
Specification for MobileNetV3-Small: Top-down Pose Estimation Network for PLPose. SE denotes whether there is a Squeeze-And-Excite in that block. NL denotes the type of nonlinearity used. HS denotes h-swish and RE denotes ReLU. NBN denotes no batch normalization. s denotes stride.

| Component | Input | Operator | exp size | #out | SE | NL | s |
|---|---|---|---|---|---|---|---|
| MobileNet V3 (small) Backbone Network | $256^2 \times 3$ | conv2d, $3 \times 3$ | – | 16 | – | HS | 2 |
| | $128^2 \times 16$ | bneck, $3 \times 3$ | 16 | 16 | SE | RE | 2 |
| | $64^2 \times 16$ | bneck, $3 \times 3$ | 72 | 24 | – | RE | 2 |
| | $32^2 \times 24$ | bneck, $3 \times 3$ | 88 | 24 | – | RE | 1 |
| | $32^2 \times 24$ | bneck, $5 \times 5$ | 96 | 40 | SE | HS | 2 |
| | $16^2 \times 40$ | bneck, $5 \times 5$ | 240 | 40 | SE | HS | 1 |
| | $16^2 \times 40$ | bneck, $5 \times 5$ | 240 | 40 | SE | HS | 1 |
| | $16^2 \times 40$ | bneck, $5 \times 5$ | 120 | 48 | SE | HS | 1 |
| | $16^2 \times 48$ | bneck, $5 \times 5$ | 144 | 48 | SE | HS | 1 |
| | $16^2 \times 48$ | bneck, $5 \times 5$ | 288 | 96 | SE | HS | 2 |
| | $8^2 \times 96$ | bneck, $5 \times 5$ | 576 | 96 | SE | HS | 1 |
| | $8^2 \times 96$ | bneck, $5 \times 5$ | 576 | 96 | SE | HS | 1 |
| Simple Key Point Head Network | $8^2 \times 96$ | dconv2d, $4 \times 4$ | – | 256 | – | RE | 2 |
| | $16^2 \times 256$ | dconv2d, $4 \times 4$ | – | 256 | – | RE | 2 |
| | $32^2 \times 256$ | dconv2d, $4 \times 4$ | – | 256 | – | RE | 2 |
| | $64^2 \times 256$ | conv2d, $1 \times 1$ | – | k | – | – | 1 |

**Table 3**
Specification for KMobileNetV3-Large: Top-down Pose Estimation Network for PLPose. See Table 2 for notation.

| Component | Input | Operator | exp size | #out | SE | NL | S |
|---|---|---|---|---|---|---|---|
| MobileNet V3 (large) Backbone Network | $256^2 \times 3$ | conv2d, $3 \times 3$ | – | 16 | – | HS | 2 |
| | $128^2 \times 16$ | bneck, $3 \times 3$ | 16 | 16 | – | RE | 1 |
| | $128^2 \times 16$ | bneck, $3 \times 3$ | 64 | 24 | – | RE | 2 |
| | $64^2 \times 24$ | bneck, $3 \times 3$ | 72 | 24 | – | RE | 1 |
| | $64^2 \times 24$ | bneck, $5 \times 5$ | 72 | 40 | SE | RE | 2 |
| | $32^2 \times 40$ | bneck, $5 \times 5$ | 120 | 40 | SE | RE | 1 |
| | $32^2 \times 40$ | bneck, $5 \times 5$ | 120 | 40 | SE | RE | 1 |
| | $32^2 \times 40$ | bneck, $3 \times 3$ | 240 | 80 | – | HS | 2 |
| | $16^2 \times 80$ | bneck, $3 \times 3$ | 200 | 80 | – | HS | 1 |
| | $16^2 \times 80$ | bneck, $3 \times 3$ | 184 | 80 | – | HS | 1 |
| | $16^2 \times 80$ | bneck, $3 \times 3$ | 184 | 80 | – | HS | 1 |
| | $16^2 \times 80$ | bneck, $3 \times 3$ | 480 | 112 | SE | HS | 1 |
| | $16^2 \times 112$ | bneck, $3 \times 3$ | 672 | 112 | SE | HS | 1 |
| | $16^2 \times 112$ | bneck, $5 \times 5$ | 672 | 160 | SE | HS | 2 |
| | $8^2 \times 160$ | bneck, $5 \times 5$ | 960 | 160 | SE | HS | 1 |
| | $8^2 \times 160$ | bneck, $5 \times 5$ | 960 | 160 | SE | HS | 1 |
| Simple Key Point Head Network | $8^2 \times 160$ | dconv2d, $4 \times 4$ | – | 256 | – | RE | 2 |
| | $16^2 \times 256$ | dconv2d, $4 \times 4$ | – | 256 | – | RE | 2 |
| | $32^2 \times 256$ | dconv2d, $4 \times 4$ | – | 256 | – | RE | 2 |
| | $64^2 \times 256$ | conv2d, $1 \times 1$ | – | k | – | – | 1 |

## 5. Experiments and analysis

### 5.1. Implementation

We utilized *mmpose* framework (Contributors, 2020), a Pytorch-based framework, to implement our proposed kMobileNetV3 and kMobileNetv3 + UDP top-down pose estimation networks for PL detection via pose estimation. Since the size of the datasets was rather small for training a deep pose estimation network, the following on-the-fly data augmentations were applied to the images in sequential order in all the training datasets.

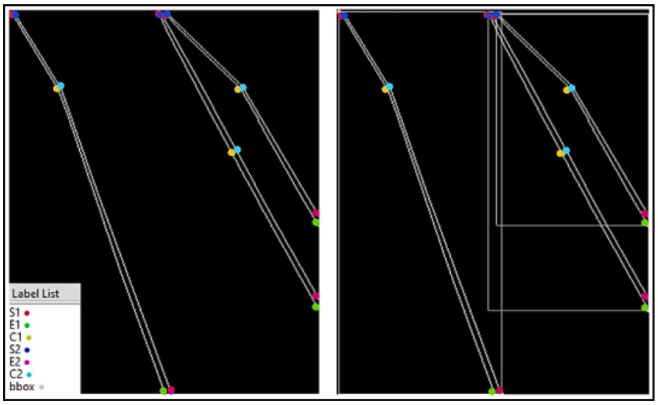


**Fig. 8.** Example of manual PL annotations with curves using six unique colored key points ($S_1$ (red), $C_1$ (yellow), $E_1$ (green), $S_2$ (dark blue), $C_2$ (cyan), $E_2$ (magenta)). First column: key point annotations; Last column: key point annotations with corresponding bounding boxes (bbox: grey).

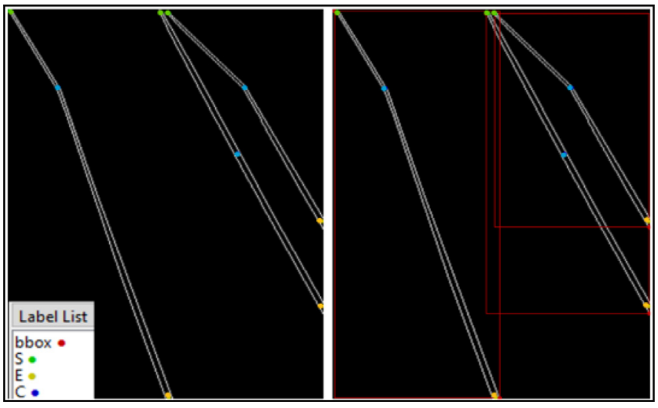


**Fig. 9.** Example of manual PL annotations with curves using three unique colored key points (S (green), C (dark blue), E (yellow)). First column: key point annotations; Last column: key point annotations with corresponding bounding boxes (bbox:red).

- Random horizontal flipping with a probability of 50%.
- Data augmentation with random scaling $[1 - scale\_factor, 1 + scale\_factor]$ and rotation $[-2 \times rotation\_factor, 2 \times rotation\_factor]$. A scale factor of 0.5 and a rotation factor of 40 with a rotation probability of 60% is applied.
- Affine data transformation with UDP enabled for the UDP variants of the network and disabled otherwise. The required scaling and rotation parameters are kept the same from the above random scaling and rotation data augmentation.
- RGB channel data normalization.

As far as the test datasets are concerned, only the horizontal flipping, affine transform and data normalization were applied during the test phase.

The network was trained on NVIDIA Tesla K80 GPU compute available via Google Colab, for 210 epochs with a batch size of 2. Mean squared error (MSE) loss was used for training the network with regular checkpointing at every 10 epochs. Adam optimizer was employed for model training with an initial learning rate (LR) of $5e^{-4}$ and a step learning policy with linear warmup iterations of 500, warmup ration 0.001 and steps in the range of [170,200]. These parameters were identified via a grid search method. No transfer learning was applied for training the proposed kMobileNetV3 and kMobileNetV3 + UDP networks in order to present a fair evaluation of the proposed networks against state-of-the-art networks commonly employed for top-down pose estimation.

### 5.2. Evaluation parameters

Since there is no public performance evaluation standard for PL detection via pose estimation, we adopted the standard COCO pose estimation criteria based on the object key point similarity (*OKS*) defined by (7) as follows:

$$OKS = \frac{\sum_i \left[ e^{\left( \frac{-d_i^2}{2s^2K_i^2} \right)} \delta(v_i > 0) \right]}{\sum_i [\delta(v_i > 0)]} \tag{7}$$

where:

- $d_i$: Euclidean distances between each corresponding ground truth and detected key point.
- $v_i$: the visibility flags of the ground truth (the detector's predicted $v_i$ are not used).

for each key point $i$.

