Taylor & Francis
Taylor & Francis Group

Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/taut20

Overhead power line detection from aerial images using segmentation approaches

Satheeswari Damodaran, Leninisha Shanmugam & N.M. Jothi Swaroopan

To cite this article: Satheeswari Damodaran, Leninisha Shanmugam & N.M. Jothi Swaroopan (2024) Overhead power line detection from aerial images using segmentation approaches, Automatika, 65:1, 261-288, DOI: 10.1080/00051144.2023.2296798

To link to this article: https://doi.org/10.1080/00051144.2023.2296798

Published online: 07 Jan 2024.

AUTOMATIKA
2024, VOL. 65, NO. 1, 261–288
https://doi.org/10.1080/00051144.2023.2296798

Overhead power line detection from aerial images using segmentation approaches

Satheeswari Damodaran[a], Leninisha Shanmugam[b] and N.M. Jothi Swaroopan[c]

[a]Department of Electronics & Communication Engineering, Meenakshi College of Engineering, Chennai, India; [b]Department of School of Computer Science & Engineering, VIT University, Chennai, India; [c]Department of Electrical & Electronics Engineering, RMK Engineering College, Kavaraipettai, India

ABSTRACT

Ensuring the optimal efficiency of electrical networks requires vigilant surveillance and preventive maintenance. While traditional methods, such as human patrols and helicopter inspections, have been longstanding practices for grid control by electrical power distribution companies, the emergence of Unmanned Aerial Vehicles (UAV) technology offers a more efficient and technologically advanced alternative. The proposed comprehensive pipeline integrates various elements, including preprocessing techniques, deep learning (DL) models, classification algorithms (CA), and the Hough transform, to effectively detect powerlines in intricate aerial images characterized by complex backgrounds. The pipeline begins with Canny edge detection, progresses through morphological reconstruction using Otsu thresholding, and concludes with the development of the RsurgeNet model. This versatile model performs binary classification and feature extraction for power line identification. The Hough transform is employed to extract semantic powerlines from intricate backgrounds. Comparative assessments against three existing architectures and classification algorithms highlight the superior performance of RsurgeNet. Experimental results on the VL-IR dataset, encompassing both visible light (VL) and infrared light (IR) images validate the effectiveness of the proposed approach. RsurgeNet demonstrates reduced computational requirements, achieving heightened accuracy and precision. This contribution significantly enhances the field of electrical network maintenance and surveillance, providing an efficient and precise solution for power line detection.

ARTICLE HISTORY
Received 8 January 2023
Accepted 10 December 2023

KEYWORDS
Power lines, UAV, Deep learning, RsurgeNet, Feature extraction, Classification algorithm, Hough transform

1. Introduction

Electricity is becoming increasingly important in modern communities. It poses major challenges when it comes to preserving the efficiency, affordability, and sustainability of the power supply. Visual checks of electrical grids are usually required by energy providers to prevent power outages and ensure a safe and reliable energy supply [1]. A combination of low-flying helicopters, airborne surveys, and foot patrols and tower climbs has been used to conduct these periodic checks. It is common for utilities and contractors to photograph faults and irregularities only, while others photograph the entire power grid, which includes conductors and power line parts as well. Detecting possible defects requires manual examination of the collected images after the light exposure. As well as being time-consuming and expensive, traditional inspection methods are also potentially dangerous since they may come into contact with live lines and result in death [2]. Several attempts have been made in recent years to automate visual power line inspections, including the use of autonomous helicopters, flying robots, and climbing robots; However, automated vision-based examinations have not yet been successfully deployed in this field because of its high accuracy standards and specific challenges. Rapid innovations in battery and fuel cell technology [3], cameras, and UAV have greatly enhanced the feasibility of using UAVs to automate data collection in automated vision-based power line examinations. To improve the accuracy of the detected lanes, the author uses an adaptive edge detection algorithm in the refinement stage after the automatic labelling phase. The processed images are then used as label data to train a second stage model, with unidentified lanes being shielded. The overall effectiveness and precision of the lane detection system are intended to be enhanced by this two-stage procedure [4]. This paper proposes a power line localization algorithm that is weakly supervised and only uses image level class labels. An algorithm based on sliding windows and convolutional neural networks (CNNs) is proposed to classify sub-regions. A sub-region is filtered out if it is classified

CONTACT Satheeswari Damodaran satheeswari.ykp@gmail.com Meenakshi College of Engineering, No.12, Vembuliamman Koil Street, West K.K.Nagar, Chennai, Tamil Nadu 600078

as having no power lines. In order to visualize a power line's location, feature maps derived from intermediate convolutional layers are combined with sub-regions in an image classified as power lines [5]. An investigation of the importance of each layer is presented in this article, as well as experiments with fusion features for image retrieval. The model is used to extract image feature information. The results demonstrated that the deep convolutional neural network (DCNN) model's fusion feature can improve image retrieval results for the datasets. AlexNet and Caffe frameworks were used for extracting image feature information. Two public image datasets, Inria Holidays and Oxford Buildings, were used in this experiment to determine the impact of different datasets [6]. The author experimented with feeding features to various classifiers by extracting them from intermediate stages of CNNs and training CNNs from scratch. A thorough list of recommended usage practices for CNNs for power line detection is produced as a result of these tests, which were conducted again with various architectures and pre-processing techniques [7]. In this paper [8], described an autonomic vision-based method for segmenting power lines from optical images captured by UAVs and analyzed with DLs as the foundation. Power line segmentation is frequently thought to be a requirement for power line inspection. The author [9] here demonstrated that model has a high recognition speed and accuracy, the Unmanned Aerial System can inspect quickly, saving time and money while increasing safety. To investigate the capability of Ensemble Learning (EL) algorithms for classifying Remote Sensing (RS) data, used three EL algorithms: Adaptive Boosting (AdaBoost) [10], Light Gradient Boosting Machine (LightGBM) [11], and Extreme Gradient Boosting (XgBoost) [12]. In order to compare the performance of these algorithms on three types of RS data, three types of classification scenarios were created: multispectral and hyperspectral data as well as PolSAR and high-resolution multispectral. Using a trajectory between two power line towers, the author [13] developed an inspection system for PL. The results of the CNN were satisfactory in identifying PLs and isolating the towers. In this paper, a sequential algorithm is proposed for detecting power lines across local and global regions. An edge map image is identified using matched filters (MF) and first-order derivatives of Gaussian (FDOG) with morphological filtering as the local criterion. High detection rates lead to over-detection. Using graph theory, the following global criterion splits line segments into PL using a graph-cut model [14]. A hough transform is used to isolate PLs using thresholding and morphological operations. In order to calculate an altitude relative to observed lines with a constant diameter, a perpendicular camera orientation is considered. This approach may be flawed due to PL accuracy or image resolution and size constraints, which may cause data loss. As a part of a study [15], the author compared the performance of various vision-based systems used to manage power lines based on various vision sensors. According to the researcher [16], power lines and their environments can be assessed using remote sensing. A variety of data sources from hard to discern satellite images to perfectly visible images from aerial unmanned vehicles were analyzed. PLE-Net integrates an encoder-decoder framework with attention mechanisms, specifically self-attention and multi-scale feature enhancement. These components aim to improve the model's ability to capture relevant contextual information for the task of PL extraction in aerial images [17]. The author [18] proposed a tool for extracting power lines from aerial images using background suppression, sky segmentation, and the hough transform, with some divergences due to line orientation and camera positioning. The author [19] used motion estimation at low altitude to create a narrow line detection method for low altitude autonomous drones used in rescue operations. To recognize cables, PLs, and wires, a pixel-level edge detection technique and a window hough transform are used. The author [20] used Canny, a typically used edge detector, to detect power towers. The contours were first extracted using the Canny edge detector. After dividing the image into 10×10 pixel boxes, straight line transformations were applied. By measuring the length and number of straight lines on the power tower, the false box is removed, and the tower is classified using hand-craft rules. Two inspection videos with a resolution of 1920 x1080 pixels and a frame rate of 25 frames per second were used to test the process. In this presentation [21], the author presented a novel technique for detecting and monitoring PLs. As a first step, PL segments are extracted using the hough transform. Based on the characteristics of PLs in aerial photos, used K-means clustering and filtering in the hough space to detect power lines. In the hough space, power lines are monitored using a kalman filter based on the continuity of a video series. Detecting and monitoring power lines in complex environments is successful using the proposed method, according to the experimental results. Based on the hough transform and a coupled neural filter, the author [22] presented a method for PL detection and monitoring. They use the Otsu threshold approach in this situation, which provides better results than PCNN filters. The algorithm employs the Radon transform for initial line segment extraction, a grouping method to connect these segments, and kalman filter technology for smoothing and connecting the segments into entire PLs [23]. Then, the work here presents a comprehensive approach for inspecting electrical infrastructure using a drone with multi-modal sensors. Computer vision methods, including infrared imaging and neural networks, are employed to detect and classify faults and damaged components, with a specific focus on

insulators [24]. Both main modules make extensive use of semantic and location information, the model can focus on PLs rather than unexpected scenes. In the course of deliberating the various methods for segmenting our dataset [25], various pre-processing techniques were explored.

Since neural networks have advanced, CNNs are able to automatically learn high-level representations of natural images. Various well-known CNN-based methods have made significant progress in this field by applying useful hierarchical features to classify pixels [26,27]. Using a simple CNN for power line and tree classification, the author [28] proposed mapping PLs and identifying potential tree conflicts. The author focuses on exploiting geometric relationships related to circle symmetry for the efficient detection of line segments in power line images. It utilizes well-known techniques such as canny edge detection and steerable filters. This algorithm's effectiveness has been demonstrated both by synthetic images and real-world images as part of the validation tests. There are also several research works that find its application in power line identification, including the You Only Look Once (YOLO) architecture, a fast and common lane detection method [4] used for road lane detection. In order to achieve satisfactory real-time diagnosis, these methods require a lot of training data and, ideally, a high-performance GPU [29]. The purpose of this paper is to develop a vision-based, fully autonomous power line inspection system. Here used Unmanned Aerial Vehicle (UAV) inspection as the primary inspection method, optical images as the primary data source, and deep learning as the foundation for data analysis and inspection in our new automatic autonomous vision-based PL inspection concept [30]. Images of the entire PL spans which may have a significant curvature are needed for these measurements. This essay evaluates, two widely used Hough transform techniques, two smartphone cameras, and a line tracing algorithm are used to evaluate the feasibility of ground-based imaging with smartphones and quick, reliable power line detection [31]. Next, a convolutional capsule network (CapsNet) is used to extract the depth hierarchical features from a photo of a power line scene. Ultimately, the decoding layer reconstructs the power lines in the scenes that have power lines by using the information generated by the CapsNet output layer to distinguish between scenes with and without power lines [32]. Using well-known classifiers on the DCT submatrices and systematically varying the DCT matrix sizes, the goal of this study is to extract signatures from DCT coefficients. Six different submatrices were chosen, with a total of three starting points and three different sizes (4, 8 8, 16 16, 32 32, 64 64, and 128 128). Block by block, (i) the top-left corner (around DC) is scanned; (ii) the entire DCT matrix is scanned; and (iv) the bottom-right corner (high frequency) is scanned [33].

This study used various image transformations, filtering methods, and morphological operations to overcome aforementioned problems, as well as proposed a novel CNN – based deep learning framework RsurgeNet. There are two applications for this model: (i) binary classification and (ii) feature extraction, which uses trained PLs images to extract features and then these feature maps are transferred into different CA. To the output of (i) and (ii), the technique hough transform for line detection is used effectively to determine the presence of a power line in aerial images. This work considered the methodologies proposed by [dataset], [7,31,32] and adapted them to address our research objective while improving efficiency.

This research work strives to provide some extensive insights into power line detection with these salient features:

- Introduction of a comprehensive pipeline integrating preprocessing techniques, deep learning models (AlexNet, VGG16, ResNet-50, RsurgeNet,),classi fication algorithms (AdaBoost, LightGBM, XgBoost), and the Hough transform for enhanced power line detection in complex aerial images.
- Targeting challenges posed by intricate background environments, diverse topographies, and varying lighting conditions encountered in aerial imagery to improve accuracy in power line detection.
- Dual functionality of the RsurgeNet model: serving as a binary classifier (distinguishing power line presence or absence) and extracting meaningful features for subsequent classification algorithms.
- Extensive evaluation on a dataset comprising visible light (VL) and infrared light (IR) images, showcasing superior performance of the RsurgeNet model compared to existing architectures and classification algorithms in terms of accuracy and computational efficiency.
- Outlining the study's scope, materials used, methodologies (including Canny edge detection, morphological reconstruction with Otsu thresholding, and gradient boosting models, hough transform), and Python-based experiments to validate power line detection.
- Providing comprehensive insights into power line detection by considering various image transformations, filtering techniques, and model architectures.
- Concluding with a summary of experimental results, emphasizing the achieved outcomes compared to existing methodologies and their implications in electrical network maintenance and surveillance.

As a scope of the present study, Section 2 covers Python experiments with materials and methods, Section 3 presents results and discussion, and Section 4 presents the conclusion.

2. Materials & methods

2.1. Proposed system workflow

The segmentation of input images is performed using preprocessing techniques to learn the patterns and characteristics of PLs. One such technique is Canny edge detection, which helps to identify the edges of objects in images. Another technique used is morphological reconstruction with Otsu thresholding, which helps to enhance the images for better segmentation(2.3). The preprocessed images are then fed into a novel DCNN model called Recurrence of Increased Learning in Neural Network (RsurgeNet- R-"Recurrence", "Surge"-Increase, "Net"-Network). This model is specifically designed for two purposes: First purpose, the model's trained images are first classified according to training accuracy, losses, validation accuracy and losses. Second purpose is to extract more meaningful features from the intermediate stages of the network, the features are fed into three gradient boosting models: AdaBoost, LightGBM and XgBoost. This dataset consists of with and without PL images, which helps to identify any false positives or false negatives in the model's predictions. These models classify and predicts the presence or absence of PLs in the images. The value of "1" indicates the presence of a PL, while "0" indicates its absence. To further validate the results of the trained model, the test images are inputted into the hough transform. The process followed for the detection of images with and without PL is presented in Figure 1.

2.2. Dataset

The aerial image dataset employed in this study originated from the Turkish Electricity Transmission Corporation (TEIAS) and is subsequently adopted by [Dataset], [7] for their exploration of high voltage transmission line detection utilizing DCNNs. This dataset comprises an array of visual light (VL) and infrared (IR) videos captured via helicopter-mounted imaging systems across 21 distinct locations within Turkey. With a total of 8000 images,4000 IR and 4000 VL images – this dataset underwent resizing to 128×128 pixels for standardized analysis, showcasing samples in Figure 2. Notably, the IR folder contains 2000 images with and without power lines, mirroring the VL folder's 2000 images for each power line scenario. Encapsulating a broad spectrum of aerial scenarios, this diversity intentionally encompasses both typical and highly intricate environments encountered during aerial inspections. One standout advantage of this dataset is its inclusion of both IR and VL imagery. While IR images offer heightened resilience to environmental changes, their lower resolution and subdued contrast between power lines and backgrounds differ from the VL counterparts. Nonetheless, both image types contribute significantly to understanding power line recognition nuances. Although the original video resolutions were 576×325 for IR and full HD for VL, the resizing to a consistent 128×128 dimensions is pivotal for uniformity, enabling precise and standardized analyzes. Beyond its representation of diverse scenarios and IR-VL incorporation, this dataset's added strengths lie in its potential for comparative analysis of environmental adaptability between IR and VL images, contributing substantially to robust power line detection algorithm development and validation.

2.3. Segmentation models

By eliminating image noise with the Otsu technique, detecting edges in images, and object filtering using morphological operations, an optical detection of power lines can be used to accurately segment power lines from an image's background.

2.3.1. Image binarization/thresholding

In image binarization [34,35], pixels are divided into either "1"s or "0"s based on the intensity of their pixels. Prior to edge detection and structural operations, image binarization is used to obtain the general features of image elements. Otsu's method [36], a global optimum threshold algorithm, is a common method of thresholding. To determine the optimum global threshold value, Otsu uses the grey level histogram of the image in an unsupervised and nonparametric way. Otsu's method, however, requires high intensity values in PL images compared to other image artifacts in order to work.

2.3.2. Edge detection

The main objective of edge detection is to identify a feature from the input image of PL, which is done primarily with canny edge detection [37]. Canny edge detection reduces the amount of data to be processed by extracting useful structural information from vision objects. The parameter setting is vital in edge detection because it allows the shape of the object to be clearly seen in a particular number by adjusting the parameter. Edge detection can be solved using a set of techniques.

2.3.2.1. Image smoothing. Image Smoothing employs a filter known as the Gaussian filter to minimize the noise effect in PL images. The Gaussian function can be represented by the equation (1).

$$H = \left(\frac{1}{2\pi\sigma^2} esp\left(-\frac{(-i-(k+1))^2}{2\sigma^2}\right)\right); \qquad 1 \le i,j \le (k+1) \tag{1}$$

In this equation, i and j refer to the pixel coordinates and its size, which $(2k+1) \times (2k+1)$ of the kernel.

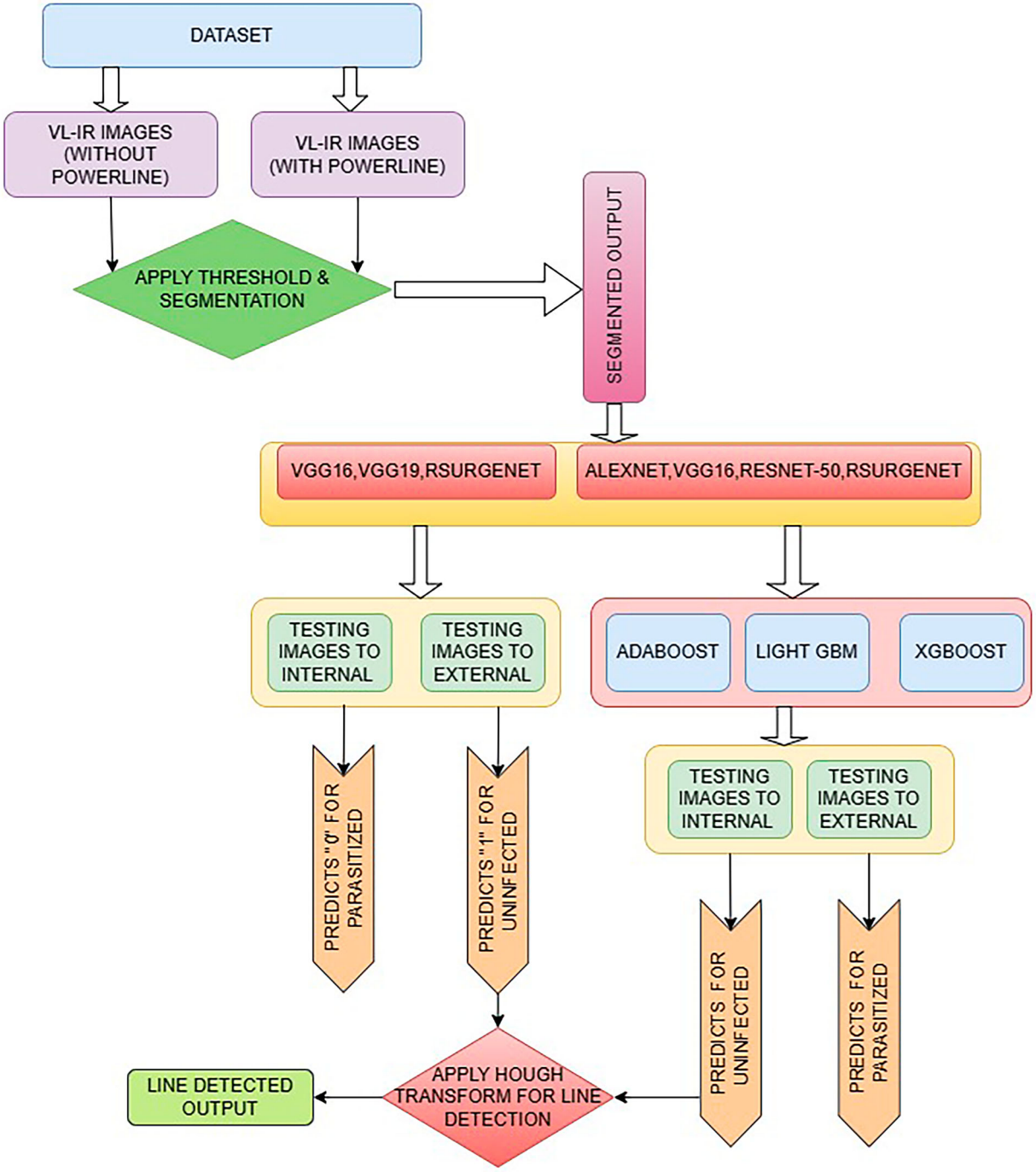

Figure 1. Proposed power line detection system workflow.

2.3.2.2. Threshold setting. The maximum and minimum thresholds are needed for suppressing the weak edges in aerial images and maintaining only the strong edges. After non-maximum suppression, the gradient pixel value of the remaining edge pixels should be compared. If the threshold value is greater than the full threshold, the edge would be strong. Otherwise, the edge would be the weak edge if the threshold value is less than the minimum threshold [38].

2.3.2.3. Edge threshold. The gradient based canny edge detector is another form of edge extraction using thresholds in UAV_PL images. Using a gaussian kernel to smooth the image, the model computes the gradient magnitude of the image. Non-maximum suppression preserves only edge points whose gradient magnitude is local maxima. The final step connects edges using hysteresis thresholding, a connectivity analysis that involves a lower and upper threshold [39]. In order to qualify as a true edge, edge candidates must score higher than the upper threshold as it is related to a true edge, while candidates that fall between the upper and lower thresholds cannot qualify as a true edge. If the edge strength of the current pixel is greater than that of other pixels, the edge strength value is retained. It is, on the other hand, silenced.

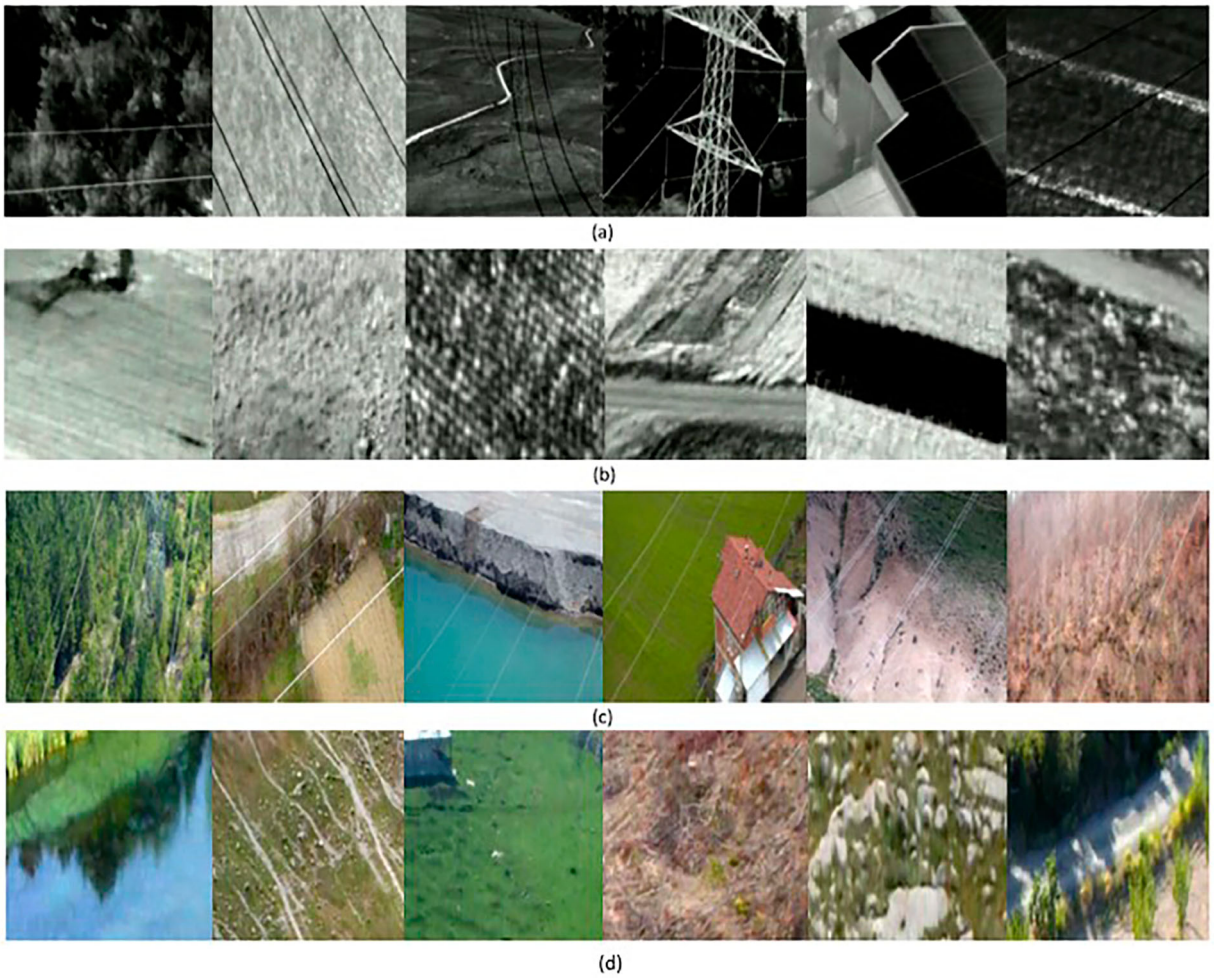

Figure 2. (a), (b) 128 × 128 IR images with and without power lines respectively; (c), (d) 128 × 128 VL images with and without power lines respectively.

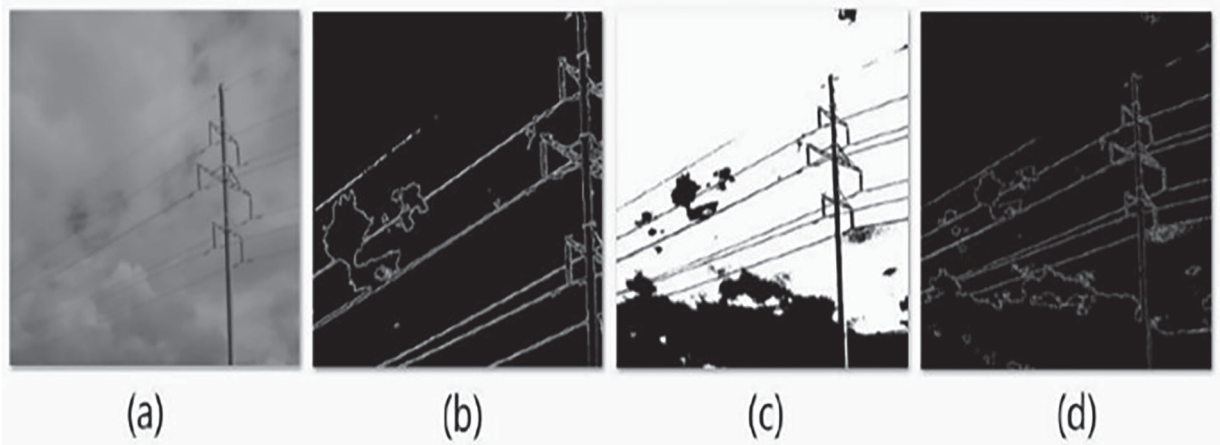

Figure 3. (a) greyscale image; (b) canny edge detection; (c) skeleton threshold output; (d) otsu binary thresholding with dilation and erosion output.

2.3.3. Morphological processing with skeleton threshold

In contrast to canny edge detection [40], skeleton thresholding, as shown in Figure 3, is a method that runs faster and needs less memory. Only the shape of the object from the available dataset will be preserved in skeleton thresholding, and each redundant pixel will be removed. Applying binary threshold, gradient calculation, declaring structuring element, erosion and dilation are the key steps in the skeleton thresholding method. The simple and straightforward thresholding is used in the first step, applying binary threshold, to assign a threshold value. There are also two scenarios in which the threshold and pixel value are compared. The pixel value is assigned to one of two values (black or white) if it falls below the specified threshold. Unless

the pixel value reaches the assigned threshold, it is given one value (either white or black). The OpenCV function THRESH_BINARY_INV implements binary thresholding by changing the colour allocated to low and high threshold pixels, and the colour can range from black to white or from white to black. Secondly, in order to measure the intensity gradient of a grey scale picture, first-order derivatives of gradients are used both horizontally and vertically. The values are calculated using the internal sobel kernel. The horizontal intensity gradient and the strength gradient in the vertical direction can be used to measure the overall gradient value. Then, in the third stage, a structuring element such as erosion and dilation must be defined in order to compute the morphological operations. The 3×3 sized and cross shaped structure element is also assigned in this process. Corresponding to the size and shape of the structuring element declaration, pixels are removed from the image's boundary in the fourth stage. According to the first erosion rule, the output pixel must have a value equal to the minimum value of all pixels around it. A binary image that contains zero pixels must have a zero-output pixel as well. The image's border pixels [41] are given the maximum value allowed by the data type. The binary image's pixels are believed to be set to 1. For uint8 images, the highest value in grey scale images is 255. Finally, the fifth step, dilation, pixels from the image's boundary are inserted, with the number of pixels added depending on the shape and size of the structuring element declaration. Input pixels' maximum values are summed to generate output pixels' maximum values. If all of the pixels in a binary image are set to value 1, the output pixels are set to 1. The padding behaviour, i.e.the pixels around the image boundaries are processed, is specified in rule two of dilation. If the pixels reach beyond the image boundary, the data type's minimum value is assigned to them. For binary images, the pixels are believed to be set to 0. For grey-scale images, the uint8 image's minimum value will be set to 0.

2.3.4. Otsu thresholding with dilation and erosion

This method of applying the Otsu threshold with erosion and dilation is identical to the previous one, which also involves erosion and dilation in the skeleton thresholding. In addition to the above discussed five steps in skeleton thresholding, this approach also includes the laplacian filter and otsu thresholding, which are used to suppress any external noises.

2.3.4.1. Adding laplacian filter. Sobel kernels of size three are used in Laplacian filtering to implement the Laplacian operator. The first-order derivative is calculated directly in the Sobel operator calculation [42]. When calculating the second order derivative of an image in both directions, the laplace operator comes in handy. The transfer function can be calculated using the laplace operator.

2.3.4.2. Otsu threshold. Image thresholding is done automatically using clustering in the Otsu threshold implementation. A method for calculating the optimal threshold separating foreground and background pixels (bi-modal histogram) is implemented in the two types of pixels (foreground and background pixels). With this approach, a variation is generated with a small spread (intra-class variance), or, to put it another way, a combination with a small spread (inter-class variance) is created, due to of pairwise squared distances being constant. The inter-class variance is then at its maximum. This method is implemented using the OpenCV function THRESH_OTSU. There is an optimal threshold difference, denoted by the letter "t". The variances are expressed by the $\sigma_0^2(t)$ and $\sigma_1^2(t)$ variables. The probabilities of foreground and background pixel groups are determined by two weights of ω_0 and ω_1. As a result, the intra-class variance is computed as in equation (2). A comparison of the preprocessing results can be seen from the Figure 3.

$$\sigma_\omega^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t) \tag{2}$$

2.4. CNN classification model

CNN are a subclass of deep neural networks that optimize the use of the dimensional structure of visual imagery by combining a fully connected layers net for classification with a parametric, sparse net for feature extraction. CNN's feature extraction process is data-dependent. Moreover, the feature extraction and classification components are trained simultaneously. The purpose of our research is to train CNNs to identify power lines from images using labelled datasets and ground truth.