To compute OKS, we pass the $d_i$ through an unnormalized Gaussian function with standard deviation $sK_i$ where $s$ is the object scale and $K_i$ is a per key point constant that controls falloff (COCO, 2016). The object scale $s$ is calculated via the bounding box area while $K_i$ is calculated via $K_i = 2\sigma_i$ where sigma ($\sigma_i$) is the normalized factor of the object skeletal key point and is calculated by the standard deviation of human annotation result. Hence, $K_i$ measures the annotation quality. The $\sigma_i$ s were set to 0.025 for all the $i$ key points in PL detection. For each key point this yielded an OKS value in the range [0,1]. Perfect predictions will have an OKS = 1 and predictions for which all key points are far off by more than a few standard deviations $sK_i$ will have OKS $\sim$ 0. The OKS is analogous to the intersection over union (IoU) in the case of object detection/segmentation. Given the OKS, we can compute AP and AR just as the IoU allows us to compute these metrics for box/segment detection by measuring the degree of overlap between the predicted and the ground truth values. As per the COCO evaluation criteria, the average precision and recall (based on OKS) are used to characterize the performance of our proposed key point detector. We used 6 COCO evaluation metrics to describe the performance of our model. They are AP, $AP_{0.5}$, $AP_{0.75}$, AR, $AR_{0.5}$ and $AR_{0.75}$ as shown in Table 4. The values 0.5 and 0.75 signify the OKS thresholds to adjust the matching criterion where 0.5 is a loose threshold and 0.75 is a stricter one. If the OKS is larger than a specified threshold, say 0.5, the corresponding ground truth and key points prediction are considered as a matching pair. The AP and AR metrics are averaged over 10 OKS threshold values (0.50:0.05:0.95). Apart from these metrics, we also considered the F1-score, model size and inference speed as additional criteria for evaluating the performance as per the PL detection studies in (Dai et al., 2020; Sumagayan et al., 2021). All the evaluation parameters are listed in Table 4.

### 5.3. Experimental results

We compared three state-of-the-art top-down pose estimation networks with our proposed networks (kMobileNetV3 and kMobileNetV3 + UDP in small and large versions), according to the evaluation parameters listed in Table 4. These networks are: HRNet-w32 (Sun et al., 2019), HRNet-w32 + UDP (J. Huang et al., 2020) and the Resnet-50 model for HPE from Simple Baselines (Xiao et al., 2018). It should be noted that basic versions of these models were selected in this study to keep the model size to minimum so that a high inference speed can be achieved for real-time monitoring. The evaluations were done on the public PL datasets

**Table 4**
PL Detection Evaluation Parameters used in this Study.

| S.# | Type | Metric | Description |
|---|---|---|---|
| 1 | Average | AP | % AP at OKS = 0.50:0.05:0.95 |
| 2 | Precision | $AP^{OKS=.50}$ | % AP at OKS = 0.50 (loose metric) |
| 3 | | $AP^{OKS=.75}$ | % AP at OKS = 0.75 (strict metric) |
| 4 | Average | AR | % AR at OKS = 0.50:0.05:0.95 |
| 5 | Recall | $AR^{OKS=.50}$ | % AR at OKS = 0.50 (strict metric) |
| 6 | | $AR^{OKS=.75}$ | % AR at OKS = 0.75 (loose metric) |
| 7 | F1-Score | F1-Score | Dice or F1-score |
| 8 | Model Size | Model Size | Trainable Model Parameters in million (M) |
| 9 | Inference Speed | Frames per second (FPS) | Detection speed in FPS |
| 10 | Total Time | Total Time | Total inference time on test dataset in seconds (s) |

labelled via the annotation module of PLPose using the annotation protocols (k = 6 and k = 3) discussed in Section 4.

For HPE, these networks have 17 channels in the output layer to predict the 17 human body joints via key points. Here, we changed the number of output channels from 17 to 6 in annotation protocol 1 (k = 6), and 17 to 3 in annotation protocol 2 (k = 3), to represent the characteristic key points of each PL.

First, we compared the performances of two annotation protocols: 1 (k = 6) and 2 (k = 3) on the PLDU and PLDM datasets to identify the optimal annotation protocol for labeling PLs via key points. The results for PLDU and PLDM datasets annotated via protocols 1 (k = 6) and 2 (k = 3) are summarized in Table 5, 6, 7 and 8 respectively. The best values for each parameter are specified in bold.

Table 5, 6, 7 and 8 reveal that the results obtained with annotation protocol 2 (k = 3) are more promising than those obtained with annotation protocol 1 (k = 6) (ref. to Discussions section for more details), hence, annotation protocol 2 (k = 3) is employed to label the Mendeley PL dataset. The performance of the mentioned top-down pose estimation networks on the Mendeley PL dataset is summarized in Table 9.

Supplementary plots depicting the tradeoff between the characteristic evaluation parameters (AP, AR, F1-Score, Model Size, and Inference Speed) for the mentioned top-down pose estimation networks, with the selected annotation protocol 2 (k = 3) are presented graphically in Figs. 10-12 for PLDU, PLDM and Mendeley PL datasets respectively. These plots reveal that our proposed networks (kMobileNetV3 and kMobileNetV3 + UDP in small and large versions) yield the most optimal configuration for the characteristic evaluation parameters, with least model size, highest inference speed and almost equivalent precision, recall and F1-Score values in comparison to the other top-down pose estimation networks (ref. to Discussion section for more details).

We also presented the results of the mentioned top-down pose estimation networks on the combined PLDU + PLDM dataset, referred to as PLD dataset in Table 10. This was done to compare the performance of the selected annotation protocol with three key points (k = 3) and the proposed top-down pose estimation networks (kMobileNetV3 and kMobileNetV3 + UDP in small and large versions) with the reported results from the paper that introduced the Convolutional Neural Network-based cable detection method (CNNCDM) (Dai et al., 2020) which used five key points and an encoder decoder style network to detect PLs in the PLD dataset.

The work in (Dai et al., 2020), CNNCDM, compared the performances of several DL-based methods on the PLD dataset, such as SSD (W. Liu et al., 2016), YOLOV3 (Redmon & Farhadi, 2018), RFBNet (S. Liu & Huang, 2018), FSSD (Zuoxin Li & Zhou, 2017), RetinaNet (Lin et al., 2017), RefineDet (Zhang et al., 2018a) and CenterNet (Duan et al., 2019) in terms of AP and processing time in frames per second (FPS). Since, the Figs. 10-12 reveal that our proposed networks achieve the best trade-off between the characteristic evaluation parameters, hence, the results of comparative analysis of the mentioned networks from (Dai et al., 2020) were only compared with the proposed networks (kMobileNetV3 and kMobileNetV3 + UDP in small and large versions) in terms of AP and FPS. For processing time, the greater the frames per second (FPS), the better the model performance. The results of AP and FPS are summarized in Table 11 and 12 respectively. All the results can be cited back from the work (Dai et al., 2020). However, it should be noted that these AP and FPS results (Tables 11-12) were obtained using a slightly different routine and hardware in comparison to our PLPose AP and FPS results. Nevertheless, we believe that these results are still comparable to our PLPose results due to the reasons stated in part 4 of the Discussions section.

We also compared the qualitative test results of various models from (Dai et al., 2020) with the modified PINet model (Sumagayan et al., 2021) and our proposed kMobileNetV3-Large + UDP on some sample images from the PLD dataset in Fig. 13. Various models from (Dai et al., 2020) include the convolutional features and structured constraints (CFSC) (Zhang et al., 2019) and some typical power cable inspection methods, like, Bi-directional cascade network perceptual edge detection (BDCN) (He et al., 2019), Richer convolutional features for edge detection (RCF) (Liu et al., 2017), Holistically-nested edge detection (HED) (Xie & Tu, 2015), Gestalt Grouping (Rajaei & von Gioi, 2018) and Canny (Canny, 1986). The sample images of the mentioned methods were copied from the CNNCDM paper (Dai et al., 2020). From our proposed models, only the kMobileNetV3-Large + UDP results are displayed in Fig. 13 due to its optimal results for all the evaluation parameters (ref. Table 7 and 8). We further qualitatively analyzed some success and failure modes of our best model variant (kMobileNetV3-Large + UDP) on some sample images from the PLD dataset in Fig. 14. All the presented results are discussed in detail in the succeeding subsection.

**Table 5**
Experimental Results of various top-down pose estimation networks (HRNet-w32, HRNet-w32 + UDP, Resnet-50 Simple Baseline, and Proposed Approach (kMobileNetV3 and kMobileNetV3 + UDP)) on PLDU Test Set (k = 6) with 120 Images.

| Network | AP % | $AP^{OKS=.50}$ % | $AP^{OKS=.75}$ % | AR % | $AR^{OKS=.50}$ % | $AR^{OKS=.75}$ % | F1-Score % | Model Size M | Inference Speed FPS | Total Times |
|---|---|---|---|---|---|---|---|---|---|---|
| HRNet-w32 | 38.8 ± 0.85 | 80.6 ± 1 | 33.6 ± 1.65 | 52.5 ± 0.83 | **86.7 ± 0.66** | 54.2 ± 1.55 | 44.6 ± 0.85 | 28.54 | 9.34 | 34.04 |
| HRNet-w32 + UDP | **40.8 ± 0.80** | **80.9 ± 0.85** | **34.6 ± 1.98** | **54.7 ± 1.43** | 86.7 ± 1.09 | 54.8 ± 2.15 | **46.7 ± 1.02** | 28.54 | 7.28 | 43.71 |
| ResNet-50 Simple Baseline | 36.0 ± 0.66 | 78.9 ± 1.40 | 28.8 ± 0.75 | 50.3 ± 0.85 | 85.8 ± 1.10 | 50.5 ± 0.85 | 41.9 ± 0.70 | 34.0 | 27.48 | 11.57 |
| kMobileNetV3-Small (Proposed) | 36.2 ± 0.15 | 76.7 ± 1.04 | 29.4 ± 1.05 | 50.2 ± 0.3 | 83.9 ± 0.6 | 51.1 ± 0.75 | 42.0 ± 0.25 | **3.36** | **42.17** | **7.55** |
| kMobileNetV3-Small + UDP (Proposed) | 38.3 ± 2.46 | 79.9 ± 2.38 | 34.9 ± 3.92 | 51.6 ± 2.67 | 84.5 ± 1.56 | 54.8 ± 3.74 | 43.9 ± 2.56 | **3.36** | 32.81 | 9.68 |
| kMobileNetV3-Large (Proposed) | 36.1 ± 1.05 | 77.0 ± 1.59 | 30.0 ± 2.63 | 49.6 ± 0.77 | 83.9 ± 0.75 | 51.4 ± 1.93 | 41.7 ± 0.95 | 5.23 | 36.43 | 8.71 |
| kMobileNetV3-Large + UDP (Proposed) | 38.8 ± 0.40 | 79.8 ± 0.93 | 33.7 ± 0.74 | 52.8 ± 1.10 | 85.4 ± 1.08 | **54.8 ± 1.12** | 44.7 ± 0.65 | 5.23 | 29.72 | 10.68 |

**Table 6**
Experimental Results of various top-down pose estimation networks (HRNet-w32, HRNet-w32 + UDP, Resnet-50 Simple Baseline, and Proposed Approach (kMobileNetV3 and kMobileNetV3 + UDP)) on PLDM Test Set (k = 6) with 50 Images.

| Network | AP % | $AP^{OKS=.50}$ % | $AP^{OKS=.75}$ % | AR % | $AR^{OKS=.50}$ % | $AR^{OKS=.75}$ % | F1-Score % | Model Size M | Inference Speed FPS | Total Times |
|---|---|---|---|---|---|---|---|---|---|---|
| HRNet-w32 | 42.4 ± 0.25 | **71.1 ± 0.70** | 45.1 ± 1.36 | **53.1 ± 0.26** | **76.7 ± 1** | 55.5 ± 1.15 | 47.1 ± 0.21 | 28.54 | 9.39 | 9.99 |
| HRNet-w32 + UDP | **46.3 ± 0.14** | 70.6 ± 1.93 | **46.0 ± 1.45** | 53.1 ± 1.16 | 74.7 ± 1.73 | 54 ± 0.86 | **49.5 ± 0.66** | 28.54 | 7.32 | 12.83 |
| ResNet-50 Simple Baseline | 38.8 ± 0.7 | 66.8 ± 0.95 | 40.8 ± 2.67 | 52.4 ± 1.30 | 75.8 ± 0.58 | **58.6 ± 3.10** | 44.6 ± 0.95 | 34.0 | 27.46 | 3.42 |
| kMobileNetV3-Small (Proposed) | 33.6 ± 0.75 | 59.8 ± 1.60 | 34.4 ± 2.65 | 48.3 ± 1.16 | 72.7 ± 1 | 52.5 ± 1.73 | 39.6 ± 0.95 | **3.36** | **42.22** | **2.20** |
| kMobileNetV3-Small + UDP (Proposed) | 36.8 ± 1.04 | 67.0 ± 1.19 | 36.8 ± 1.70 | 47.1 ± 1.01 | 73.7 ± 1.53 | 48.5 ± 1.15 | 41.3 ± 1.02 | **3.36** | 32.62 | 2.86 |
| kMobileNetV3-Large (Proposed) | 37.8 ± 0.85 | 67.9 ± 1.10 | 36.1 ± 1.02 | 49.2 ± 0.62 | 73.7 ± 0.57 | 50.5 ± 1 | 42.7 ± 0.73 | 5.23 | 37.59 | 2.49 |
| kMobileNetV3-Large + UDP (Proposed) | 38.4 ± 0.21 | 62.0 ± 1.85 | 44.5 ± 1.83 | 50.4 ± 0.40 | 70.7 ± 1.53 | 56.6 ± 2.05 | 43.5 ± 0.25 | 5.23 | 28.19 | 3.32 |

**Table 7**
Experimental Results of various top-down pose estimation networks (HRNet-w32, HRNet-w32 + UDP, Resnet-50 Simple Baseline, and Proposed Approach (kMobileNetV3 and kMobileNetV3 + UDP)) on PLDU Test Set (k = 3) with 120 Images.

| Network | AP % | $AP^{OKS=.50}$ % | $AP^{OKS=.75}$ % | AR % | $AR^{OKS=.50}$ % | $AR^{OKS=.75}$ % | F1-Score % | Model Size M | Inference Speed FPS | Total Times |
|---|---|---|---|---|---|---|---|---|---|---|
| HRNet-w32 | 57.9 ± 1.40 | 89.3 ± 0.86 | 59.4 ± | 70.1 ± 1.05 | 92.0 ± 0.5 | 74.6 ± 1.85 | 63.43 ± 1.26 | 28.54 | 9.37 | 33.92 |
| HRNet-w32 + UDP | **61.3 ± 0.49** | **91.0 ± 1.47** | 64.4 ± 3.42 | **72.1 ± 1.20** | **92.6 ± 1.53** | 76.5 ± 1.88 | **66.20 ± 1.38** | 28.54 | 7.18 | 44.26 |
| ResNet-50 Simple Baseline | 55.2 ± 1.05 | 88.9 ± 1.04 | 56.5 ± 0.70 | 67.5 ± 0.80 | 91.6 ± 0.6 | 71.5 ± 0.52 | 60.70 ± 0.98 | 34.0 | 27.42 | 11.59 |
| kMobileNetV3-Small (Proposed) | 56.5 ± 1.06 | 88.2 ± 1.13 | 62.5 ± 3.60 | 68.2 ± 0.91 | 90.7 ± 0.84 | 75.9 ± 2.89 | 61.70 ± 0.91 | **3.36** | **42.15** | **7.53** |
| kMobileNetV3-Small + UDP (Proposed) | 56.8 ± 0.15 | 89.2 ± 1.18 | 59.5 ± 1.36 | 67.8 ± 0.87 | 90.4 ± 0.86 | 73.7 ± 1.26 | 61.81 ± 0.46 | **3.36** | 31.56 | 10.07 |
| kMobileNetV3-Large (Proposed) | 56.0 ± 1.11 | 88.6 ± 0.78 | 56.5 ± 1.61 | 67.0 ± 0.85 | 90.4 ± 0.45 | 71.5 ± 0.86 | 60.9 ± 0.95 | 5.23 | 36.36 | 8.73 |
| kMobileNetV3-Large + UDP (Proposed) | 59.5 ± 0.90 | 90.7 ± 2.85 | **65.5 ± 2.32** | 69.9 ± 1 | 92.0 ± 2.36 | **76.8 ± 2** | 64.3 ± 0.97 | 5.23 | 28.48 | 11.15 |

**Table 8**
Experimental Results of various top-down pose estimation networks (HRNet-w32, HRNet-w32 + UDP, Resnet-50 Simple Baseline, and Proposed Approach (kMobileNetV3 and kMobileNetV3 + UDP)) on PLDM Test Set (k = 3) with 50 Images.

| Network | AP % | $AP^{OKS=.50}$ % | $AP^{OKS=.75}$ % | AR % | $AR^{OKS=.50}$ % | $AR^{OKS=.75}$ % | F1-Score % | Model Size M | Inference Speed FPS | Total Times |
|---|---|---|---|---|---|---|---|---|---|---|
| HRNet-w32 | 49.0 ± 0.75 | 71.5 ± 1.65 | 54.7 ± 1.40 | 58.8 ± 0.20 | 75.8 ± 1.05 | 63.6 ± 0.57 | 53.4 ± 0.40 | 28.54 | 9.38 | 10.04 |
| HRNet-w32 + UDP | **57.8 ± 1.13** | 72.8 ± 2.45 | **62.2** ± 1.05 | **65.4** ± 1.70 | 76.8 ± 2.05 | **68.7 ± 1.53** | **61.3 ± 1.34** | 28.54 | 7.10 | 13.23 |
| ResNet-50 Simple Baseline | 44.9 ± 1.33 | **73.4 ± 3.21** | 49.1 ± 1.71 | 56.6 ± 0.61 | **77.8 ± 1.53** | 61.6 ± 0.57 | 50.0 ± 0.98 | 34.0 | 27.84 | 3.37 |
| kMobileNetV3-Small (Proposed) | 38.4 ± 0.61 | 65 ± 0.45 | 38.3 ± 0.78 | 50.4 ± 1.02 | 73.7 ± 1 | 54.5 ± 1.05 | 43.6 ± 0.79 | **3.36** | **42.88** | **2.17** |
| kMobileNetV3-Small + UDP (Proposed) | 47.5 ± 1.15 | 68.2 ± 3.51 | 55.4 ± 3.44 | 56.9 ± 1.58 | 74.7 ± 1.57 | 63.6 ± 3 | 51.0 ± 0.90 | **3.36** | 33.69 | 2.81 |
| kMobileNetV3-Large (Proposed) | 47.1 ± 0.1 | 73.2 ± 1.76 | 52.2 ± 1.49 | 57.3 ± 0.17 | 76.8 ± 1.21 | 64.6 ± 0.57 | 51.7 ± 0.15 | 5.23 | 36.97 | 2.51 |
| kMobileNetV3-Large + UDP (Proposed) | 49.3 ± 0.83 | 70.4 ± 1.45 | 54.2 ± 1.60 | 57.0 ± 0.93 | 74.7 ± 1.57 | 61.6 ± 1 | 52.8 ± 0.55 | 5.23 | 29.22 | 3.19 |

**Table 9**
Experimental Results of various top-down pose estimation networks (HRNet-w32, HRNet-w32 + UDP, Resnet-50 Simple Baseline, and Proposed Approach (kMobileNetV3 and kMobileNetV3 + UDP)) on Mendeley PL Test Set (k = 3) with 40 Images.