2.4.1. Development of our model RsurgeNet

A classic CNN is constructed using several convolutional layers, pooling (or subsampling) layers, fully connected layers, and a softmax decision layer. The purpose of pooling layers in CNNs is twofold: The first stage is to define invariance into the input's small translations. The second objective is to gradually reduce the spatial component of the input volume, thereby lowering the number of parameters and the computation time in the network. Fully-connected layers are made up of interconnected neurons to all previous layer activations. Typically, fully connected layers are responsible for increased reasoning in CNNs. To be more precise, fully connected layers are typically attached to the end of CNNs to create a major semantics and to perform classification based on features derived by the previous layers. Numerous deep CNN frameworks, including AlexNet [43], Vgg16 [44] and ResNet-50 [45], have

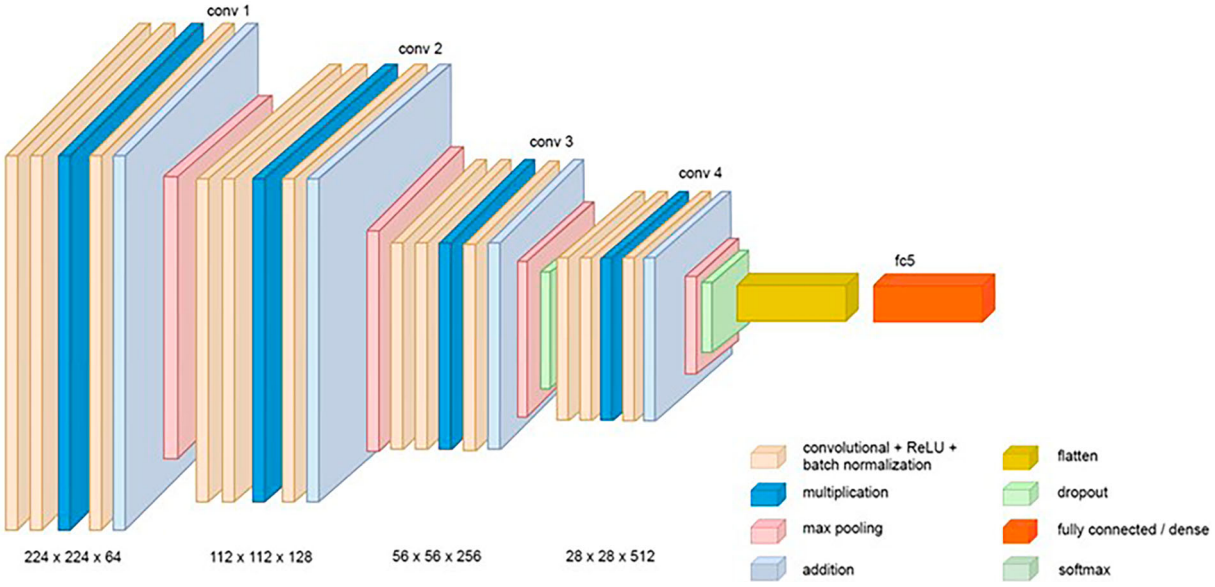

Figure 4. The proposed RsurgeNet architecture.

been constructed by simply stacking numerous convolutional, pooling, and fully connected layers. Because, the information flowing through those deep CNNs undergoes multiple stages of multiplication, the gradients must be backpropagated through multiple stages during training. Other layers like dropout, batch normalization (BN) [46], and group normalization are frequently added to CNNs in means to enhance performance and avoid overfitting. Three fully connected layers and thirteen convolutional layers make up the original VGG16 network, which is separated into five phases by four pooling layers. One of the most desirable aspects is that convolutional layers with varying speeds can capture multi-scale variables and provide hierarchically coarse-to-fine responses. Therefore, our framework is based on the VGG16 architecture, as shown in Figure 4. The proposed architecture uses five convolution layers, over and above the layers already present in conventional method, to create a basic network that produces intermediate layers' outputs. A multitude of helpful fine details are provided by intermediate layers, while convolutional features gradually soften. VGG16's convolution layers are included in our RSURGENET system, along with addition layers (AL) and multiplication layers (ML). These two layers can be used to build a deeper model that will extract more features for extremely precise feature extraction, predictions, and classifications. This section provides a detailed explanation of RsurgeNet. The network architecture of RsurgeNet, the suggested power line detection framework, is described in detail in this section. In the profounded work, RsurgeNet involves four blocks and in each block, there are "3" Convolutional layers(Convol_layer), "2" Batch normalization layer (BN), "1" Multiplication layer (ML), "1" Addition layer (AL), "1" Max Pooling layer (MaxPool_layer) and "1" dropout layer (DP_L). By default, all layers use stride "1" and padding. DP_L dropout is used to eliminate irrelevant features which is applied only to the third and fourth blocks. The first block consists of Convol_layer_1, Convol_layer_2 and Convol_layer_3; Convol_layer_1 contains 64 filters with a kernel size of (3,3), "Relu" activation with "same" padding, followed by BN_1. In the first block, the Convol_layer_2 has 64 filters with a kernel size of (3,3) and "Relu" activation with the "Same" padding. Next comes a ML_1 where Convol_layer_1 and Convol_layer_2 is multiplied element by element, followed by a BN_2. Within the first block, Convol_layer_3 is composed of 64 filters, each with a kernel size of (3, 3) followed by "Relu" activation with the same padding, and then AL_1, where element wise addition occurs, and the output of Convol_layer_3 combines with the output of ML_1 to produce the outcome. The second block consists of Convol_layer_4, Convol_layer_5, and Convol_layer_6; The Convol_layer_4 containing 128 filters with kernel size of (3,3), as well as "Relu" activation with "Same" padding, followed by a BN_3. The second block's Convol_layer_5 provides 128 filters with a kernel size of (3,3) and the same activation with "Same" padding, followed by a ML_2, which multiplies the Convol_layer_4 and Convol_layer_5, followed by a BN_4. Within the second block, the Convol_layer_6 consist of 128 filters with kernel size of (3,3) and "Relu" activation followed by "Same" padding is followed by an AL_2 that performs element-by-element addition. The output of Convol_layer_6 then combines with the output of ML_2 to generate the outcome. Third block is made up of Convol_layer_7, Convol_layer_8, and Convol_layer_9, with Convol_layer_7 integrating 256 filters with kernel size (3,3) and "Relu" activation with "Same" padding, followed by BN_5. There are 256 filters in convol_layer_8 in the third block with a kernel size of (3,3) and a "Relu" activation with a "Same" padding. Following this, ML_3 is followed by element-by-element

multiplication of Convol_layer_7 and Convol_layer_8 and is succeeded by a BN_6 layer. Third block consists 256 filters with a kernel size of (3,3) and "Relu" activation followed by same padding and an AL_3 that adds elements by element. The output of Convol_layer_9 merges with the output of ML_3 to gives the outcome. Next to the AL_3 is the DP_L1. The fourth block consists of three Convol_layer: Convol_layer_10, Convol_layer_11, and Convol_layer_12; Convol_layer_10 is 512 filters with kernel size (3,3) and "Relu" activation with "Same" padding, followed by BN_7. Convol_layer_11 in the fourth block has 512 filters with kernel size of (3,3) and "Relu" activation with the same padding. ML_4 where element wise multiplication of Convol_layer_10 and Convol_layer_11 takes place, followed by a BN_8. In the fourth block of the proposal, Convol_layer_12 is composed of 512 filters with kernel size of (3,3) and "Relu" activation with "Same" padding. This is followed by an AL_4 in which the output of Convol_layer_12 combines with the output of ML_4. The DP_L2 is located next to the AL_4. Adding dropout, flatten, and dense layers with sigmoid activation produces a vector size of 100352 when used with binary classifiers. Relu, SoftMax, Cross entropy, Adam, and batch size as 32 are the essential qualities for this model. In the 2D convolutional layer, the filters identify that the kernel used is 3x 3 and then combine it with the input images to create the activation map for our dataset. In BN, input features are stabilized for every mini batch. This reduces the number of epochs involved in training the RsurgeNet model. In the proposed CNN model maxpooling eliminates overfitting by allowing down samples without losing valid data. Padding is to retain the original data size and to carry out the convolution process. The role of a neuron is determined by its activation negative values and only provides a linear output for positive values, resulting in a model that can satisfy certain properties. The optimizer in neural network(NN) is Adam [47] that learns the past weights and is calculated by using equation (3).

$$f(X) = \begin{cases} 0 \; for \; X \; < \; 0 \\ X \; for \; X \; \geq \; 0 \end{cases} \tag{3}$$

2.4.1.1. Working of multiplication layer and addition layer in RsurgeNet. The multiplication layer in the novel model operates element by element with two elements. When the two matrices I and J are multiplied, the operand K has the same dimensions as I and J, which are depicted in Figure 5(a). The neural network learns more features as the features are multiplied, resulting in increased accuracy. Similarly, addition is also performed element by element on two elements A and B, resulting in an output matrix C with the same dimensions as A and B, as shown in Figure 5(b). The addition layer in RsurgeNet adds features from previous layers, allowing remembrance of previous features necessary for predicting target objects.

2.4.1.2. Feature extraction technique. An image must contain distinct features that can be extracted [33]. Different information from the images is gathered in each filter. Here, feature extraction is accomplished with the aid of filters. The model becomes more focused and is better able to discover the underlying patterns and relationships in the dataset by eliminating redundant and unnecessary data. With the poorly cluttered background and while using classifier it eliminates the false prediction of classes and gives the best outcome. This results in decreased computation complexity, decreasing overfitting, speeding up learning, quicker learning times and increased accuracy. Additionally, optimizing the data used in machine learning models is crucial to data optimization, improving data visualization, and managing memory.

2.4.2. Loss function for classification

Equation (4) provides the SoftMax loss function, which is used to estimate the confidence score C_i^P each proposed bounding box i per category p, with one label per bounding box, $x_{ij}^P = 0{,}1$ indicates that the i^{th} proposed box and the j^{th} ground truth box of category p match.

$$\dot{C}_i^p = \frac{\exp\left(c_i^p\right)}{\sum_q \exp\left(c_i^q\right)} \tag{4}$$

2.4.2.1. Classification loss. When training a machine learning model, selecting a loss function is essential because it directs the optimization process that modifies the model's parameters to reduce prediction errors. Popular options for classification tasks are binary cross-entropy and sparse categorical cross-entropy, each of which is appropriate given the particulars of the situation.

2.4.2.1.1. Cross entropy loss. It is calculated as the sum of the negatives of all predicted probabilities/outputs multiplied by (actual outputs) for as many classes as there are and its function are given by equation (5),

$$L_{\text{class}}(x, c) = \sum_{i \in Pos}^{N} \sum_j x_{ij}^p \log\left(\hat{c}_i^p\right) - \sum_{i \in Neg} \log\left(\hat{c}_i^p\right) M_{gt} \tag{5}$$

The actual output is y_i, the predicted output probability at the ith position is $p(y_i)$, and the number of training examples is N.

2.4.2.1.2. Hinge loss. Typically, in machine learning methods, this function penalizes predictions not only when they are incorrect, but also when they are insecure. It penalizes extremely accurate predictions. Correctly predicted outcomes, on the other hand, are not

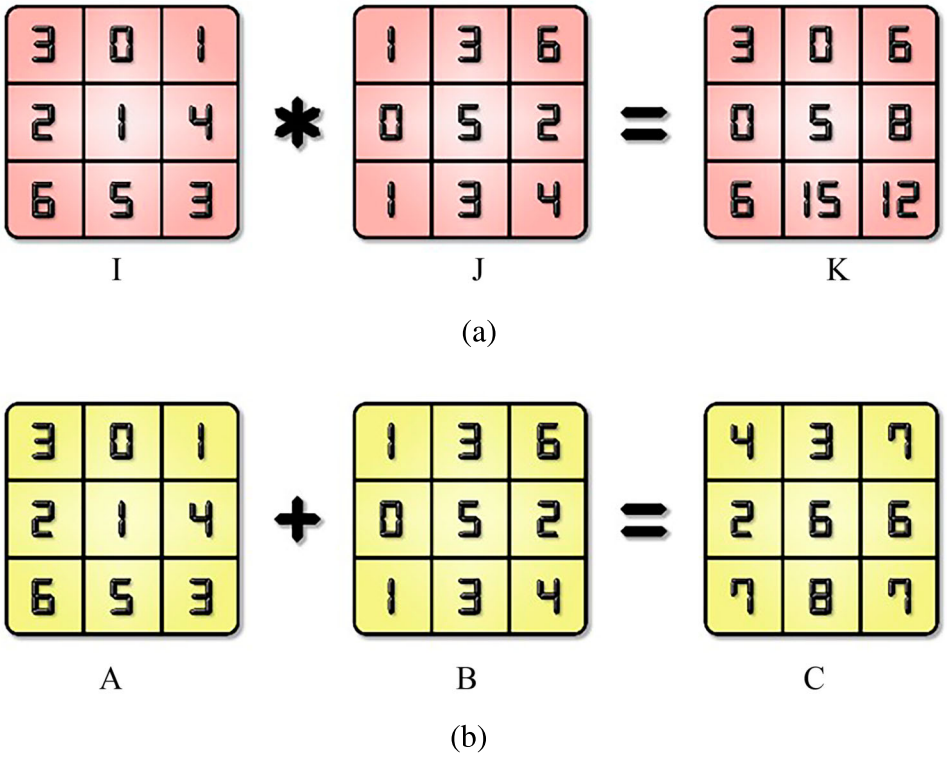

Figure 5. (a). Element wise multiplication. (b). Element wise addition.

penalized and its evaluated using equation (6).

$$\sum \max(0.1 - yi * h\theta(xi)) \tag{6}$$

2.4.2.1.3. Mask loss function. At the pixel-level, the mask loss function is Binary Cross Entropy (BCE) loss between the predicted mask $M_{pr}$and the ground truth mask $\boldsymbol{M_{gt}}$. The calculated mask's accuracy can be improved by using BCE loss, where $L_{mask} = BCE\ (M_{gt}, M_{pr})$.

2.4.3. Optimizer

Optimizers determine the amount to which the loss function is minimized in CNN designs. An optimizer is a function that improves accuracy by updating model parameters. Weight updates will vary depending on the optimizer's preferences. To verify the efficiency of the three proposed components' faulty architecture, three optimizer methods were chosen: stochastic gradient descent (SGD), RMSProp, and Adam. An adaptive learning rate optimization technique is used in Adam [47] to train deep neural networks. To precisely determine learning rates for every parameter, the algorithms in this proposed work use adaptive learning rate approaches. Adam can be thought of as an RMSprop and SGD hybrid with momentum. By utilizing a moving average of the gradient rather than the gradient itself, it takes full advantage of momentum, similar to SGD with momentum, and scales the learning rate using squared gradients. Adam optimizer is used to train the network, with a learning rate of 0.001 and a gamma decrease of 0.5 (multiplicative factor) every 30 epochs. After 350 iterations, the weights with the best performance on the validation set were saved.

2.4.4. Learning rate

A hyperparameter measuring the rate at which the model weight can be increased to learn the problem modified optimally is known as learning rate. When training a neural network, it is one of the most important tuning parameters to adjust. The study looked at the effect of changing the learning rate while keeping the other parameters and hyperparameters fixed.

2.5. Various machine learning classifiers

In order to classify CNN features, used three different classification algorithms (CA). These algorithms classify an image as containing or not containing power lines based on ground truth data and a labelled dataset.

2.5.1. Tuning of hyper parameters

The hyper tuning parameters of various machine learning classifiers are discussed below.

2.5.1.1. Estimators. Machine learning applications benefit from its consistent interface. Scikit-Learn implements all machine learning applications using the Estimator API. An estimator is a data-learning object(fits data). Raw data can also be used for extraction of useful features, in addition to classification. In a model, the likelihood of previously unseen data is high (higher is better). One of the most important API objects is the estimator. Estimators(n) are objects that

infer properties from new data based on some training data.

2.5.2. Adaptive boosting

AdaBoost [10] is a machine learning method and a popular boosting technique. AdaBoost is a supervised learning algorithm. It is an EL method also called meta learning, that helps in increasing the classifier efficiency. AdaBoost is a very good learner that learns from weak classifiers and makes the weak learner, strong learners. It helps in decreasing the bias error when the model is not able to identify the relevant images. Boosting algorithms develop a strong predictive model from the weaker model. The model is created from the training data and the new model is created based on the previous model by reducing the error. Reducing the bias error produces strong learners, and determining the model's classification accuracy is vital. The CNN are trained initially. Once the models are trained with training images; they normalize all the input images and the features are extracted using the model for the respective original samples. Through the feature extracted layers, the feature maps are further carried out to the AdaBoost classifier with the trained labels as "0", "1" for the respective classes, thereby assisting in predicting the classified turnout.

2.5.3. Light gradient boosting machine

Light GBM [11] is a powerful decision tree algorithm that utilizes a gradient boosting framework to effectively handle various ML tasks, including ranking and classification. The leaf-wise algorithm employed by Light GBM offers several advantages over traditional depth-wise or level-wise splitting. By splitting the tree leaf-wise, Light GBM is able to achieve the simplest fit possible, which ultimately leads to more accurate results. The leaf-wise algorithm is designed to reduce loss more efficiently, resulting in significantly improved accuracy compared to existing boosting algorithms. In the proposed work, the image classifier algorithm Light GBM plays a crucial role in accurately predicting the classified outcome. This algorithm is utilized after a series of preprocessing steps. Initially, all input images undergo normalization to ensure consistency and improve the performance of the model. Following the normalization of the images, a CNN model passes them through its convolutional layers. Images are analyzed using these layers in order to extract important features. After proper training, these features are identified as values, such as "0" or "1". It produces classified output images after extracting the feature maps, which represent the learned features from the images.

2.5.4. Extreme gradient boosting

XgBoost [12] is a sequential EL algorithm. It is a supervised learning algorithm which is faster when compared to other algorithms, because of its parallel and distributed computing, XgBoost works on the top of gradient boosting. It is independent, portable and integrable with cloud. It can also handle missing values automatically. XgBoost has adjustable parameters such as lambda, gamma and alpha which makes it powerful. Tuning those parameters according to the dataset reaches the best results. XgBoost creates a base model by calculating the average of the target features. This CA produces residuals and calculates the similarity score for each node using the formula (7).

$$Similarity\ score\ = \frac{(sum\ of\ residuals)^2}{R+\lambda} \tag{7}$$

Where, R – Number of residuals

λ – lambda

Gain = similarity score after split – similarity score before split

Whenever the gamma value is less than the gain value, then a split will happen. The growth of the tree will depend on the higher proportion of gain by splitting and if the gain becomes lesser than or equal to gamma value, then the split ends, and the model is pruned. In order to control the overfitting of a model, regularization of its parameters must be increased. In the proposed work, the image classifier algorithm XgBoost helps in predicting the outcome exactly. Initially, all the input images are normalized and passed on to the convolutional layers of CNN model. Only then, the model becomes perfect in extracting feature maps with the trained values as "0", "1". The extracted features maps are transferred into the XgBoost classifier of machine learning algorithms. Finally, XgBoost is subsequently trained with the machine trainable features maps and it throws classified output in the form of array or images.