| Network | AP % | $AP^{OKS=.50}$ % | $AP^{OKS=.75}$ % | AR % | $AR^{OKS=.50}$ % | $AR^{OKS=.75}$ % | F1-Score % | Model Size M | Inference Speed FPS | Total Times |
|---|---|---|---|---|---|---|---|---|---|---|
| HRNet-w32 | 83.7 ± 4.76 | **97.0 ± 0** | 95.0 ± 2.68 | 87.8 ± 3.37 | 97.0 ± 0.46 | 95.0 ± 2.13 | 85.7 ± 4.11 | 28.54 | 9.40 | 4.33 |
| HRNet-w32 + UDP | **87.3 ± 1.04** | **97.0 ± 0** | 97.0 ± 3.34 | **90.2 ± 0.58** | 97.0 ± 0.46 | 97.0 ± 2.28 | **88.72 ± 0.8** | 28.54 | 7.19 | 5.65 |
| ResNet-50 Simple Baseline | 82.4 ± 1.08 | **97.0 ± 0** | 90.0 ± 0.36 | 86.5 ± 0.35 | **97.8 ± 0** | 91.3 ± 0 | 84.40 ± 0.77 | 34.0 | 27.10 | 1.50 |
| kMobileNetV3-Small (Proposed) | 78.6 ± 3.30 | 97.0 ± 1.15 | 90.6 ± 3.55 | 83.5 ± 2.88 | 97.0 ± 1.05 | 91.3 ± 3.30 | 80.97 ± 3.11 | **3.36** | **42.24** | **0.97** |
| kMobileNetV3-Small + UDP (Proposed) | 80.7 ± 1.48 | **97.0 ± 0** | 92.9 ± 0.21 | 85.2 ± 0.98 | 97.0 ± 0.46 | 93.5 ± 0 | 82.88 ± 1.23 | **3.36** | 33.93 | 1.19 |
| kMobileNetV3-Large (Proposed) | 83.1 ± 0.1 | 97.0 ± 0.15 | 93.0 ± 0.95 | 87.2 ± 0.28 | **97.8 ± 0** | 93.5 ± 0 | 85.10 ± 0.15 | 5.23 | 36.79 | 1.09 |
| kMobileNetV3-Large + UDP (Proposed) | 84.0 ± 0.26 | 97.0 ± 0.05 | **97.0 ± 0.05** | 88.0 ± 0.2 | **97.8 ± 0** | **97.8 ± 0** | 86 ± 0.2 | 5.23 | 28.88 | 1.40 |

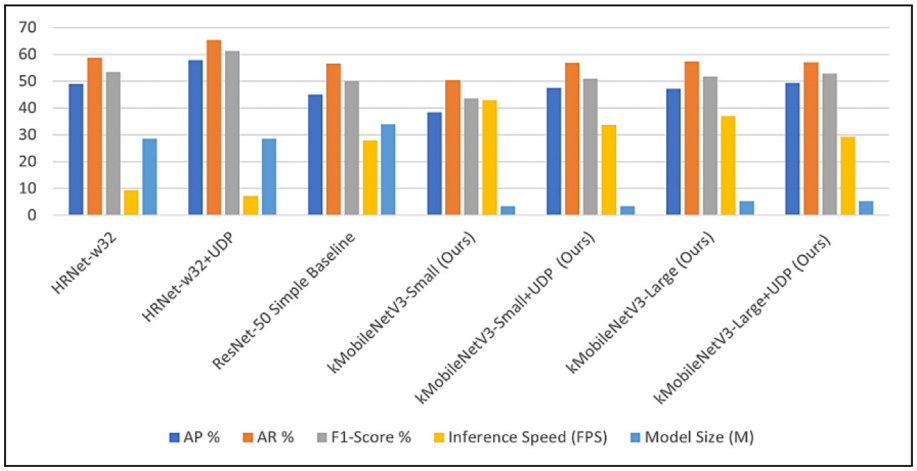


**Fig. 10.** Comparative analysis of various top-down pose estimation networks (HRNet-w32, HRNet-w32 + UDP, Resnet-50 Simple Baseline, and Proposed Approach (kMobileNetV3-Small, kMobileNetV3-Small + UDP, kMobileNetV3-Large and kMobileNetV3-Large + UDP)) in terms of tradeoff between the characteristic evaluation parameters on PLDU test dataset with 120 images.

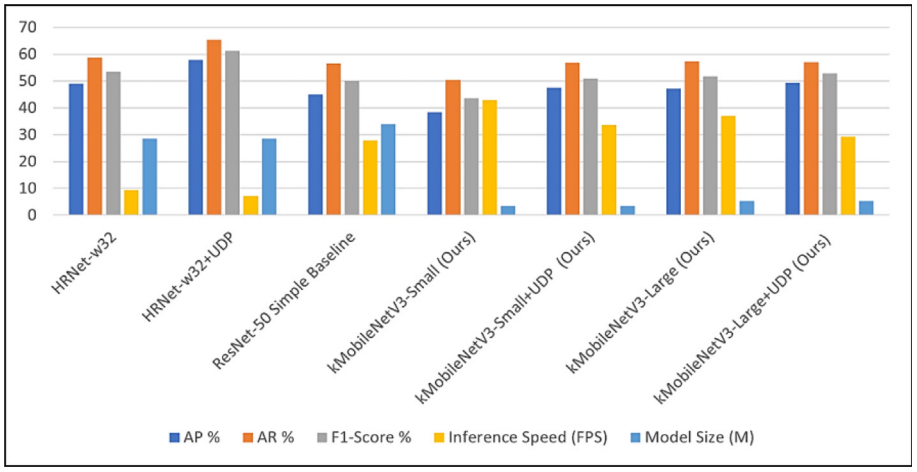


**Fig. 11.** Comparative analysis of various top-down pose estimation networks (HRNet-w32, HRNet-w32 + UDP, Resnet-50 Simple Baseline, and Proposed Approach (kMobileNetV3-Small, kMobileNetV3-Small + UDP, kMobileNetV3-Large and kMobileNetV3-Large + UDP)) in terms of tradeoff between the characteristic evaluation parameters on PLDM test dataset with 50 images.

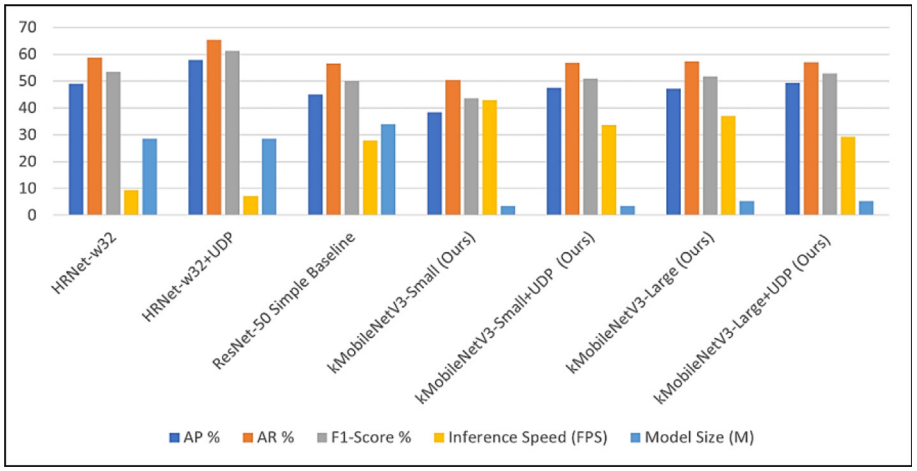


**Fig. 12.** Comparative analysis of various top-down pose estimation networks (HRNet-w32, HRNet-w32 + UDP, Resnet-50 Simple Baseline, and Proposed Approach (kMobileNetV3-Small, kMobileNetV3-Small + UDP, kMobileNetV3-Large and kMobileNetV3-Large + UDP)) in terms of tradeoff between the characteristic evaluation parameters on Mendeley PL test dataset with 40 images.

**Table 10**
Experimental Results of various top-down pose estimation networks (HRNet-w32, HRNet-w32 + UDP, Resnet-50 Simple Baseline, and Proposed Approach (kMobileNetV3 and kMobileNetV3 + UDP)) on PLD Test Set (k = 3) with 170 Images.

| Network | AP % | $AP^{OKS=.50}$ % | $AP^{OKS=.75}$ % | AR % | $AR^{OKS=.50}$ % | $AR^{OKS=.75}$ % | F1-Score % | Model Size M | Inference Speed FPS | Total Times |
|---|---|---|---|---|---|---|---|---|---|---|
| HRNet-w32 | 55.7 ± 1.09 | 84.3 ± 0.66 | 57.8 ± 1.15 | 67.2 ± 1.10 | 87.4 ± 0.7 | 71.6 ± 1.27 | 60.90 ± 1.11 | 28.54 | 9.22 | 45.21 |
| HRNet-w32 + UDP | **58.9 ± 0.50** | **86.8 ± 0.58** | **60.0 ± 1.07** | **70.2 ± 0.11** | **89.6 ± 0.11** | **73.2 ± 0.40** | **64.02 ± 0.31** | 28.54 | 7.14 | 58.41 |
| ResNet-50 Simple Baseline | 53.4 ± 0.55 | 85.3 ± 0.75 | 55.9 ± 1.31 | 65.8 ± 0.75 | 88.9 ± 0.62 | 71.6 ± 1.34 | 58.9 ± 0.65 | 34.0 | 27.60 | 15.11 |
| kMobileNetV3-Small (Proposed) | 48.3 ± 0.45 | 82.8 ± 1.12 | 48.8 ± 0.90 | 60.9 ± 0.5 | 86.0 ± 0.69 | 65.9 ± 0.96 | 53.8 ± 0.40 | **3.36** | **42.71** | **9.76** |
| kMobileNetV3-Small + UDP (Proposed) | 54.6 ± 0.58 | 84.0 ± 0.89 | 58.9 ± 1.73 | 66.2 ± 0.76 | 86.7 ± 0.7 | 72.3 ± 1.32 | 59.8 ± 0.60 | **3.36** | 31.99 | 13.03 |
| kMobileNetV3-Large (Proposed) | 52.2 ± 0.23 | 85.2 ± 0.55 | 53.1 ± 0.49 | 64.5 ± 0.35 | 87.7 ± 0.62 | 68.7 ± 0.40 | 57.7 ± 0.32 | 5.23 | 37.23 | 11.19 |
| kMobileNetV3-Large + UDP (Proposed) | 56.5 ± 0.72 | 85.0 ± 0.26 | **59.9 ± 0.85** | 67.7 ± 0.65 | 87.4 ± 0.25 | 73.0 ± 0.85 | 61.56 ± 0.69 | 5.23 | 29.10 | 14.32 |

**Table 11**
Performance in average precision (AP) of Various Key Point Detectors on PLD Test Set with 170 Images.

| S.# | Model | Key Points | AP (%) |
|---|---|---|---|
| 1 | SSD512 (Liu et al., 2016) | 5 | 30.04 |
| 2 | YOLOV3 (Redmon & Farhadi, 2018) | | 15.36 |
| 3 | RFBNet (S. Liu & Huang, 2018) | | 32.34 |
| 4 | FSSD (Zuoxin Li & Zhou, 2017) | | 40.80 |
| 5 | RetinaNet (Lin et al., 2017) | | 32.87 |
| 6 | RefineDet (Zhang et al., 2018) | | 26.13 |
| 7 | CenterNet (Duan et al., 2019) | | 52.71 |
| 8 | CNNCDM (Dai et al., 2020) | | 54.80 |
| 9 | **kMobileNetV3-Small (Proposed)** | **3** | 48.30 |
| 10 | **kMobileNetV3-Small + UDP (Proposed)** | | 54.60 |
| 11 | **kMobileNetV3-Large (Proposed)** | | 52.20 |
| 12 | **kMobileNetV3-Large + UDP (Proposed)** | | **56.50** |

**Table 12**
Processing Time in FPS of Various Key Point Detectors on PLD Test Set with 170 Images.

| S.# | Model | FPS |
|---|---|---|
| 1 | SSD512 (Liu et al., 2016) | 17.15 |
| 2 | YOLOV3 (Redmon & Farhadi, 2018) | 19.26 |
| 3 | RFBNet (Liu & Huang, 2018) | 14.48 |
| 4 | FSSD (Li & Zhou, 2017) | 15.91 |
| 5 | RetinaNet (Lin et al., 2017) | 15.36 |
| 6 | RefineDet (Zhang et al., 2018) | 14.35 |
| 7 | CenterNet (Duan et al., 2019) | 18.97 |
| 8 | CNNCDM (Dai et al., 2020) | 16.21 |
| 9 | **kMobileNetV3-Small (Proposed)** | **42.71** |
| 10 | **kMobileNetV3-Small + UDP (Proposed)** | 31.99 |
| 11 | **kMobileNetV3-Large (Proposed)** | 37.23 |
| 12 | **kMobileNetV3-Large + UDP (Proposed)** | 29.10 |

## 5.4. Discussion

### 5.4.1. Comparative analysis of proposed annotation protocols

It can be seen from the results depicted in Tables 5 and 7 that the annotation protocol 2 (k = 3) increases the precision results of all the top-down pose estimation networks by 18–20% and recall results by 16–18% while maintaining approximately equivalent inference speeds for all the networks on the PLDU dataset. For the PLDM dataset, annotation protocol 2 (k = 3) increases the precision values by 4–12% and recall values by 2–12% in comparison to the annotation protocol 1 (k = 6) as shown in Table 6 and 8, while maintaining approximately equivalent inference speeds for all the networks. It is also noteworthy that the UDP networks (HRNet-w32 + UDP, kMobileNetV3-Small + UDP and kMobileNetV3-Large + UDP) show the highest improvements in terms of precision and recall values for both the PLDU and PLDM datasets for annotation protocol 2 (k = 3) in comparison to the annotation protocol 1 (k = 6). These results imply that the investigation of the annotation protocol in top-down key point pose estimation networks is a significant but often neglected aspect. The performance gain in the results of annotation protocol 2 (k = 3) in comparison to the annotation protocol 1 (k = 6) is because the OKS calculation (ref. Eq (7)) divides the sum of negative exponential functions, of Euclidean distances between the ground-truth and the predicted key points, for all the key points by the total number of key points. For similar numerator values, dividing by a larger number (k = 6 in annotation protocol 1) instead of a smaller one (k = 3 in annotation protocol 2) would yield lower OKS results and vice-versa. We also believe that the performance gain is because the annotation protocol 2 (k = 3) utilizes only 3 output channels to predict 3 key points while annotation protocol 1 (k = 6) requires double the channels for predicting 6 key points. Increasing the number of key points will lead to a performance drop because the model needs more capacity to learn more things. Hence, annotation protocol 2 (k = 3) is identified as the optimal protocol for labelling the PL key points and is employed to label the key points for the third Mendeley PL dataset.

### 5.4.2. Comparative analysis of proposed kMobileNetV3 and kMobileNetV3 + UDP with various State-of-the-art Top-Down pose estimation networks

The results in Table 7, 8 and 9 depict that our proposed kMobileNetV3 and kMobileNetV3 + UDP (in small and large versions) top-down pose estimation networks outperform all the other top-down pose estimation networks (HRNet-w32, HRNet-w32 + UDP and Resnet-50 Simple Baseline) in terms of least model size, the highest inference speed and the highest total inference time for PLDU, PLDM and Mendeley datasets respectively at the cost of little to no degradation in the precision (AP), recall (AR) and F1-Score values. Figs. 10-12 depict these results graphically by highlighting the trade-off between the mentioned characteristic evaluation parameters for the PLDU, PLDM and Mendeley datasets respectively. Overall HRNet-w32 + UDP network yields the best AP, AR, and F1-Score values for PLDU (Table 7), PLDM (Table 8) and Mendeley PL (Table 9) datasets but it suffers from high model complexity (28.54 M) and painfully low inference speeds (∼7 FPS). In comparison to HRNet-w32 and its UDP variant, Resnet-50 Simple Baseline model yields better inference speeds (∼27 FPS) but suffers a minor degradation in the range of ∼ 2–4% in the AP and AR values and comprises of even a bigger model size (34.0 M parameters). However, even with a larger backbone network, our proposed kMobileNetV3-Large and kMobileNetV3-Large + UDP yield a model size of only 5.23 M parameters and an inference speed in the range of 28–37 FPS which is better than the HRNet-w32, HRNet-w32 + UDP and Resnet50 Simple Baseline networks. As far as AP, AR and F1-Score values are concerned, our proposed

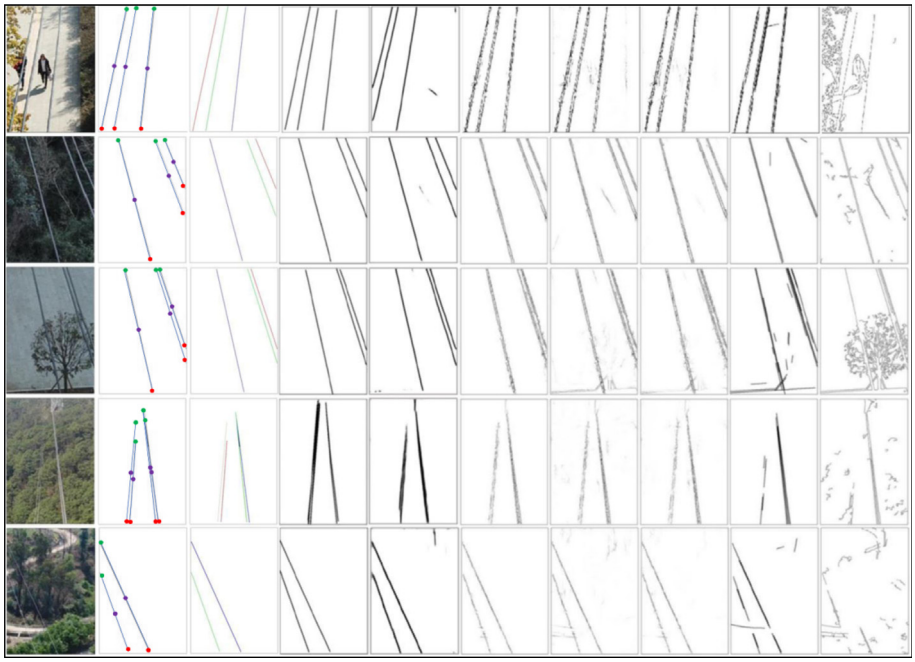

**Fig. 13.** Results on some sample images from PLD dataset (From left to right: original image, proposed kMobileNetV3-Large + UDP model with the three key points: S (green), C (purple), E (red), modified PINet model (Sumagayan et al., 2021), CNNCDM, BDCN, CFSC, RCF, HED, Gestalt Grouping and Canny). Our model can detect the power lines using the connections between three key points only. The connections between the key points are estimated directly as pose of the PL from our proposed model.

kMobileNetV3-Large + UDP suffers a minor degradation of ∼ 2%, ∼7% and ∼ 3% in evaluation parameter values for PLDU (Table 7), PLDM (Table 8) and Mendeley (Table 9) datasets respectively. For the PLDU, PLDM and Mendeley datasets, our proposed kMobileNetV3-Large + UDP yields equivalent AP, AR, and F1-Score values as HRNet-w32 with a much lower model size (5.23 M compared to 28.54 M) and a higher inference speed (∼29 FPS compared to ∼ 9 FPS). Our proposed kMobileNetV3-Large + UDP surpasses the Resnet-50 Simple Baseline network in terms of all evaluation parameters on all the three datasets.

Amongst our proposed range of models, kMobileNetV3-Small yields the least model size (3.36 M), best inference speed (∼42 FPS) and the subsequent least total inference time on all the three datasets. However, kMobileNetV3-Small lags behind its larger counterpart (kMobileNetV3-Large) in terms of AP, AR and F1-Scores due to a smaller feature extraction backbone structure (ref. Table 2 and 3). Hence, keeping in mind all the evaluation parameters (ref. Figs. 10-12), we conclude that our proposed kMobileNetV3-Large + UDP is the most optimal top-down pose estimation network model for PL detection due to higher AP, AR, and F1-Scores with lower model size and higher inference speeds and is therefore, adopted as the main PL detection network in our proposed PLPose framework. We tested the processing speeds of all the models on the same Nvidia Tesla K80 GPU that was utilized for training these networks.