2.6. Line detection using hough transform

A detected line is usually the maximum local response point in a modified parametric space when using edge detectors such as canny [39] and Sobel [42]. An edge image can be extracted with the hough transform if the lines are not all connected. The algorithm generates a set of lines that pass through an edge point, as shown in equation (8). (x_i,y_i) using the usual line representation and by modifying the parameter, "r" represents a distance between a perpendicular point over the line and the origin, indicating an angle between the x and y-axes. Accumulator cells are generated and sub-divided in a matrix from a selected range of θ and r values. For every change in θ at each point (x_i,y_i), their associated r values are computed. The accumulator cell with the correct parameter range is increased for each pair of θ and r. Finally, an accumulator cell threshold is introduced by the local maxima of the accumulator cells, allowing the extraction of line candidate parameters and forming probable line parameters of lines in the real picture.

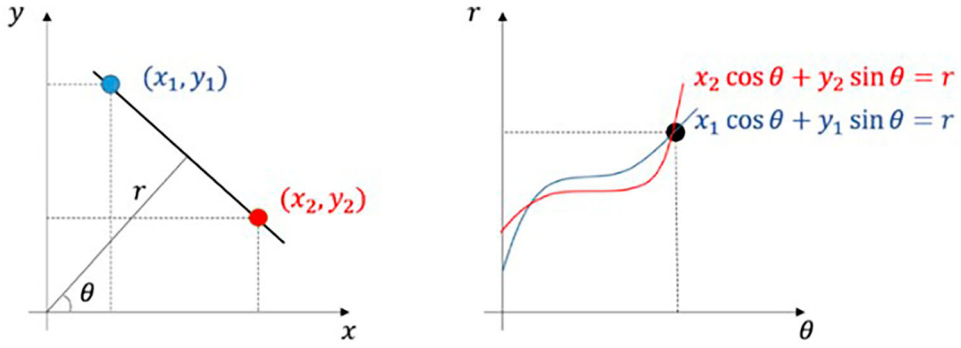

Figure 6. Mapped out point represented in the hough space.

Figure 6 displays the mapped-out point that is depicted in the rough space. In a θ and r interval, accumulator cell values that surpass the threshold are regarded as true lines. Because power line cables are straight, the hough transform has been used extensively in transmission line detection because it can effectively segment the cable edges from an image [21,48]. However, since edge images are rarely noise-free, the hough transform may result in unwanted line detections that call for additional line filtering, depending on parameter values.

$$x_i \cos\theta + y_i \sin\theta = r \tag{8}$$

3. Results and discussion

The study on power line detection utilized Infrared (IR), Visible Light (VL) datasets and implemented the backbone architecture on the TensorFlow 1 API Model, a deep learning framework [34]. The experiments were conducted on a Windows 10 system with 8.00GB RAM and an Intel Core i7-1185G7 CPU @ 3.00 GHz. The execution environment included Google Colab Pro [49] with a backend TPU (GPU_Python 3 Google Computer Engine, P100-PCIE-16GB). The VL-IR dataset [7] underwent image transformations, involving canny edge detection, morphological transformations, and otsu thresholding for images both with and without power lines. The models were trained separately using VL, IR, and the entire VL-IR dataset, divided in an 80:20 ratio. The RsurgeNet model and other existing models like AlexNet, Vgg16, Vgg19 and ResNet-50 were employed for binary classification and feature extraction. This classification predicted the presence of power lines as "1" and their absence as "0," considering training accuracy, training losses, validation accuracy, and validation losses. In another approach, the models extracted features from images, and the obtained feature maps were fed into different CAs such as AdaBoost, Light GBM, and XgBoost for image prediction. The fourth step involved applying Hough transform to test images to extract power lines. All these steps aimed to compute and analyze the convergence of power line image extraction and classification for VL, IR, and VL-IR datasets, providing an unbiased comparison. Throughout the model, numerous attributes could be adjusted and fine-tuned. The model's expressive capacity increased with the growing number of available parameters, but caution is exercised to avoid over fitting due to limited training data. The learning rate for weight optimization is 0.001 after 350 training steps. After the convolution layer and before the nonlinearity layers, BN is applied to prevent overfitting and accelerate hyper parameter optimization. A holdout validation using shuffling is conducted to obtain an average detection result for all models. Filters played a crucial role in feature extraction, collecting information from images in distinct ways. To enhance accuracy, reduce over fitting, and focus on learning underlying patterns, machine learning models needed optimized data. The evaluation of the RsurgeNet model's performance involved training on VL, IR, and VL-IR datasets separately, enabling an analysis of its behaviour under diverse image conditions.

The experimental phase explores three key stages of image transformation: canny edge detection, morphological transformation, and otsu thresholding. The results of these transformations are presented in three figures, namely Figure 7, Figure 8, and Figure 9, each corresponding to with and without power lines. Figure 7, providing a comprehensive understanding of the various stages of image transformation in the preprocessing pipeline for the IR dataset: In Figure 7, (a) and (a1) represents original images with power lines, capturing scenarios where power lines are present. (a2) corresponds to additional original images without power lines, contributing to a diverse dataset for training and evaluation. The Canny edge detection outputs highlight edges in different scenarios. Specifically, (b) and (b1) emphasizes edges in images with power lines, aiding in the detection of contours and boundaries associated with power lines. (b2) corresponds to the Canny edge output for a broader set of images without power lines, contributing to the model's understanding of edge features in diverse contexts. Subfigures (c) and (c1) showcase the results of morphological transformations on images with and without power lines, respectively. Morphological transformations are

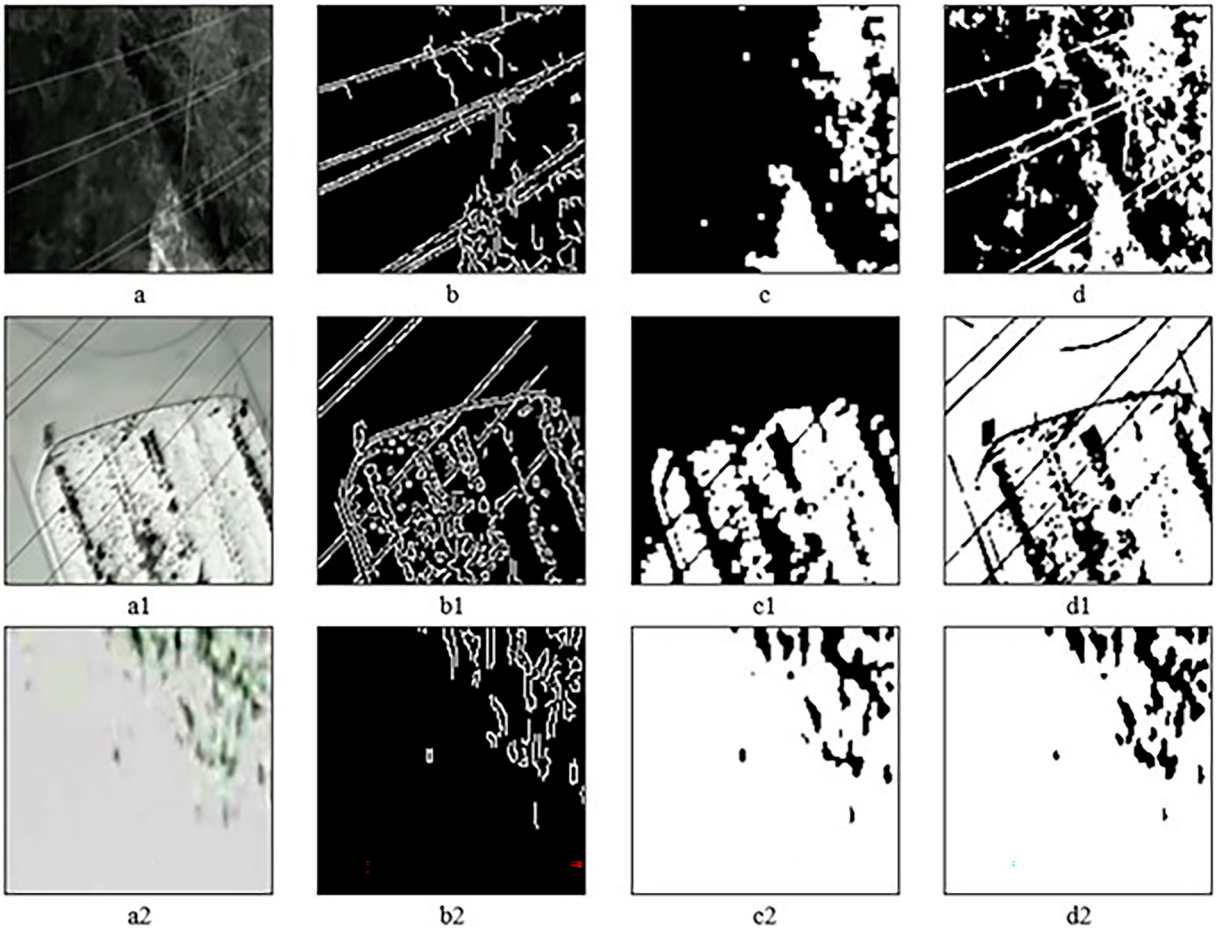

Figure 7. Preprocessed IR dataset segmentation output: (a)(a1) with PL a2) without PL, (b)(b1) canny edge output with PL, (b2) canny edge output without PL, (c)(c1) morphological transformation output with PL, (c2) morphological transformation output without PL, (d)(d1) otsu output with PL, (d2) otsu output without PL.

employed to refine the structural features of power lines and background elements. Specifically, (c) and (c1) represents the morphological transformation output for images with power lines, Additionally, (c2) illustrates the morphological transformation output for an extended set of images without power lines, contributing to a more comprehensive understanding of structural enhancements in diverse scenarios. The Otsu thresholding outputs provide binary representations for images with and without power lines. In (d), the binary representation distinguishes pixels belonging to power lines from the background in images with power lines. Similarly, (d1) achieves this binary distinction for images with power lines. (d2) extends this process to a broader set of images without power lines, contributing to the creation of clear boundaries and distinctions between power lines and the surrounding background. Figure 8. presents a detailed exploration of the image transformation stages within the preprocessing pipeline for the VL dataset. The starting point encompasses a, a1, and a2, with (a), (a1) featuring original images depicting scenarios with power lines, and (a2) introducing additional original images without power lines to enhance dataset diversity. Moving to the canny edge detection outputs, denoted as b, b1, and b2, (b), (b1) focuses on images with power lines, emphasizing detected edges crucial for contour identification, (b2) extends this output to a broader set of images without power lines, enhancing the model's grasp of edge characteristics in diverse contexts. Subsequently, morphological transformations, represented by c, c1, and c2, refine structural features in images with and without power lines. (c) and (c1) Illustrates morphological transformation output for images with power lines, (c2) extends this output to a broader set of images without power lines, providing insights into diverse structural enhancements. Finally, Otsu thresholding outputs, depicted as d, d1, and d2, generate binary representations for images with and without power lines. (d) and (d1)distinguishes pixels belonging to power lines from the background in images with power lines, (d2) extends this process to a broader set of images without power lines, contributing to the creation of clear boundaries and distinctions between power lines and the surrounding background. Collectively, these stages enrich the dataset, ensuring it is well-prepared for subsequent machine learning tasks in accurately discerning power lines in varied scenarios. Figure 9, presents the segmentation output after preprocessing for the VL-IR(8000) dataset, introducing a more complex dataset

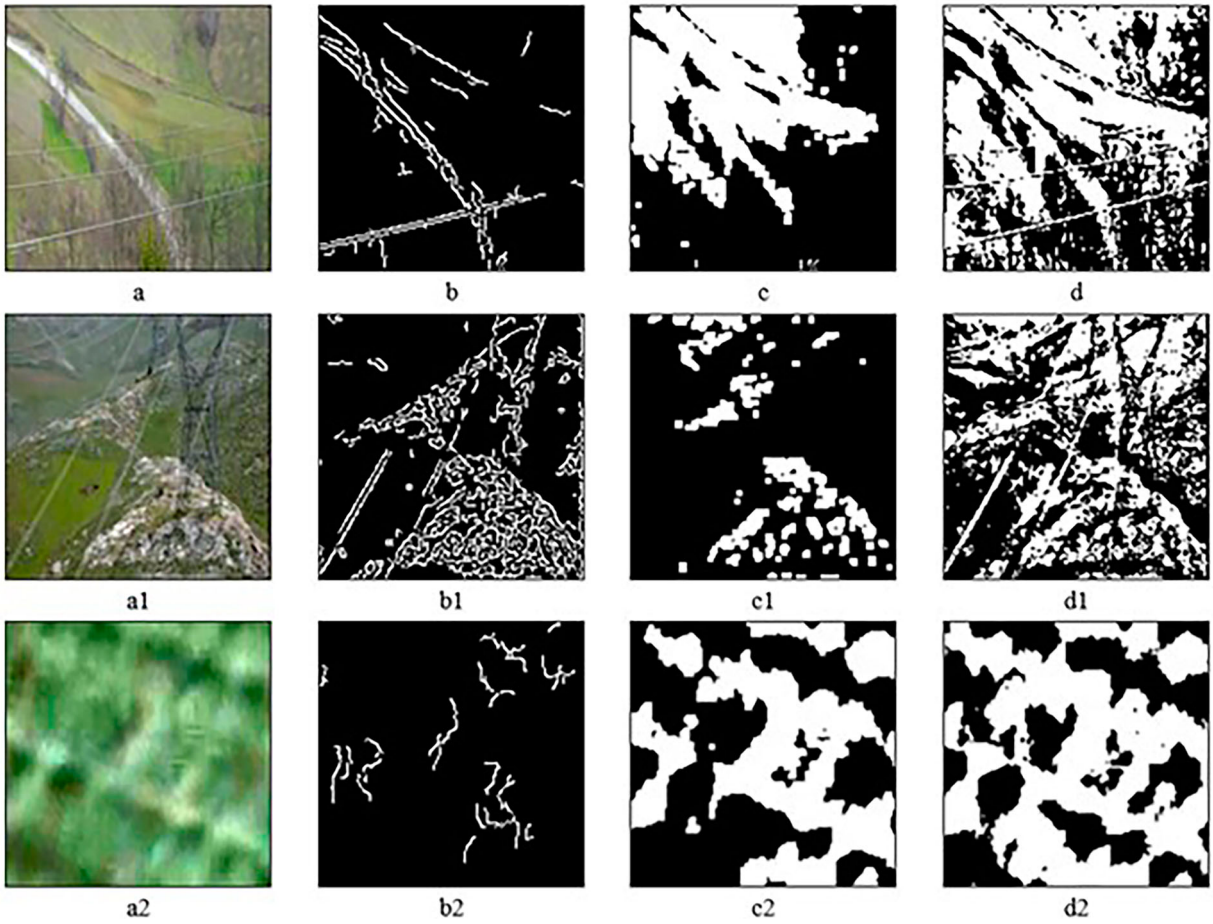

Figure 8. Preprocessed VL dataset segmentation output: (a)(a1) with PL, a2) without PL, (b)(b1) canny edge output with PL, (b2) canny edge output without PL, (c)(c1) morphological transformation output with PL, (c2) morphological transformation output without PL, (d)(d1) otsu output with PL, (d2) otsu output without PL.

that combines scenarios from both the IR and VL datasets. The set of original images (a, a1, a2, a3) captures scenarios with and without power lines for both datasets, enriching the training data. The canny edge detection outputs (b, b1, b2, b3) emphasize edges in diverse scenarios, ensuring the model's adeptness in identifying contours across a broader spectrum. Morphological transformations (c, c1, c2, c3) refine structural features for both IR and VL datasets, contributing to a nuanced understanding of power line shapes. Otsu thresholding outputs (d, d1, d2, d3) create binary representations, facilitating clear distinctions between power lines and the background in both datasets. This comprehensive preprocessing prepares the model for discerning power lines in a diverse and intricate dataset, incorporating variations from both IR and VL scenarios. Secondly, the experimental phase involved the application of the RsurgeNet model as a binary classifier to determine the presence or absence of power lines in the VL, IR, and combined VL-IR datasets. The outcomes of this experiment are visually depicted in Figure 10, Figure 11, and Figure 12, each representing the model's training evaluation metrics under different datasets. In Figure 10, the focus is on the evaluation metrics for the RsurgeNet model trained on IR images. Subplot (a) visually presents the training and validation accuracy, offering insights into the model's effectiveness in identifying power lines. Subplot (b) complements this by showcasing the training and validation loss, providing a glimpse into the model's convergence and learning behaviour throughout the training process on IR images. Moving to Figure 11, the evaluation metrics for the RsurgeNet model trained on VL images are depicted. Subplot (a) illustrates the training and validation accuracy, reflecting the model's capability in discering PLs in VL dataset. Simultaneously, Subplot (b) reveals the corresponding training and validation loss, offering a perspective on the model's convergence and overall performance during training on VL images. Figure 12, delves into the model training evaluation metrics for the RsurgeNet model trained on the combined VL-IR dataset. Subplot (a) visually represents the training and validation accuracy, highlighting the model's performance in scenarios that involve both Visible Light and Infrared images. Subplot (b) illustrates the corresponding training and validation loss, providing insights into the convergence behaviour and overall performance of the model in the combined dataset, Figure 10, Figure 11, and Figure 12, collectively offer a comprehensive insight into the performance

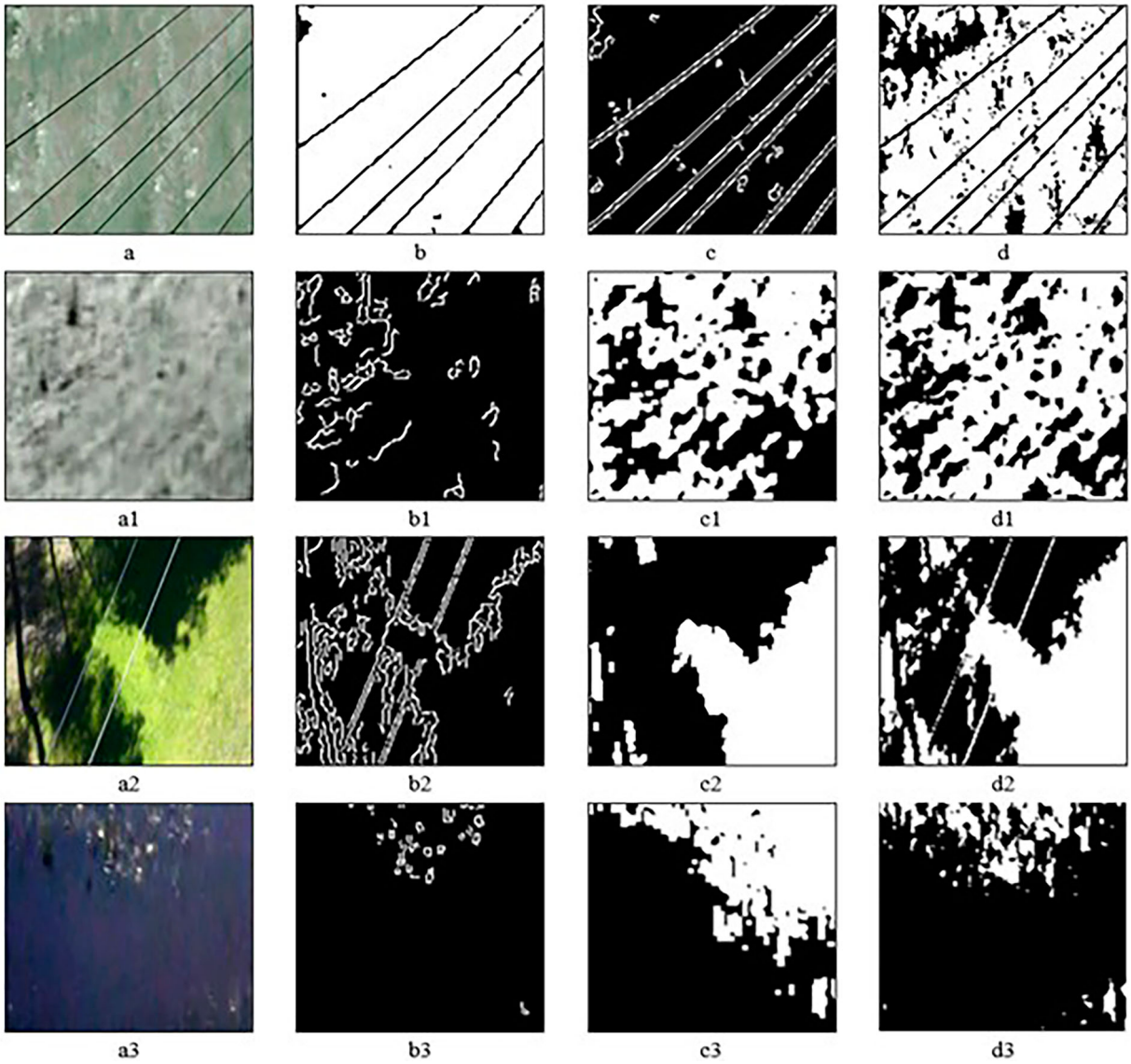

Figure 9. Preprocessed VL-IR dataset segmentation output: (a)(a1) original images with and without PL for IR, (a2)(a3) with and without PL for VL (b)(b1) canny edge output with and without PL for IR, (b2)(b3) canny edge output for with and without PL for VL, (c)(c1) morphological transformation output with and without PL for IR, (c2)(c3) morphological transformation output for with and without PL for VL, (d)(d1) otsu output with and without PL for IR, (d2)(d3) otsu output with and without PL for VL.

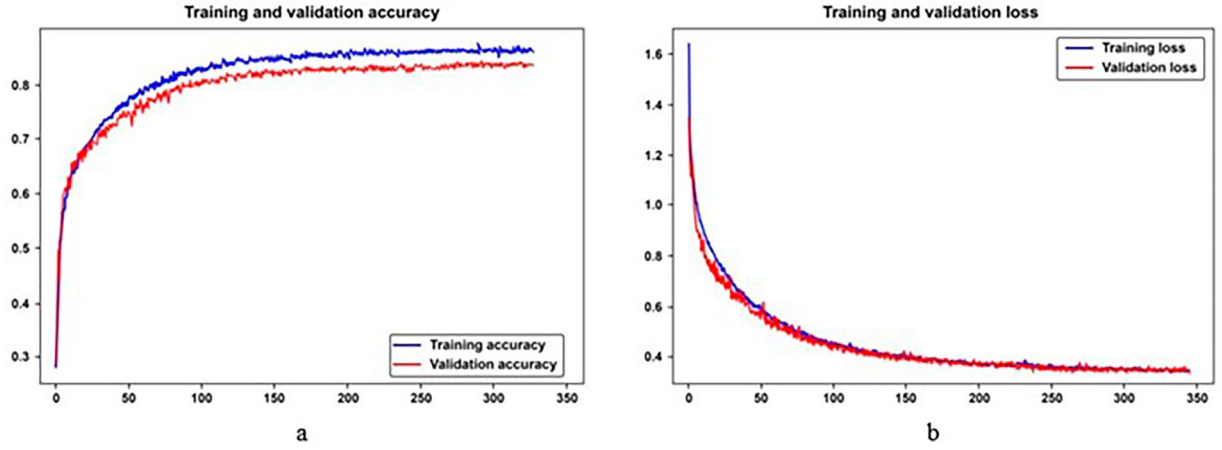

Figure 10. Model training evaluation metrics for IR images (a) training and validation accuracy (b) training and validation loss.

of the RsurgeNet model as a binary classifier across distinct datasets. These visualizations contribute to a detailed understanding of the model's accuracy, loss, and convergence patterns during both training and validation phases, providing valuable insights into its effectiveness in identifying power lines across diverse

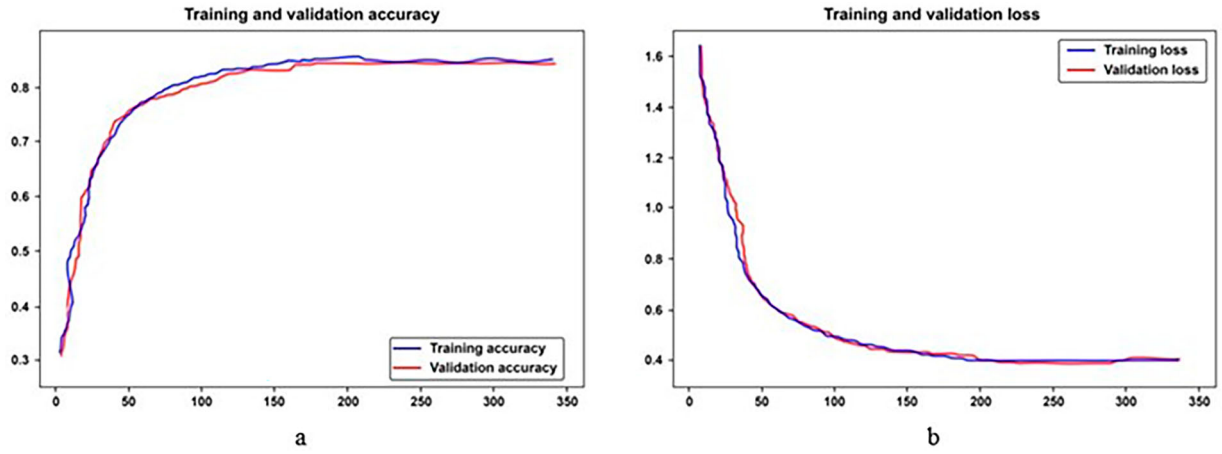

Figure 11. Model training evaluation metrics for VL images (a) training and validation accuracy (b) training and validation loss.

imaging scenarios. Additionally, the process of removing redundant and unnecessary data from the dataset plays a crucial role in enhancing the model's ability to discern relationships. This optimization leads to higher accuracy, faster learning times, and reduced computational complexity. Furthermore, the streamlined dataset contributes to improved generalization and efficiency in model training and validation processes. The plotted graphs illustrate the accuracy and loss metrics for the model evaluations conducted on the VL, IR, and combined VL-IR datasets during both the training and validation phases. Each dataset's binary classification Receiver Operating Characteristic (ROC) curve is visually presented in Figure 13, offering a comparative analysis of the model's performance in discerning power lines across diverse imaging scenarios. The ROC curve provides valuable insights into the model's ability to balance true positive rate and false positive rate, aiding in the assessment of its overall discriminatory power in differentiating between the presence and absence of power lines. Figure 13, serves as a comprehensive visualization, summarizing the model's performance across the VL, IR, and VL-IR datasets, facilitating a holistic understanding of its effectiveness binary classification tasks.

Table 1, presents a comprehensive analysis of the classification output of the RsurgeNet model using three different datasets: IR with 4000 images, VL-IR with 8000 images, and VL with 4000 images. The key metrics include accuracy, threshold, confusion matrix, Area Under the Curve (AUC), model parameters, and time duration for one epoch. For the IR dataset, the RsurgeNet model achieved a remarkable training accuracy of 91.874%. The confusion matrix indicates that the model made 385 correct predictions for power lines and 22 false positives. The AUC, measuring the model's ability to discriminate between positive and negative instances, is high at 0.9856. The threshold for decision-making is set at 0.1689. The parameter count for the model is 253953, and the training time for one epoch is 206 s over 20 epochs. In the case of the VL dataset, the RsurgeNet model achieved an accuracy of 87.74%. The confusion matrix displayed 358 correct predictions for power lines and 49 false positives. The AUC is calculated at 0.9488, demonstrating a robust discriminatory capability. The model used a threshold of 3.15717 for decision-making. The total number of parameters in the model is 253953, and the training time for one epoch is 211 s over 34 epochs. For the combined VL-IR dataset with 8000 images, the RsurgeNet model achieved an accuracy of 85.63%. The confusion matrix revealed 727 correct predictions for power lines and 95 false positives. The AUC is 0.96112, indicating a strong discriminatory power. The threshold for decision-making is 0.765901. The model had 253953 parameters, and the training time for one epoch is 430 s over 17 epochs. Additionally, the table includes results from other architectures, namely VGG16 and VGG19, with segmentation. The results highlight the superior performance of the RsurgeNet model, especially in the IR dataset, showcasing its robustness, interpretability, flexibility, and overall effectiveness in power line classification tasks. The detailed metrics provide a thorough understanding of the model's performance across different datasets and architectures. Table 2, provides an extensive examination and comparison of various CNN models utilizing distinct CAs across three datasets: VL with 4000 images, IR with 4000 images, and the combined VL-IR with 8000 images. One noteworthy observation is the substantial performance improvement exhibited by the proposed model, RsurgeNet, when applied to the VL-IR dataset of 8000 images, achieving an impressive accuracy of 97.8%. This marks a significant enhancement from its previous performances of 87.1%, 80.1%, 79.5%, and 97.43% [32] when validated with other CAs. In the IR dataset (4000 images), AdaBoost achieved an accuracy of 48.75%, with a time duration of 206 s for one epoch. The accuracy significantly improved to 64.75% for the more extensive VL-IR dataset (8000

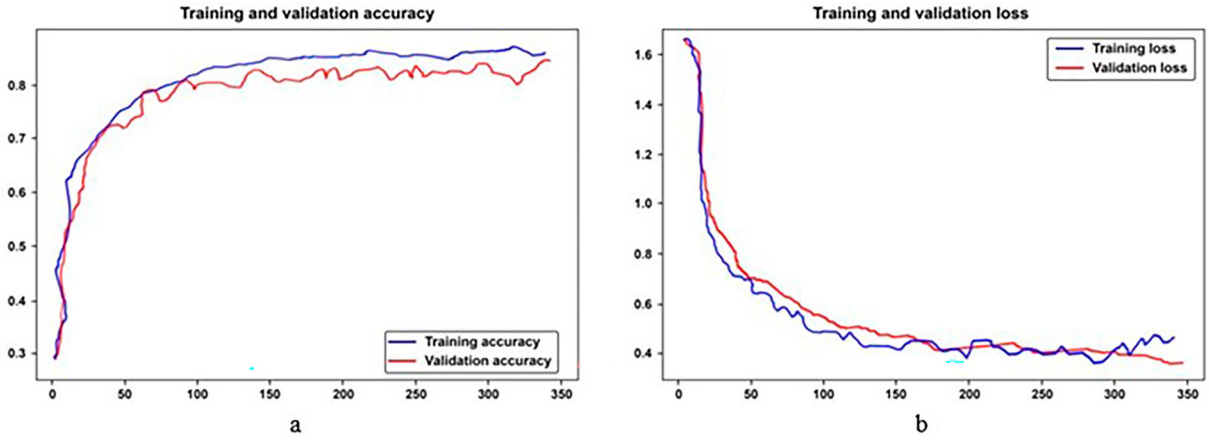

Figure 12. Model training evaluation metrics for VL-IR images (a) training and validation accuracy, (b) training and validation loss.

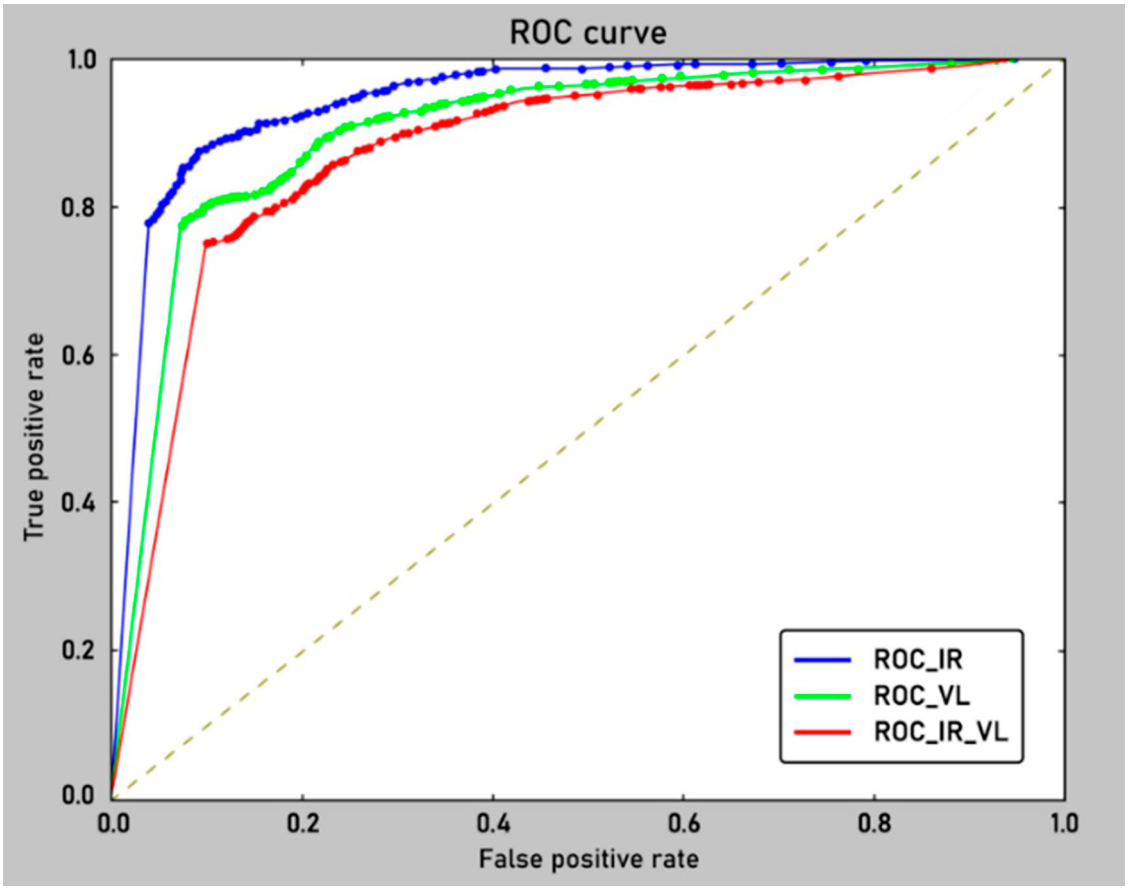

Figure 13. ROC curve comparison for VL, IR and VL-IR image.