The overall precision and recall values for all the datasets are much higher for a lower and more flexible OKS threshold of 0.50 as compared to the more restricted OKS = 0.75 and OKS = 0.5–0.95. It should also be noted that amongst the three datasets, the evaluation results for the Mendeley PL dataset are the highest, followed by PLDU and then the PLDM datasets. This is because the PL images in PLDU and PLDM datasets are more complex and cluttered as compared to those in the Mendeley PL dataset.

### 5.4.3. Effect of unbiased data processing (UDP) module

It can be observed from the results in Tables 5-10 that the UDP variants of the networks (HRNet-w32 + UDP, kMobileNetV3-Small + UDP and kMobileNetV3-Large + UDP) help to increase the AP, AR, and F1-Score values, on all the three datasets and for both the annotation protocols (k = 6 and k = 3), in comparison to their vanilla counterparts (HRNet-w32, kMobileNetV3-Small and kMobileNetV3-Large). On average, an increment of 3–4% is observed in the results of all the networks that employ a UDP module for PL detection. This gain is the result of semantic alignment by UDP, between the input PL key points and the predicted output key points, during the data transformation and inference phases of PL detection via top-down pose estimation. This semantic alignment is essential because the standard top-down pose estimation networks produce unaligned results, by employing pixel as a means of image measurement, when performing the de facto standard flipping strategy during inference (Sun et al., 2019; Xiao et al., 2018). In this work, PL detection via conventional top-down pose estimation networks shifts the output heatmap by 1 pixel (following the works in (Sun et al., 2019; Xiao et al., 2018)) to compensate these unaligned results between the input and the predicted PL key points. This compensation requires additional post-processing and degrades the accuracy of PL predictions. Since the UDP module is based on unit-length measurement of images instead of pixels (J. Huang et al., 2020), it successfully handles the unaligned PL key points without any heatmap shifting, when deployed for the conventional top-down pose estimation networks. However, it should be noted that the deployment of UDP module to the top-down pose estimation networks reduces their inference speeds due to the associated data processing.

### 5.4.4. Comparative analysis of PLPose with various State-of-the-art key point detectors for PL detection

The results of our proposed PLPose, comprising of annotation protocol (k = 3) and the best performing top-down pose estimation network (kMobileNetV3-Large + UDP), are compared with the state-of-the-art PL key point detectors from (Dai et al., 2020) on the combined PLD dataset, in Table 11. The reported results from state-of-the-art PL detectors in Table 11 are selected at threshold 16 (ref. Table 6 in (Dai et al., 2020)). Table 11 shows that our

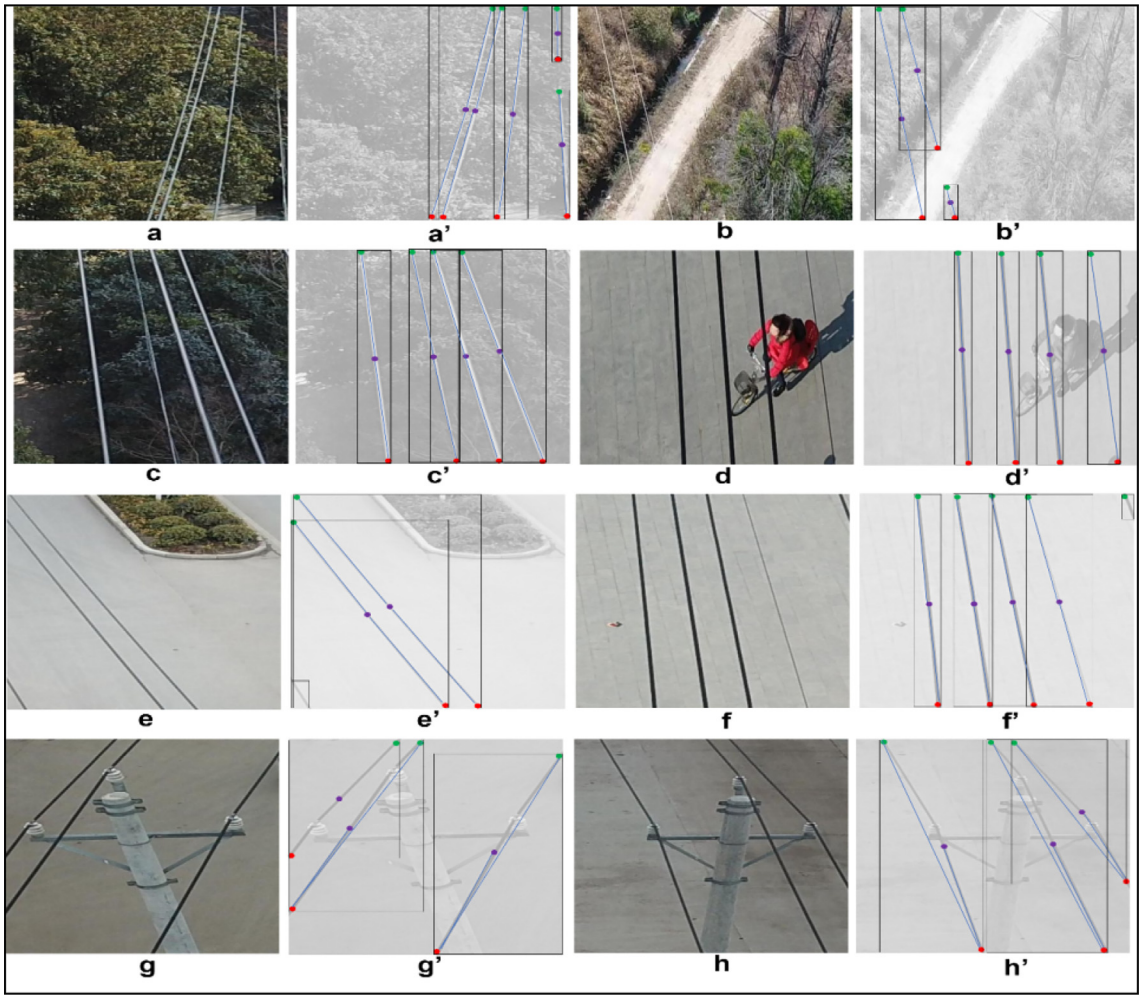


**Fig. 14.** Success (top two rows) and failure (bottom two rows) cases of PLPose on some sample images from PLD dataset. Success modes: Row 1: occlusion handling (a)-(b) by treating non-occluded PL parts as separate PL instances, Row 2: effective multiple PL detection in diverse backgrounds (c)-(d); Failure modes: Row 3: No or missing key point detections of short-length PLs at image corners (e)-(f), Row 4: Incorrect center point detection and pose estimation for curved PLs (g)-(h). All the three key point (S (green), C (purple), E (red)) detection results are depicted on washed out images for better visualization.

proposed annotation scheme with only three key points and the top-down pose estimation network (kMobileNetV3-Large + UDP) achieves a precision of 56.50% in comparison to the top-performing CNNCDM network which uses five key points and an encoder-decoder style network to achieve a precision of 54.80%. Even higher precision values (>56.50%) can be achieved by our proposed detection model for less strict OKS thresholds of OKS = 0.5 and OKS = 0.75 as depicted in Table 10. It should be noted that the AP values for CNNCDM are also calculated using a Euclidean distance-based matching method, to compare the ground-truth and detected key points, like our OKS evaluation metric. Hence, the results from the two methods are comparable. As far as the processing speed is concerned, Table 12 clearly depicts that our proposed model and its variants outperform all the other PL key point detectors on the PLD dataset. The processing speed results of other models are copied from CNNCDM paper and are obtained using Nvidia GTX 1070 GPU which is more powerful than the Nvidia Tesla K80 GPU utilized for our experiments. Even with a less powerful GPU, our proposed kMobileNetV3-Small model achieves the highest processing speed of 42.71 FPS in comparison to the top-performing method YOLOv3 which has a processing speed of 20.60 FPS (a gain of $\sim$ 22%). Moreover, our least performing model kMobileNetV3-Large + UDP also achieves a processing speed of 29.10 FPS which is higher than the processing speed of YOLOv3 (a gain of $\sim$ 9%) which confirms the effectiveness of our proposed models.

*5.4.5. Qualitative assessment of proposed PLPose with various State-of-the-art key point detectors for PL detection*

Fig. 13 shows a qualitative assessment of the proposed PLPose in comparison to other PL detectors from (Dai et al., 2020; Sumagayan et al., 2021). It should be noted that all of these methods except the CNNCDM (Dai et al., 2020) and PINet (Sumagayan et al., 2021) produce resulting images of predicted edges. Meanwhile, our proposed method predicts the key points (S (green), C (purple), E (red)) similar to CNNCDM and PINet. However, being a pose estimation network, the proposed method predicts the connections between the key points directly as pose, unlike the CNNCDM and PINet which employ additional curve-fitting methods to group the detected key points. Lastly, our proposed annotation scheme that employs only three representative key points is more efficient than the annotation scheme used in CNNCDM (five key points) and PINet (dense key points).

In this manner, our proposed PLPose significantly advances the state-of-the-art on various PL benchmark datasets and is hence, a suitable framework for detecting PLs via top-down key point pose estimation method.