images), taking 431 s. For the VL dataset (4000 images), AdaBoost demonstrated an accuracy of 52.6%, and for the VL-IR dataset, it further improved to 75.37%. For the IR dataset, Light GBM achieved an accuracy of 50%, taking 179 s for one epoch. The accuracy increased to 73% for the VL-IR dataset, requiring 430 s. In the VL dataset, Light GBM showed an accuracy of 54.3%, and for the VL-IR dataset, it further improved to 76.8%. XgBoost achieved an accuracy of 51.3% for the IR dataset, taking 138 s for one epoch. The accuracy significantly improved to 79.5% for the VL-IR dataset, with time duration of 353 s. In the VL dataset, XgBoost demonstrated an accuracy of 55.62%, and for the VL-IR dataset, it further improved to 97.8%. Discussing the inference time, Table 1 reveals that running the experiment with RsurgeNet with binary classification takes 430 s for the VL-IR dataset, slightly longer than the 206 s and 211 s required for the IR and VL datasets, respectively. This emphasizes the trade-off between model complexity and computational efficiency, especially in handling the more intricate VL-IR dataset. Table 2 provides a comprehensive overview of a combined model employing various CAs, namely AlexNet, VGG16, ResNet-50, and RsurgeNet, coupled with boosting techniques such as AdaBoost, Light GBM, and XgBoost. Notably, the execution time per

epoch for RsurgeNet_VL-IR_XgBoost stands out at a significantly lower 152 s when compared to ResNet 50_VL-IR_XgBoost (183 s), Vgg16_VL-IR_XgBoost (256 s), and AlexNet_VL-IR_XgBoost (353 s). This underscores the efficiency of the proposed model, especially in conjunction with XgBoost. However, it's worth noting a drawback in AdaBoost, where the inference time lengthens with the number of estimators despite lower predicted image accuracy. In contrast, Light GBM emerges as a proficient algorithm for managing large datasets, offering accurate predictions even with larger estimators and faster image classification compared to AdaBoost. XgBoost, with its advantages of low bias, low variance, short inference time, and high accuracy of 97.8%, excels in predicting images rapidly. In the context of power line (PL) prediction, XgBoost outperforms the other two algorithms, namely LightGBM and AdaBoost. According to the presented tables, the proposed models, RsurgeNet with binary classification and RsurgeNet with XgBoost, outshine other models across various metrics, including AlexNet_AdaBoost, AlexNet_LightGBM, AlexNet_XgBoost, VGG16_AdaBoost, VGG16_LightGBM, VGG16_XgBoost, ResNet-50_AdaBoost, and ResNet-50_LightGBM. Interestingly, ResNet-50_XgBoost performs poorly despite the incorporation of feature extraction. Moreover, the confusion matrix (CM) [17] details the performance of each CA on different datasets, offering insights into true positives, false positives, true negatives, and false negatives. This matrix provides a comprehensive evaluation of the model's classification accuracy and errors across diverse scenarios and are depicted in Table 1 and Table 3. The presented figures, namely Figure 14, Figure 15, and Figure 16, offer a glimpse into the model's validation accuracy on different datasets, providing visual insights into its performance. In these visualizations, a small sample of the model's predictions is showcased, emphasizing the correspondence between the predicted labels ("0" for no power line and "1" for power line) and the actual labels, representing the ground truth of each image Figure 14, specifically captures the model's predictions on the IR dataset. The actual label values of "0" or "1" indicate whether a power line is absent or present, and the prediction label values are compared to validate the accuracy of power line detection. This visual representation aids in assessing the model's proficiency in correctly classifying images within the IR dataset. Moving to Figure 15, the focus shifts to predictions on the VL dataset. Similar to the previous figure, the alignment between predicted and actual labels is crucial for evaluating the model's accuracy in identifying power lines within the context of the VL dataset. The visual depiction provides a clearer understanding of the model's performance in diverse scenarios. Figure 16 extends this analysis to predictions on the VL-IR dataset. This figure encapsulates the model's ability to handle the

Table 1. Performance analysis of various CNN models and its comparison with the proposed RsurgeNet approach FR the VL-IR dataset.

Architecture	Dataset	Accuracy	Threshold	Confusion matrix			AUC	Time duration for 1 Epoch in secs	Parameters
				Actual/Predicted	True +VE	False +VE			
VGG16(Binary Classification)	IR	49.125	1.502217	FALSE -VE	407	0	0.5	256	6,50,57,473
				TRUE -VE	393	0			
	VL	49.125	1.502217	FALSE -VE	407	0	0.5	287	6,50,57,473
				TRUE -VE	393	0			
VGG19(Binary Classification)	IR	49.125	1.501603	FALSE -VE	407	0	0.5	307	70367169
				TRUE -VE	393	0			
	VL	49.125	1.501603	FALSE -VE	407	0	0.5	321	70367169
				TRUE -VE	393	0			
RSURGENET (Binary Classification)	IR(4000)	91.874	0.1689	FALSE -VE	385	22	0.9856	206	253953
				TRUE -VE	22	371			
	VL(4000)	87.74	3.15717	FALSE -VE	358	49	0.9488	211	253953
				TRUE -VE	49	344			
	(VL_IR)8000	85.63	0.765901	FALSE -VE	727	95	0.96112	430	253953
				TRUE -VE	92	694			

Table 2. The performance analysis and comparison of CNN models with CA for the VL-IR datasets.

Classification algorithm	Dataset	ALEXNET		VGG16		RESNET 50		RSURGENET		Threshold
		Accuracy	Time Duration for 1 epoch in (Secs)	Accuracy	Time Duration for 1 epoch in (Secs)	Accuracy	Time Duration for 1 epoch in (Secs)	Accuracy	Time Duration for 1 epoch in (Secs)	
AdaBoost	IR (4000)	48.75	206	49.87	177	51.1	115	56.6	91	1.5
	VL (4000)	52.6	317	55.87	273	56.37	154	57.6	117	1.5
	VL_IR (8000)	64.75	431	65	375	74.8	268	75.37	171	0.965
Light GBM	IR (4000)	50	179	52.875	153	53.62	111	60.3	84	1.5
	VL (4000)	54.3	287	54.625	187	54.8	162	62.7	98	1.5
	VL_IR (8000)	73	430	75.25	291	75.6	198	76.8	169	0.965
XgBoost	IR (4000)	51.3	138	60.8	112	66.8	93	76.6	83	1.5
	VL (4000)	55.62	223	73.5	144	82.1	104	85.5	92	1.5
	VL_IR (8000)	79.5	353	80.1	256	87.1	183	97.8	152	0.965

combined IR and VL dataset, showcasing its performance in scenarios where both types of images are present. The visual examination of predicted labels against actual labels contributes to a nuanced evaluation of the model's classification and prediction capabilities across varied datasets. To further validate the model's results, the Hough transform is applied to the evaluated outputs, specifically focusing on the results containing power lines. This additional step aims to draw out the lines more distinctly, facilitating easier identification of power lines without unnecessary complexity. The incorporation of the Hough transform adds an extra layer of validation, enhancing the credibility of the classification and prediction model results.

The outcomes of the Hough transform applied to prediction samples containing only power lines are visually presented in Figure 17, Figure 18, Figure 19, Figure 20, and Figure 21, shedding light on the model's line detection capabilities in different scenarios. Figure 17, provides a detailed assessment of the RsurgeNet model's binary classification performance, particularly in detecting power lines within the IR dataset. The figure includes multiple subplots, each showcasing binary classified output images that offer insights into the line detection process. Subplots (a), (a1), and (a2) display binary classified output images for the IR dataset, demonstrating instances where the RsurgeNet model has performed binary classification to identify power lines. The variability seen in these images highlights the model's varying accuracy in detecting power lines across different scenarios within the IR dataset. Additionally, Similarly, (b), (b1), and (b2) illustrate binary classified output images for the VL dataset, indicating the model's performance in power line detection within VL imagery. These subplots enable a detailed analysis of the model's precision in detecting power lines in specific scenarios. Subplots (c), (c1), and (c2) extend the analysis to the IR-VL combination, presenting binary classified output images that demonstrate the RsurgeNet model's effectiveness in differentiating between power lines and their absence within a mixed dataset containing VL images. This inclusion adds complexity to the classification task, revealing insights into the model's capabilities under such conditions. By utilizing the robust Hough transform technique for line detection, the visual outputs in Figure 17, critically evaluate the model's accuracy in detecting power lines. Variations in accuracy across different scenarios depicted in the binary classified output images offer valuable insights into the model's strengths and areas for potential improvement. Overall, this comprehensive analysis enhances understanding of the RsurgeNet model's performance, particularly in its proficiency in detecting power lines within the IR dataset. The subsequent investigation reveals that the RsurgeNet model, coupled with the XgBoost classifier, achieves a noteworthy improvement of 97.8%

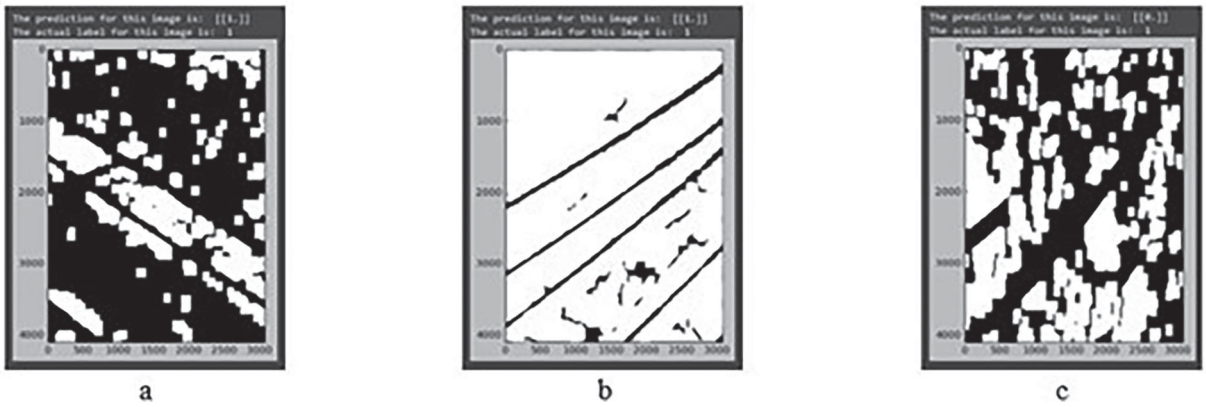

Figure 14. Predictions on the IR dataset.

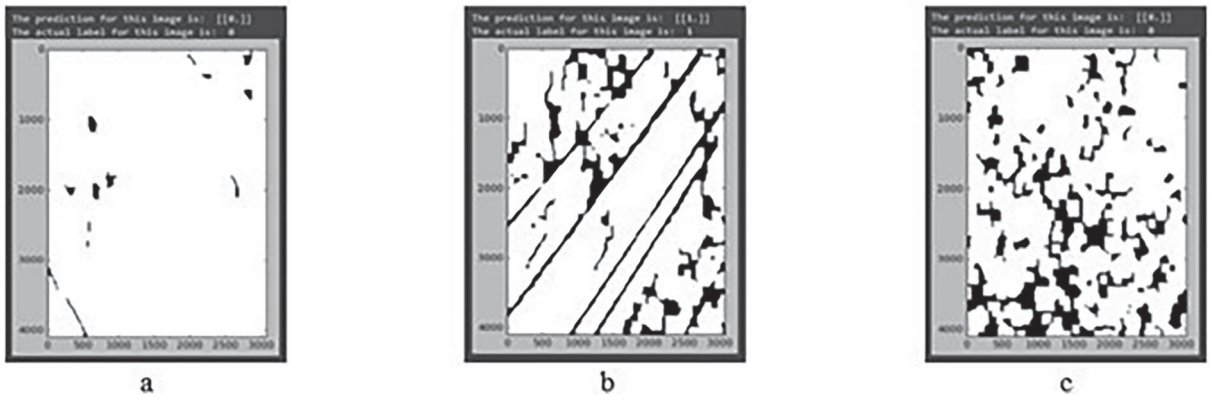

Figure 15. Predictions on the VL dataset.

Table 3. Analysis of confusion matrixes for models with CA algorithms.

			Confusion matrix							
			ALEXNET		VGG16		RESNET 50		RSURGENET	
Classification algorithm	Dataset		True +VE	False +VE	True +VE	False +VE	True +VE	False +VE	True +VE	False +VE
AdaBoost	IR (4000)	FALSE -VE	156	287	182	112	253	219	359	48
		TRUE -VE	123	234	289	217	172	156	299	94
	VL (4000)	FALSE -VE	312	288	247	191	249	126	342	65
		TRUE -VE	91	109	162	200	223	202	274	119
	VL_IR (8000)	FALSE -VE	252	552	581	401	685	249	517	361
		TRUE -VE	12	784	159	459	154	512	33	689
Light GBM	IR (4000)	FALSE -VE	232	384	174	153	267	192	278	129
		TRUE -VE	16	168	224	249	179	162	188	205
	VL (4000)	FALSE -VE	9	77	244	275	112	47	323	99
		TRUE -VE	288	151	88	193	314	327	199	179
	VL_IR (8000)	FALSE -VE	687	25	681	164	434	296	679	313
		TRUE -VE	407	481	232	523	93	777	57	551
XgBoost	IR (4000)	FALSE -VE	161	111	249	158	294	182	362	113
		TRUE -VE	278	250	155	238	83	241	74	251
	VL (4000)	FALSE -VE	244	167	391	168	398	106	431	62
		TRUE -VE	188	201	44	197	37	259	54	253
	VL_IR (8000)	FALSE -VE	683	83	724	70	721	52	783	25
		TRUE -VE	245	589	247	559	154	673	9	783

in accuracy compared to the existing method. This advancement is particularly evident when the features generated by the model are transferred to various classifiers. The corresponding outcomes of this enhancement are meticulously showcased in Figure 18, Figure 19, Figure 20, and Figure 21, each shedding light on the Hough transform line detection output for sample prediction power line images. Figure 18, delves into the Hough transform line detection output, specifically exploring scenarios where AlexNet is employed in conjunction with three different classifiers (CA). Subplots (a) and (a1) represent IR images with power lines, providing a baseline for evaluating detection accuracy. Subplots (a2), and (a3) extend the analysis to VL images with power lines, offering additional perspectives. Furthermore, (a4) and (a5) introduces a VL-IR image with power lines, contributing to a more diverse dataset for comprehensive evaluation. The subsequent

subplots, denoted as [b-b5], depict preprocessed segmented output images, illustrating the intermediary steps in the image processing pipeline. These images serve as crucial inputs to the subsequent classifiers, capturing the refined representations of power lines after segmentation. Subplots [c-c5] showcase the output of Adaboost when combined with AlexNet. These visual representations provide insights into Adaboost

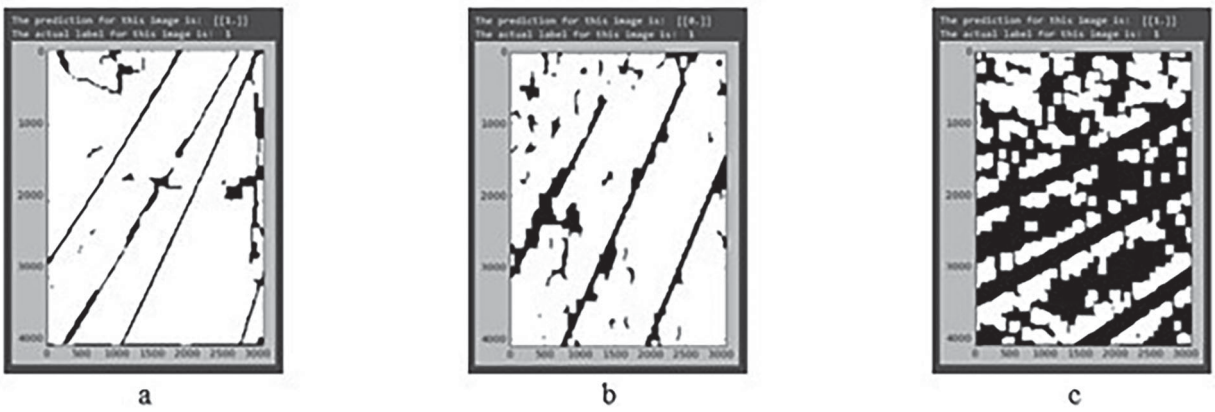

Figure 16. Predictions on the VL-IR dataset.

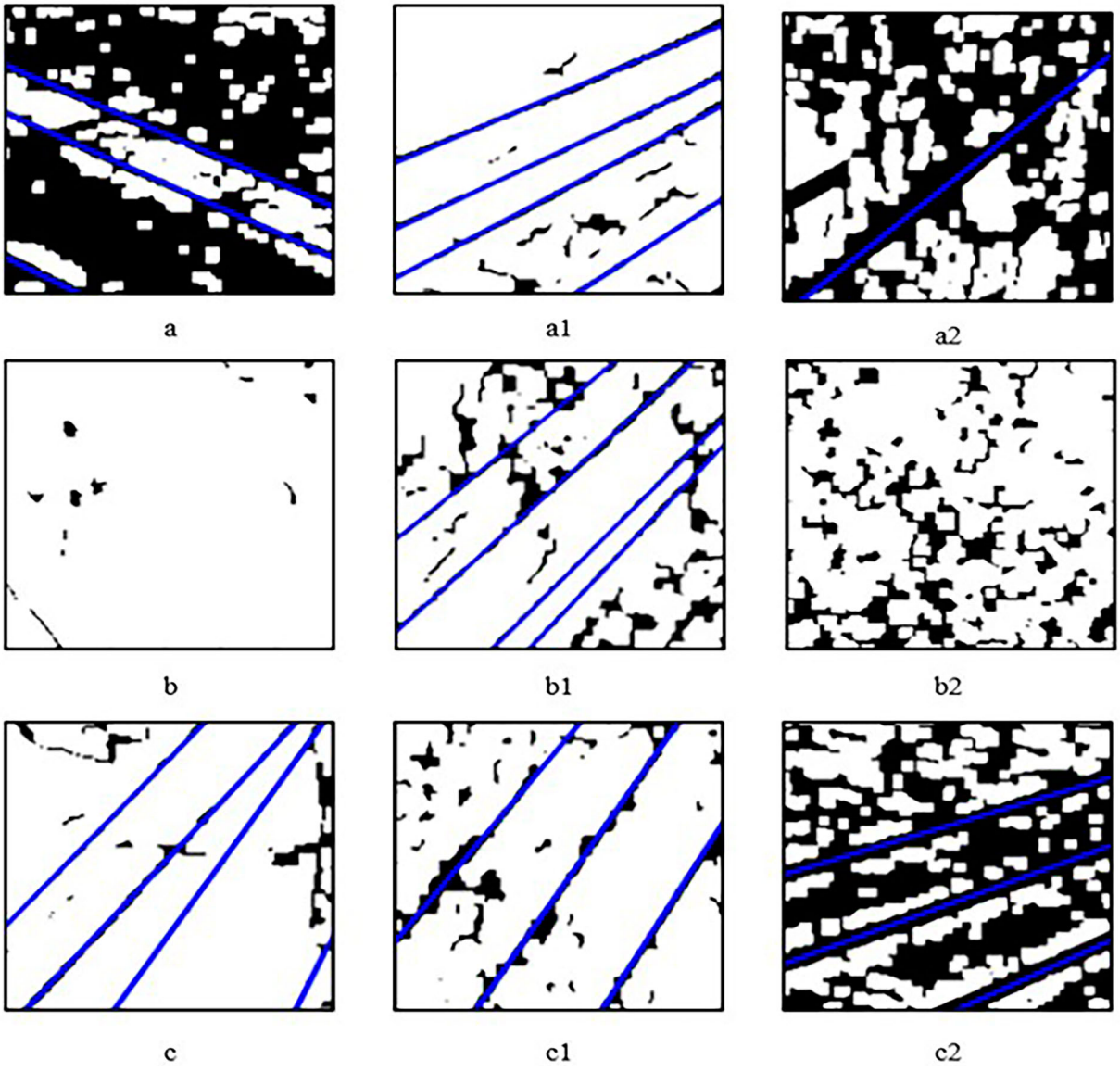

Figure 17. Hough transform line detection output for the sample prediction PL images. (a)(a1)(a2) binary classified output images for IR, (b)(b1)(b2) binary classified output images for VL, (c)(c1)(c2) binary classified output images for IR-VL combination.

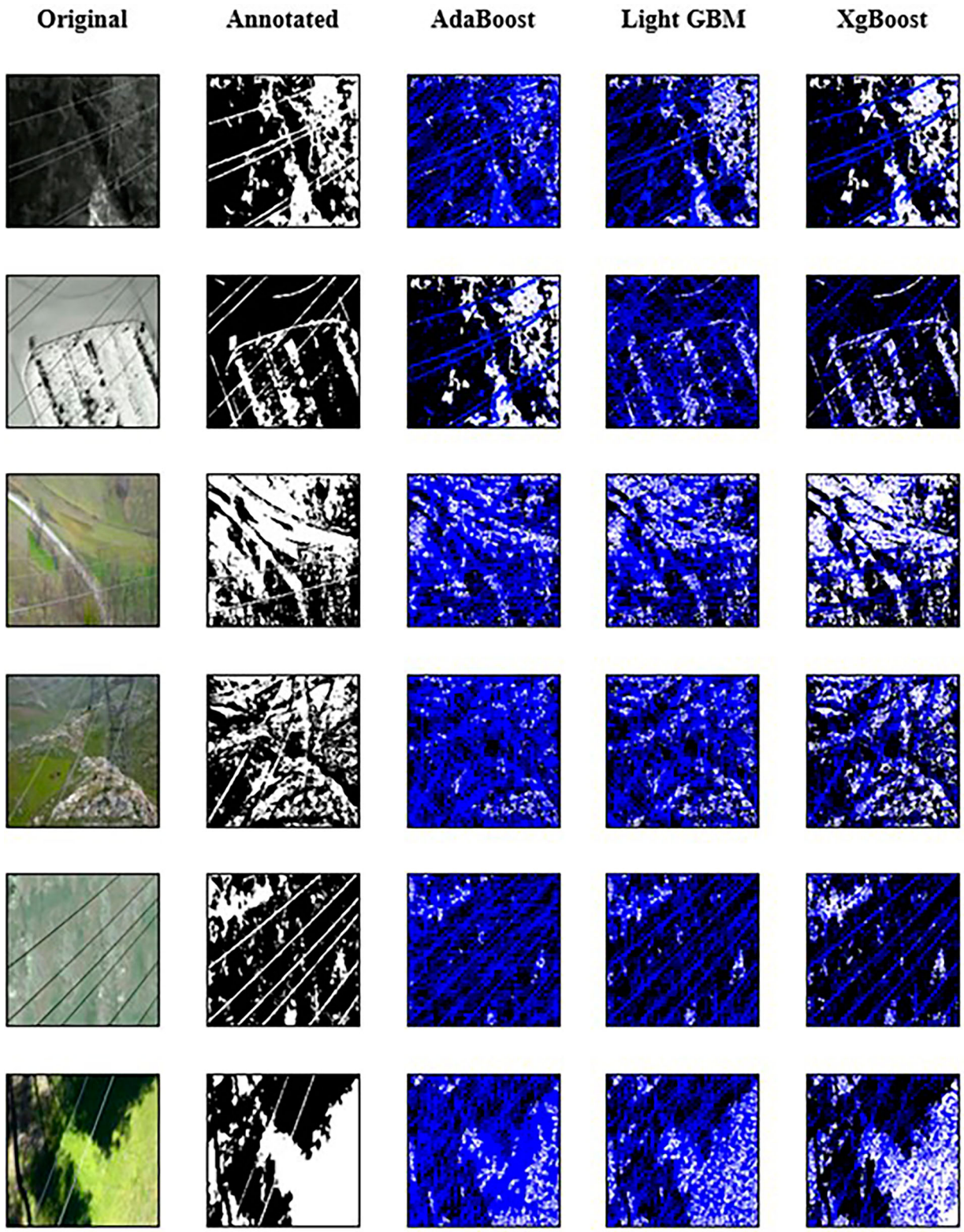

Figure 18. Hough transform line detection output for the sample prediction PL images of AlexNet with three CA: (a)(a1) IR images with PL, (a2)(a3) VL images with PL, (a4)(a5) VL-IR images with PL, [b-b5] preprocessed segmented output, [c-c5] output of Adaboost with AlexNet, [d-d5] output of LightGBM with AlexNet, [e-e5]output of XgBoost with AlexNet.

that interprets the features generated by AlexNet for power line detection. Similarly, [d- d5] and [e-e5] display the output of LightGBM and XgBoost respectively, when integrated with AlexNet. The detailed examination of Figure 18, contributes to a nuanced understanding of the RsurgeNet model, in tandem with XgBoost, outperforms the existing method, offering substantial improvements in accuracy. This analysis is pivotal for assessing the efficacy of the proposed model in diverse scenarios, providing valuable insights for further optimization and refinement. In Figure 19, the Hough transform line detection output for predicted

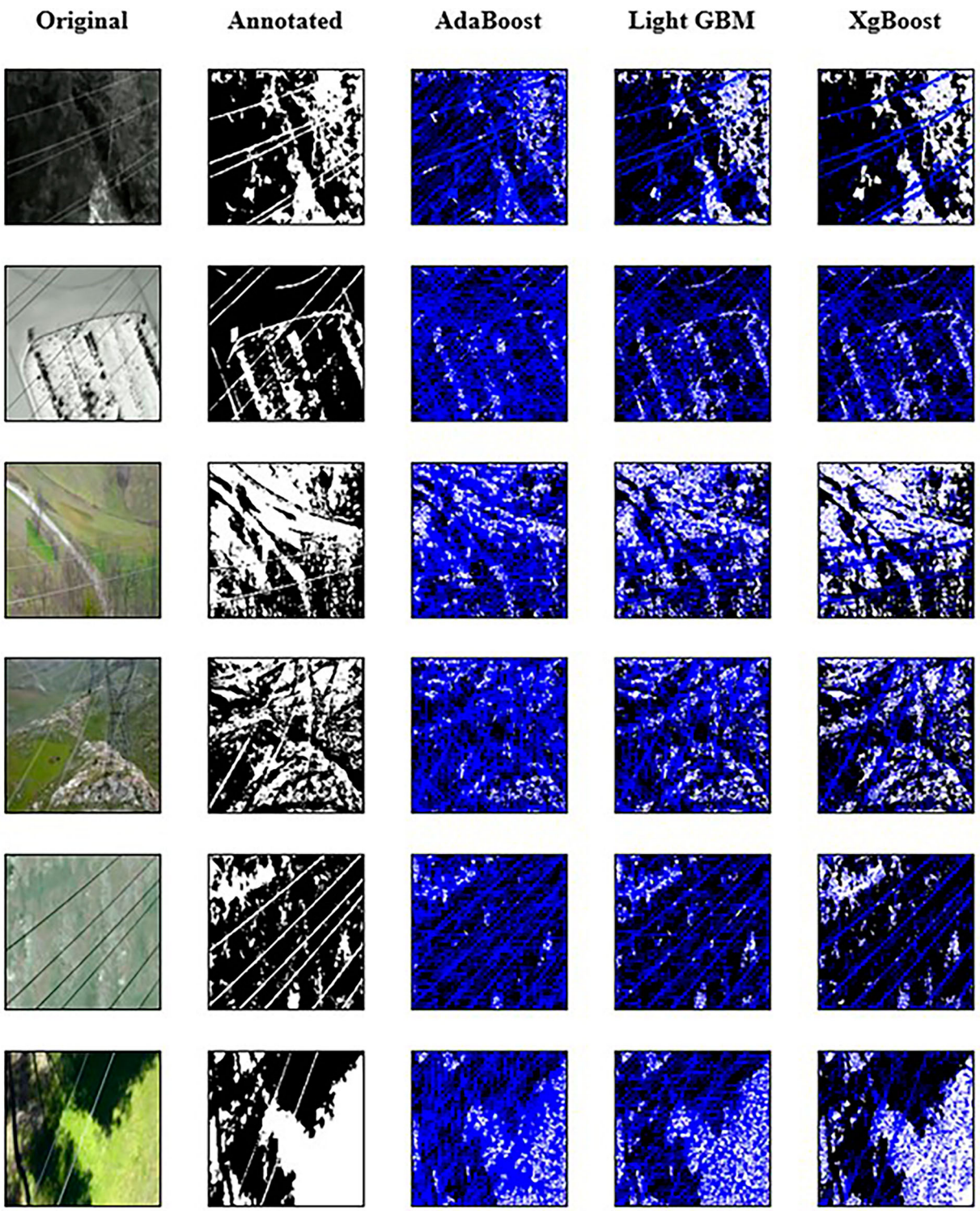

Figure 19. Hough transform line detection output for the sample prediction PL images of VGG16 with three CA: (a)(a1) IR images with PL, (a2)(a3) VL images with PL, (a4)(a5) VL-IR images with PL, [b-b5] preprocessed segmented output, [c-c5] output of Adaboost with VGG16, [d-d5] output of LightGBM with VGG16, [e-e5] output of XgBoost with VGG16.

PL images using VGG16 with three distinct CA is presented in a detailed breakdown. Subfigures (a) and (a1), depicts the images for IR with power lines, visually representing the model's predictions on scenarios with IR imagery featuring power lines. On the other hand, subfigures (a2), (a3) and (a4) showcase the images for VL and VL-IR scenarios with power lines, providing visual insights into VGG16 interpretation and classification of PL features in datasets containing both VL and IR images. The set of subfigures [b-b5] displays preprocessed segmented output, illustrating the results of segmenting the images to highlight PL features. These images help visualize the structural enhancements and modifications applied during the preprocessing stage. Subsequently, subfigures [c-c5] show the output of Adaboost with VGG16, demonstrating the way of Adaboost interpretation and categorizes the PL images generated by VGG16, contributing to

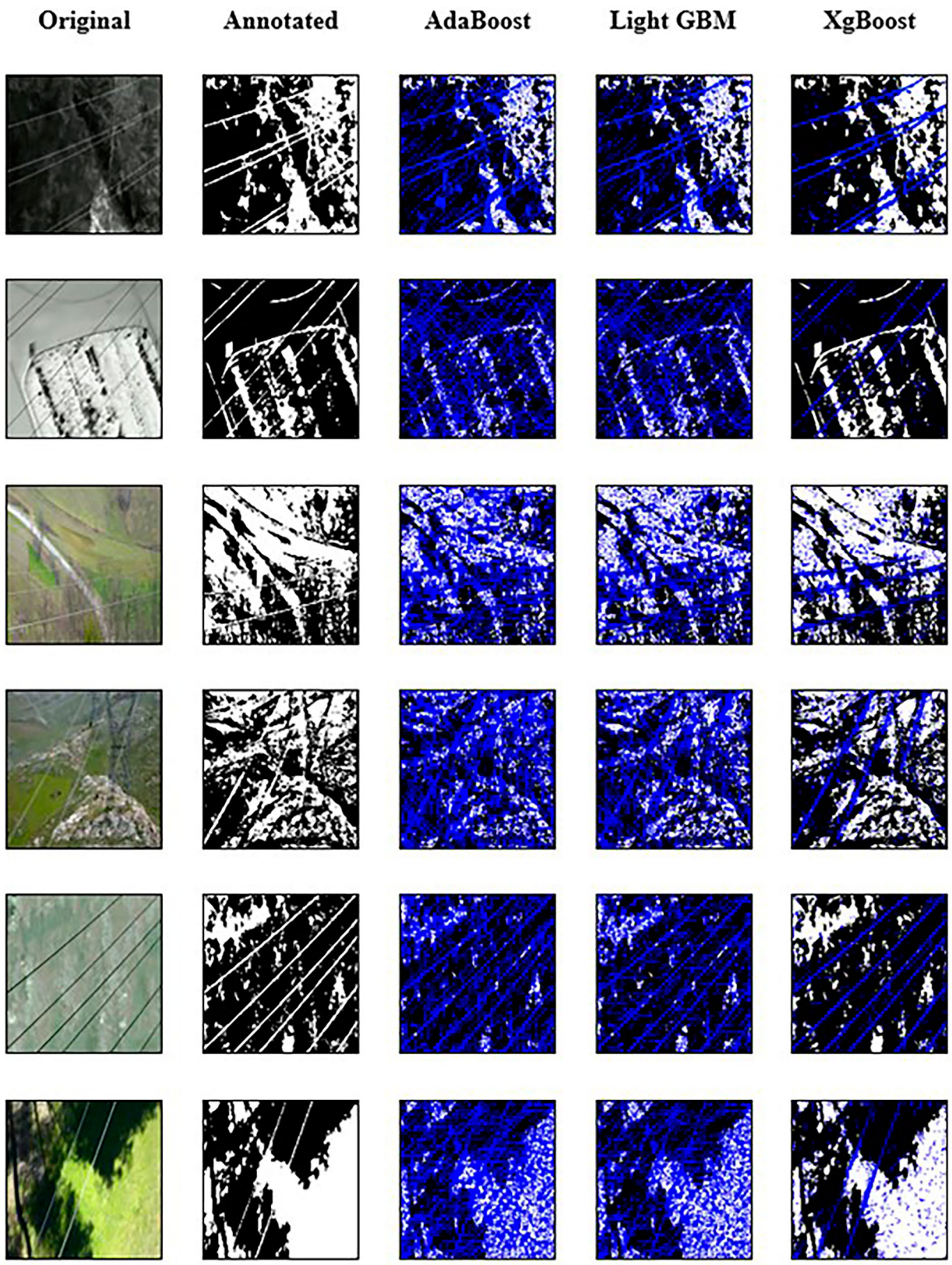

Figure 20. Hough transform line detection output for the sample prediction PL images of ResNet-50 with three CA:(a)(a1) IR images with PL, (a2)(a3) VL images with PL, (a4)(a5) VL-IR images with PL, [b-b5] preprocessed segmented output, [c-c5] output of Adaboost with ResNet-50, [d-d5] output of LightGBM with ResNet-50, [e-e5] output of XgBoost with ResNet-50.