*5.4.6. Qualitative assessment of failure and success modes*

Some examples of complex images from the PLD dataset where our best model variant (kMobileNetV3-Large + UDP) does and does not work well in detecting PLs are depicted in Fig. 14. The top two rows in Fig. 14 highlight the success cases while the bottom two rows highlight the failure ones. The images are labelled via lowercase letters and their corresponding detections by complemented lowercase letters. All the detection results (key point: S (green), C (purple), E (red), pose and a supplementary bounding box) are depicted on washed out images for better visualization. The visualization of the bounding boxes is optional in our proposed PLPose. The first row in Fig. 14 reveals the occlusion case where parts of the PLs are occluded in the middle (a, b) and proposed model handles this anomaly by detecting the non-occluded PL parts (a', b') as separate PL instances. The second row depicts that multiple PLs under diverse and complex backgrounds: (c) trees and (d) human are detected perfectly (c', d') with the three key points and pose connections between them. The third row in Fig. 14 highlights a failure case involving the detection of short-length corner PLs (e, f). The short-length corner PLs (e, f) are detected by our PLPose via a bounding box in Fig. 14 (e') and a bounding box and an additional key point (S) in Fig. 14 (f') without the other key points and their corresponding pose connections. Lastly, the proposed PLPose also fails in effectively detecting the curved PLs depicted in Fig. 14 (g, h). In the case of curved PLs, the center point C, was annotated to highlight the curve in the proposed annotation scheme (ref. Fig. 9). However, these center points are not detected correctly at the point of curve (Fig. 14 (g'), (h)') and are instead localized at the actual centers of the PLs. We believe that this might be due to the scarcity of the curved PL samples in the benchmark datasets and might improve otherwise. Moreover, the pose connections in Fig. 14 (g'), (h)' are also not inferred correctly with the E key point being connected to the S key point directly without any connection between the C and S key points. The efficient handling of these failure cases is left as future work.

## 6. Conclusion

In this study, we framed PL detection as a key points-based pose estimation task. We introduced PLPose, an end-to-end processing and detecting framework for PL key points pose estimation. For the PL key points detection within PLPose, we first introduced an efficient key points annotation protocol with three key points only and then adapted the MobileNetV3 CNN for top-down key points pose estimation by adding a simple key point head with dconv layers for predicting the key point heatmaps and named it as kMobileNetV3. kMobileNetV3 was introduced in two versions: kMobileNetV3-Small and kMobileNetV3-Large to investigate the effect of feature extraction backbone size on the model performance. We also extended the proposed kMobileNetV3 by combining it with the unbiased data processing (UDP) module for better detection performance, which yielded a novel data-centric top-down pose estimation model. It was found that kMobileNetV3-La rge + UDP yielded the most optimal results (high AP, AR, F1-Scores, Inference Speed and low model size and total inference time) amongst all other variants (kMobileNetV3-Small, kMobileNetV3-Small + UDP, kMobileNetV3-Large) of the proposed kMobileNetV3. The models were evaluated on three public PL datasets (PLDU, PLDM and Mendeley PL). Our proposed network (kMobileNetV3-Large + UDP) outperformed the HRNet-w32 and Resnet-50 Simple Baseline state-of-the-art top-down pose estimation networks in terms of all evaluation parameters on all the three PL datasets. However, the proposed kMobileNetV3-Large + UDP lagged the HRNet-w32 + UDP model in terms of AP, AR, and F1-Scores on 2 out of 3 datasets but successfully surpassed the HRNet-w32 + UDP in terms of model size and inference speed on all the three datasets. Therefore, kMobileNetV3-Large + UDP was chosen to be the most optimal top-down pose estimation network for our proposed PLPose framework. Moreover, our proposed annotation protocol with only 3 key points also surpassed other annotation protocols for key points-based PL detection by yielding better precision and processing time results. This implied that the examination of the annotation protocol is an important but often underestimated step in the top-down key point pose estimation networks.

Limitations and future directions

Currently, the proposed PLPose cannot distinguish and segment the individual PL instances. Moreover, kMobileNetV3-Large + UDP, the top-down pose estimation network of PLPose, currently lags behind HRNet-w32 + UDP in terms of AP, AR, and F1-Scores. All these improvements along with the handling of the mentioned failure cases will be made as part of our future work. The investigation of bottom up HPE detectors for PL detection and the effect of several pre- and post-processing techniques will also be carried out in future.

Lastly, it is important to highlight that all the results discussed in this study were presented on benchmark PL datasets comprising of aerial images. However, for any computer vision application, the performance of the detection models can vary greatly with the scale and altitude of image data as in the case of object detection from high-resolution UAV images (Zhang et al., 2021; Zhang et al., 2020). Hence, a study of the proposed detection framework on PL images acquired at a much higher or much lower altitudes with varying resolutions than those presented in this work will also be an interesting future direction.

## CRediT authorship contribution statement

**Rabeea Jaffari:** Conceptualization, Investigation, Formal analysis, Validation, Writing – original draft. **Manzoor Ahmed Hashmani:** Conceptualization, Data curation, Supervision. **Constantino Carlos Reyes-Aldasoro:** Supervision. **Aisha Zahid Junejo:** . **Hasmi Taib:** Conceptualization, Data curation. **M. Nasir B. Abdullah:** Conceptualization, Data curation.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

The authors would like to extend their deepest gratitude to Universiti Teknologi PETRONAS and Yayasan UTP (UTP) – Grant number: 015LC0-332 for provision of materials and resources to carry out this research work. The authors would also like to express their gratitude to the owners of mmpose framework for idea regarding the kMobileNet.

## Appendix A. Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jksuci.2023.101615.

## References

Abdelfattah, R., Wang, X., Wang, S., 2022. PLGAN: Generative Adversarial Networks for Power-Line Segmentation in Aerial Images. arXiv preprint arXiv:2204.07243.