a comprehensive understanding of the classification outcomes. Moreover, subfigures [d-d5] present the output of LightGBM with VGG16, illustrating the way of LightGBM processes and classifies the PL images generated by VGG16, providing insights into the algorithm's interpretation and classification performance. Finally, subfigures [e-e5] showcase the output of XgBoost with VGG16, depicting the results of employing XgBoost as the classification algorithm for the features generated by VGG16. The images illustrate the way in

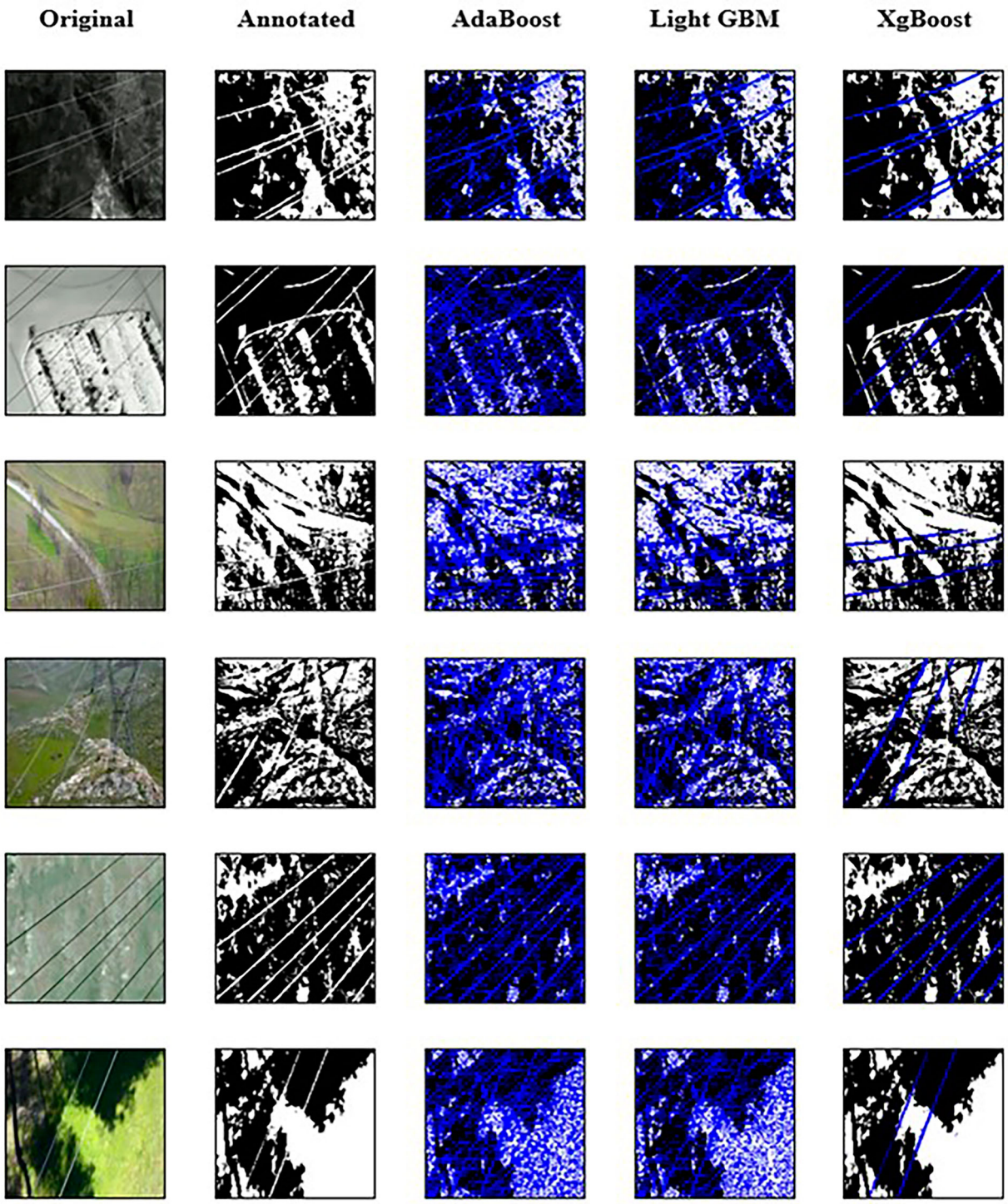

Figure 21. Hough transform line detection output for the sample prediction PL images of RsurgeNet with three CA: (a)(a1) IR images with PL, (a2)(a3) VL images with PL, (a4)(a5) VL-IR images with PL, [b-b5] preprocessed segmented output, [c-c5] output of Adaboost with RsurgeNet, [d-d5] output of LightGBM with RsurgeNet, [e-e5] output of XgBoost with RsurgeNet.

which XgBoost interprets and categorizes PL images, offering a detailed examination of the model's performance with different classification algorithms on IR and VL-IR datasets. In Figure 20, a comprehensive analysis of the Hough transform line detection output for predicted PL images is presented, focusing on the implementation of ResNet-50 with three distinct CA. Subfigures (a) and (a1) illustrate the images for IR scenarios with power lines, showcasing the model's predictions on infrared (IR) imagery featuring power lines. Additionally, subfigures (a2), (a3), (a4), and (a5) present the images for VL and VL-IR scenarios with power lines, allowing for a visual understanding of ResNet-50,which interprets and classifies PL features in datasets containing both VL and IR images. The subsequent set of subfigures [b-b5] displays preprocessed segmented output, offering insights into the results of segmenting the images to emphasize PL features. These images provide a visual representation of the structural enhancements and modifications applied during the

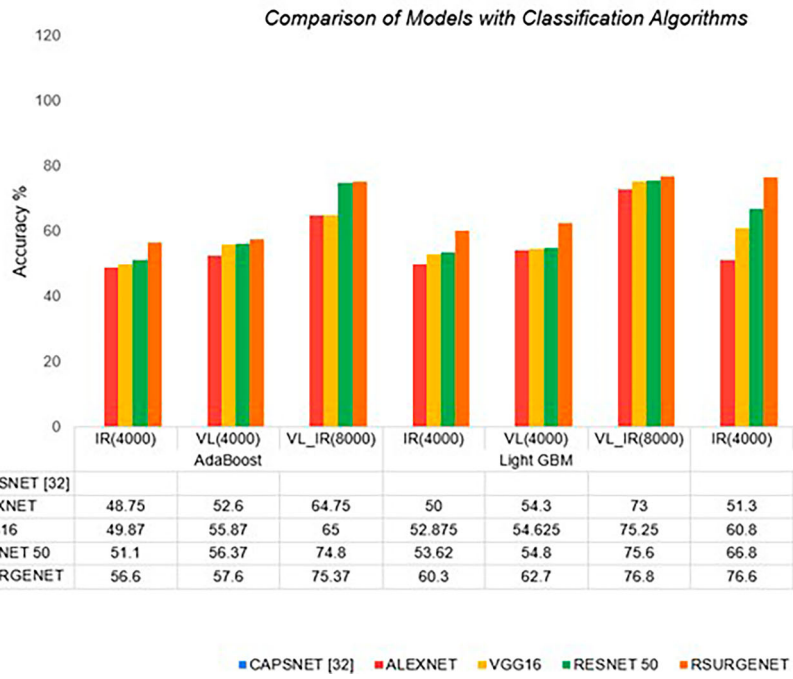

Figure 22. Performance Indices of the proposed topology with other architecture.

preprocessing stage. Following this, subfigures [c-c5] exhibit the output of Adaboost with ResNet-50, revealing the interpretation and categorization of PL images generated by ResNet-50 through the Adaboost CA. This provides a detailed examination of the classification outcomes and the algorithm's performance. Moreover, subfigures [d-d5] showcase the output of LightGBM with ResNet-50, illustrating the way of LightGBM processes and classifies the PL images generated by ResNet-50. This visual representation contributes to an understanding of the algorithm's interpretation and classification performance in different scenarios. Lastly, subfigures [e-e5] depict the output of XgBoost with ResNet-50, offering insights into the results of employing XgBoost as the CA for the features generated by ResNet-50. These images provide a detailed examination of XgBoost interpretation and categorization of PL images, contributing to a comprehensive analysis of the model's performance with various CAs IR,VL and VL-IR datasets. In Figure 21, a detailed predicted power line images is presented, focusing on the implementation of RsurgeNet with three distinct CA, Subfigures (a) and (a1) illustrate binary classified output images for IR scenarios with power lines, showcasing the model's predictions on IR imagery featuring power lines. Additionally, subfigures (a2), (a3), (a4), and (a5) present binary classified output images for VL and VL-IR scenarios with power lines, allowing for a visual understanding of RsurgeNet interpretation and classification of power line features in datasets containing both VL and IR images. The subsequent set of subfigures [b-b5] displays preprocessed segmented output, offering insights into the results of segmenting the images to emphasize power line features. These images provide a visual representation of the structural enhancements and modifications applied during the preprocessing stage. Following this, subfigures [c-c5] exhibit the output of AdaBoost with RsurgeNet, revealing the interpretation and categorization of power line images generated by RsurgeNet through the CA. This provides a detailed examination of the classification outcomes and the algorithm's performance. Moreover, subfigures [d-d5] showcase the output of LightGBM with RsurgeNet, illustrating the way of LightGBM processes and classifies the PL images generated by RsurgeNet. This visual representation contributes to an understanding of the algorithm's interpretation and classification performance in different scenarios. Lastly, subfigures [e-e5] depict the output of XgBoost with RsurgeNet, offering insights into the results of employing XgBoost as the CA for the features generated by RsurgeNet. These images provide a detailed examination of XgBoost interpretation and categorization of PL images, contributing to a comprehensive analysis of the model's performance with various CAs on IR, VL and VL-IR datasets. Figure 22 offers a comprehensive overview of the performance metrics for four distinct models with classifiers utilized in this study, including CAPSNET [32], ALEXNET, VGG16, RESNET-50, and RSURGENET. The evaluation spans different datasets, namely IR (4000 images), VL (4000 images), and VL-IR (8000 images). The suggested topology, represented by RsurgeNet, emerges as particularly noteworthy for HighVoltage Transmission Line (HVTL) applications, demonstrating impressive accuracy, especially in the context of larger datasets. For the AdaBoost classifier applied to the IR dataset (4000 images), AlexNet achieved an accuracy of 48.75%, while Vgg16 and ResNet- 50 exhibited accuracies of

49.87% and 51.1%, respectively. In contrast, RsurgeNet outperformed them with an accuracy of 56.6%. Similar trends are observed in the VL (4000 images) and VL-IR (8000 images) datasets, solidifying RsurgeNet superiority in accuracy across various scenarios. When considering the Light GBM classifier, RsurgeNet consistently surpasses other models across the IR, VL, and VL-IR datasets, showcasing its effectiveness in diverse imaging scenarios. The same holds true for the XgBoost classifier, where RsurgeNet attains higher accuracy compared to other models across all datasets. Partial Least Squares Regression (PLSR) [32] applied to the VL-IR dataset (8000 images). Here, RsurgeNet stands out with an impressive accuracy of 97.8%, highlighting its proficiency in handling complex datasets relevant to HVTL. In summary, these results underscore the effectiveness of RsurgeNet across diverse datasets and its potential suitability for HVTL applications, particularly in scenarios involving larger datasets.

4. Conclusion

The pre-processing strategy played a pivotal role in significantly enhancing the classification model's performance. The Infrared (IR) and Visible Light (VL) dataset emerged as an ideal foundation for a robust power line detection system. The proposed model demonstrated noteworthy speed advantages over competitors, achieving an impressive accuracy of 91.874% in both training and validation sets, with consistent values for loss and ROC. Despite the achieved high accuracy, opportunities for improvement exist by capitalizing on the model as a feature extractor and seamlessly integrating extracted features into the CA. The RsurgeNet_XgBoost model notably outperformed its counterparts, exhibiting accelerated convergence and achieving a 97.8% accuracy on the VL-IR(8000 images) dataset with reduced inference time. Future research endeavours could enhance this work by delving into CNN frameworks with diverse architectures for the instance segmentation of power line images, integrating non-neural classifiers. Furthermore, broadening the scope to detect faults in power lines and insulators would contribute to a more comprehensive coverage for autonomous power line inspection.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

[1] Katrasnik J, Pernus F, Likar B. A survey of mobile robots for distribution power line inspection. IEEE Trans Power Deliv. 2010;25:485–493. doi:10.1109/TPWRD.2009.2035427

[2] Nayyerloo M, Chen X, Wang W, et al. Cable climbing robots for power transmission lines inspection, Mob Robot - State Art Land, Sea, Air, Collab Mission, 2009, http://hdl.handle.net/10092/4129.

[3] Zhang T, Li Q, Zhang Cs, et al. Current trends in the development of intelligent unmanned autonomous systems. Front Inf Technol Electron Eng. 2017;18:68–85. doi:10.1631/FITEE.1601650

[4] Zhang X, Yang W, Tang X, et al. A fast learning method for accurate and robust lane detection using two-stage feature extraction with YOLO v3. Sensors. 2018;18; doi:10.3390/s18124308

[5] Lee SJ, Yun JP, Choi H, et al. Weakly supervised learning with convolutional neural networks for power line localization, IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–8, 2017. doi:10.1109/SSCI.2017.8285410

[6] Yuan Z-W, Zhang J. Feature extraction and image retrieval based on AlexNet, Eighth International Conference on Digital Image Processing, pp. 10033, 2016. doi:10.1117/12.2243849

[7] Yetgin ÖE, Benligiray B, Gerek ÖN. Power line recognition from aerial images With deep learning. IEEE Trans Aerospace Electron Syst. 2019;55(5):2241–2252. doi:10.1109/TAES.2018.2883879

[8] Zhang H, Yang W, Yu H, et al. Detecting power lines in UAV images with convolutional features and structured constraints. Remote Sens. 2019;11(11):1342), doi:10.3390/rs11111342

[9] Vemula S, Frye M. Real-time PL detection system for an unmanned aircraft system, IEEE International Conference on Systems, Man, and Cybernetics, pp. 4493-4497, 2020. doi:10.1109/SMC42975.2020.9283354

[10] Solomatine DP, Shrestha DL. Adaboost.RT: a boosting algorithm for regression problems. IEEE Intern Joint Confer Neural Netw. 2004;1162(2):1163–1168. doi:10.1109/IJCNN.2004.1380102

[11] Sun X, Guo L, Zhang W, et al. Small aerial target detection for airborne infrared detection systems using LightGBM and trajectory constraints. IEEE J Select Topics Appl Earth Observ Remote Sens. 2021;14:9959–9973. doi:10.1109/JSTARS.2021.3115637

[12] Ma Y, Pan H, Qian G, et al. Prediction of transmission line icing using machine learning based on GS-XGBoost. J Sensors. 2022; doi:10.1155/2022/2753583

[13] Bi H, Sun J, Xu Z. A graph-based semisupervised deep learning model for PolSAR image classification. IEEE Trans Geosci Remote Sens. 2019;57(4):2116–2132. doi:10.1109/TGRS.2018.2871504

[14] Song B, Li X. Power line detection from optical images. Neurocomputing. 2014;129:350–361. doi:10.1016/j.neucom.2013.09.023

[15] Miralles F, Pouliot N, Montambault S. State-of-the-art review of computer vision for the management of power transmission lines, 3rd Int Conf Appl Robot Power Ind, 2014. doi:10.1109/CARPI.2014.7030068

[16] Matikainen L, Lehtomäki M, Ahokas E, et al. Remote sensing methods for power line corridor surveys. ISPRS J Photogramm Remote Sens. 2016;119:10–31. doi:10.1016/j.isprsjprs.2016.04.011

[17] Yang L, Fan J, Huo B, et al. PLE-Net: automatic power line extraction method using deep learning from aerial images. Expert Syst Applic. 2022;198:116771), doi:10.1016/j.eswa.2022.116771

[18] M Bhujade R, V A, S H, P B, Detection of powerlines in complex natural surroundings, Comput. Sci. Inf. Technol, Academy & Industry Research Collaboration Center, pp. 101–8, 2013. doi:10.5121/csit.2013.3910

[19] Candamo J, Kasturi R, Goldgof D, et al. Detection of thin lines using low-quality video from low-altitude aircraft in urban settings. IEEE Trans Aerosp Electron Syst. 2009;45:937–949. doi:10.1109/TAES.2009.5259175
[20] Tragulnuch P, Chanvimaluang T, Kasetkasem T, et al. High voltage transmission tower detection and tracking in aerial video sequence using object-based image classification. Int Conf Embed Syst Intell Technol Int Conf Inf Commun Technol Embed Syst. 2018: 1–4. doi:10.1109/ICESIT-ICICTES.2018.8442058
[21] Zhang J, Liu L, Wang B, et al. High speed automatic power line detection and tracking for a UAV-based inspection. Int Conf Ind Control Electron Eng. 2012: 266–269. doi:10.1109/ICICEE.2012.77
[22] Li Z, Liu Y, Walker R, et al. Towards automatic power line detection for a UAV surveil-lance system using pulse coupled neural filter and