Amadi, H., Okafor, E., 2015. Analysis of methodologies for the evaluation of power outage costs. Int. J. Eng. Res. Technol. 4 (5), 956.
Avizonis, P., Barron, B., 1999. Low cost wire detection system. In: Paper presented at the Gateway to the New Millennium. 18th Digital Avionics Systems Conference. Proceedings (Cat. No. 99CH37033).
Baker, N., Lu, H., Erlikhman, G., Kellman, P.J., 2018. Deep convolutional networks do not classify based on global object shape. PLoS Comput. Biol. 14 (12), e1006613.
Bhola, R., Krishna, N.H., Ramesh, K., Senthilnath, J., Anand, G., 2018. Detection of the power lines in UAV remote sensed images using spectral-spatial methods. J. Environ. Manage. 206, 1233–1242.
Candamo, J., Kasturi, R., Goldgof, D., Sarkar, S., 2006. Vision-based on-board collision avoidance system for aircraft navigation. In: Paper presented at the Unmanned Systems Technology VIII.
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 Trans. Aerosp. Electron. Syst. 45 (3), 937–949.
Canny, J., 1986. A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. 6, 679–698.
Ceron, A., Prieto, F., 2014. Power line detection using a circle based search with UAV images. In: Paper presented at the 2014 International Conference on Unmanned Aircraft Systems (ICUAS).
Chen, Y., Tian, Y., He, M., 2020. Monocular human pose estimation: a survey of deep learning-based methods. Comput. Vis. Image Underst. 192, 102897.
Choi, H., Koo, G., Kim, B. J., Kim, S.W., 2019. Real-time Power Line Detection Network using Visible Light and Infrared Images. In: Paper presented at the 2019 International Conference on Image and Vision Computing New Zealand (IVCNZ).
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. Syst.. Appl. 165, 113895.
COCO, M., 2016. COCO Keypoint Evaluation. Retrieved from https://cocodataset.org/#keypoints-eval.
Contributors, M., 2020. MMPose, OpenMMLab Pose Estimation Toolbox and Benchmark. Retrieved from https://github.com/open-mmlab/mmpose.
Dai, Z., Yi, J., Zhang, Y., Zhou, B., He, L., 2020. Fast and accurate cable detection using CNN. Appl. Intell. 50 (12), 4688–4707.
Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q., 2019. Centernet: Keypoint triplets for object detection. In: Paper presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision.
Gao, F., Li, Q., Ji, Y., Ji, S., Guo, J., Sun, H., Wang, N., 2021. EWNet: an early warning classification framework for smart grid based on local-to-global perception. Neurocomputing 443, 199–212.
Ghafoorian, M., Nugteren, C., Baka, N., Booij, O., Hofmann, M., 2018. EL-GAN: embedding loss driven generative adversarial networks for lane detection. In: Paper presented at the Proceedings of the European Conference on Computer Vision (ECCV).
Golightly, I.T., 2006. Visual control of an unmanned aerial vehicle for power line inspection. Bangor University, United Kingdom.
Golightly, I., Jones, D., 2005. Visual control of an unmanned aerial vehicle for power line inspection. In: Paper presented at the ICAR'05. Proceedings, 12th International Conference on Advanced Robotics, 2005.
Gupta, S., Tan, M., 2019. EfficientNet-EdgeTPU: Creating accelerator-optimized neural networks with AutoML. Google AI Blog 2.
Hao, S., Zhou, Y., Guo, Y., 2020. A brief survey on semantic segmentation with deep learning. Neurocomputing 406, 302–321.
He, J., Zhang, S., Yang, M., Shan, Y., Huang, T., 2019. Bi-directional cascade network for perceptual edge detection. In: Paper presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.
Hough, P.V., 1962. Method and means for recognizing complex patterns: Google Patents.
Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Vasudevan, V., 2019. Searching for mobilenetv3. In: Paper presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision.
Huang, L., Chen, C., Yun, J., Sun, Y., Tian, J., Hao, Z., Ma, H., 2022. Multi-scale feature fusion convolutional neural network for indoor small target detection. Front. Neurorob. 16.
Huang, K., Wang, Y., Zhou, Z., Ding, T., Gao, S., Ma, Y., 2018. Learning to parse wireframes in images of man-made environments. In: Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W., 2019. Ccnet: Criss-cross attention for semantic segmentation. In: Paper presented at the Proceedings of the IEEE/CVF international conference on computer vision.
Huang, J., Zhu, Z., Guo, F., & Huang, G., 2020. The devil is in the details: Delving into unbiased data processing for human pose estimation. In: Paper presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.
Jaffari, R., Hashmani, M.A., Reyes-Aldasoro, C.C., Aziz, N., Rizvi, S.S.H., 2021. Deep Learning Object Detection Techniques for Thin Objects in Computer Vision: An Experimental Investigation. In: Paper presented at the 2021 7th International Conference on Control, Automation and Robotics (ICCAR).
Jaffari, R., Hashmani, M.A., Reyes-Aldasoro, C.C., 2021a. A novel focal phi loss for power line segmentation with auxiliary classifier U-Net. Sensors 21 (8), 2803.
Kasturi, R., Camps, O., Huang, Y., Narasimhamurthy, A., Pande, N., 2002. Wire detection algorithms for navigation. NASA Tech Rep.
Li, Z., Zhou, F., 2017. FSSD: feature fusion single shot multibox detector. arXiv preprint arXiv:1712.00960.
Li, Z., Liu, Y., Hayward, R., Zhang, J., Cai, J., 2008. Knowledge-based power line detection for UAV surveillance and inspection systems. In: Paper presented at the 2008 23rd International Conference Image and Vision Computing New Zealand.
Li, X., Li, J., Hu, X., Yang, J., 2019a. Line-CNN: end-to-end traffic line detection with line proposal unit. IEEE Trans. Intell. Transp. Syst.
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. Mach. Vis. Appl. 21 (5), 677–686.
Li, Y., Pan, C., Cao, X., Wu, D., 2018. Power line detection by pyramidal patch classification. IEEE Transactions on Emerging Topics in Computational Intelligence 3 (6), 416–426.
Li, Y., Xiao, Z., Zhen, X., Cao, X., 2019b. Attentional information fusion networks for cross-scene power line detection. IEEE Geosci. Remote Sens. Lett. 16 (10), 1635–1639.
Li, H., Yu, H., Wang, J., Yang, W., Yu, L., Scherer, S., 2021. ULSD: unified line segment detection across pinhole, fisheye, and spherical cameras. ISPRS J. Photogramm. Remote Sens. 178, 187–202.
Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., 2014. Microsoft coco: Common objects in context. In: Paper presented at the European conference on computer vision.
Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal loss for dense object detection. In: Paper presented at the Proceedings of the IEEE international conference on computer vision.
Liu, S., Huang, D., 2018. Receptive field block net for accurate and fast object detection. In: Paper presented at the Proceedings of the European Conference on Computer Vision (ECCV).
Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C., 2016. Ssd: Single shot multibox detector. In: Paper presented at the European conference on computer vision.
Liu, Y., Cheng, M.-M., Hu, X., Wang, K., Bai, X., 2017. Richer convolutional features for edge detection. In: Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.
Madaan, R., Maturana, D., Scherer, S., 2017. Wire detection using synthetic data and dilated convolutional networks for unmanned aerial vehicles. In: Paper presented at the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
Memon, M.M., Hashmani, M.A., Junejo, A.Z., Rizvi, S.S., Arain, A.A., 2021. A novel luminance-based algorithm for classification of semi-dark images. Appl. Sci. 11 (18), 8694.
Miao, X., Liu, X., Chen, J., Zhuang, S., Fan, J., Jiang, H., 2019. Insulator detection in aerial images for transmission line inspection using single shot multibox detector. IEEE Access 7, 9945–9956.
Nguyen, V. N., Jenssen, R., Roverso, D., 2019. Ls-net: Fast single-shot line-segment detector. arXiv preprint arXiv:1912.09532.
Pan, C., Shan, H., Cao, X., Li, X., Wu, D., 2017. Leveraging spatial context disparity for power line detection. Cogn. Comput. 9 (6), 766–779.
Rajaei, B., von Gioi, R.G., 2018. Gestaltic grouping of line segments. Image Process. 8, 37–50.
Redmon, J., Farhadi, A., 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.
Russell, B.D., Benner, C.L., Wischkaemper, J., Jewell, W., McCalley, J., 2007. Reliability based vegetation management through intelligent system monitoring. Texas A&M University, Texas.
Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In: Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.
Santos, T., Moreira, M., Almeida, J., Dias, A., Martins, A., Dinis, J., Silva, E., 2017. PLineD: Vision-based power lines detection for Unmanned Aerial Vehicles. In: Paper presented at the 2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC).
Saurav, S., Gidde, P., Singh, S., Saini, R., 2019. Power line segmentation in aerial images using convolutional neural networks. In: Paper presented at the International Conference on Pattern Recognition and Machine Intelligence.
Shan, H., Zhang, J., Cao, X., Li, X., Wu, D., 2017. Multiple auxiliaries assisted airborne power line detection. IEEE Trans. Ind. Electron. 64 (6), 4810–4819.
Su, H., Deng, J., Fei-Fei, L., 2012. Crowdsourcing annotations for visual object detection. In: Paper presented at the Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence.
Sumagayan, M.U., Premachandra, C., Mangorsi, R.B., Salaan, C.J., Premachandra, H. W.H., Kawanaka, H., 2021. Detecting power lines using point instance network for distribution line inspection. IEEE Access 9, 107998–108008.
Sun, K., Xiao, B., Liu, D., Wang, J., 2019. Deep high-resolution representation learning for human pose estimation. In: Paper presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.
Wang, S., Huang, L., Jiang, D., Sun, Y., Jiang, G., Li, J., Xiong, H., 2022. Improved multi-stream convolutional block attention module for sEMG-based gesture recognition. Front. Bioeng. Biotechnol. 10.
Wang, L., 2016. The fault causes of overhead lines in distribution network. In: Paper presented at the MATEC Web of Conferences.
Wu, Q., An, J., Yang, R., 2010. Extraction of power lines from aerial images based on Hough transform. In: Paper presented at the Earth observing missions and sensors: Development, implementation, and characterization.
Xiao, B., Wu, H., Wei, Y., 2018. Simple baselines for human pose estimation and tracking. In: Paper presented at the Proceedings of the European conference on computer vision (ECCV).

Xie, S., Tu, Z., 2015. Holistically-nested edge detection. In: Paper presented at the Proceedings of the IEEE international conference on computer vision.
Xu, Y., Xu, W., Cheung, D., Tu, Z., 2021. Line segment detection using transformers without edges. In: Paper presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.
Xue, N., Bai, S., Wang, F., Xia, G.-S., Wu, T., Zhang, L., 2019. Learning attraction field representation for robust line segment detection. In: Paper presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.
Xue, N., Wu, T., Bai, S., Wang, F., Xia, G.-S., Zhang, L., Torr, P.H., 2020. Holistically-attracted wireframe parsing. In: Paper presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.
Yan, G., Li, C., Zhou, G., Zhang, W., Li, X., 2007. Automatic extraction of power lines from aerial images. IEEE Geosci. Remote Sens. Lett. 4 (3), 387–391.
Yetgin, Ö.E., Gerek, Ö.N., 2019. Ground Truth of Powerline Dataset (Infrared-IR and Visible Light-VL). Retrieved from: https://doi.org/10.17632/twxp8xccsw.9.
Yetgin, Ö.E., Benligiray, B., Gerek, Ö.N., 2018. Power line recognition from aerial images with deep learning. IEEE Trans. Aerosp. Electron. Syst. 55 (5), 2241–2252.
Yetgin, Ö.E., Gerek, Ö.N., 2018a. Automatic recognition of scenes with power line wires in real life aerial images using DCT-based features. Digital Signal Process. 77, 102–119.
Yetgin, Ö.E., Gerek, Ö.N., 2018b. Feature extraction, selection and classification code for power line scene recognition. SoftwareX 8, 43–47.
Yun, J., Jiang, D., Sun, Y., Huang, L., Tao, B., Jiang, G., Fang, Z., 2022. Grasping pose detection for loose stacked object based on convolutional neural network with multiple self-powered sensors information. IEEE Sens. J.
Zhang, J., Liu, L., Wang, B., Chen, X., Wang, Q., Zheng, T., 2012. High speed automatic power line detection and tracking for a UAV-based inspection. In: Paper presented at the 2012 International Conference on Industrial Control and Electronics Engineering.
Zhang, S., Wen, L., Bian, X., Lei, Z., Li, S.Z., 2018. Single-shot refinement neural network for object detection. In: Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.
Zhang, X., Xiao, G., Gong, K., Zhao, J., Bavirisetti, D. P., 2018. Automatic power line detection for low-altitude aircraft safety based on deep learning. In: Paper presented at the International Conference on Aerospace System Science and Engineering.
Zhang, H., Yang, W., Yu, H., Zhang, H. and Xia, G.S., 2019a. Powerline Dataset for Mountain Scenes (PLDM) [Image]. Retrieved from: https://github.com/SnorkerHeng/PLD-UAV.
Zhang, H., Yang, W., Yu, H., Zhang, H. and Xia, G.S., 2019b. Powerline Dataset for Urban Scenes (PLDU) [Image]. Retrieved from: https://github.com/SnorkerHeng/PLD-UAV.
Zhang, W., Liu, C., Chang, F., Song, Y., 2020. Multi-scale and occlusion aware network for vehicle detection and segmentation on UAV aerial images. Remote Sens. (Basel) 12 (11), 1760.
Zhang, J., Shan, H., Cao, X., Yan, P., Li, X., 2014. Pylon line spatial correlation assisted transmission line detection. IEEE Trans. Aerosp. Electron. Syst. 50 (4), 2890–2905.
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 Sens. (Basel) 11 (11), 1342.
Zhang, H., Sun, M., Li, Q., Liu, L., Liu, M., Ji, Y., 2021. An empirical study of multi-scale object detection in high resolution UAV images. Neurocomputing 421, 173–182.
Zhao, S., Wang, Y., Yang, Z., Cai, D., 2019. Region mutual information loss for semantic segmentation. Adv. Neural Inf. Proces. Syst. 32.
Zheng, C., Wu, W., Yang, T., Zhu, S., Chen, C., Liu, R., Shah, M., 2020. Deep learning-based human pose estimation: a survey. arXiv preprint arXiv:2012.13392.
Zhou, G., Yuan, J., Yen, I.-L., Bastani, F., 2016. Robust real-time UAV based power line detection and tracking. In: Paper presented at the 2016 IEEE International Conference on Image Processing (ICIP).
Zhou, Y., Qi, H., Ma, Y., 2019. End-to-end wireframe parsing. In: Paper presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision.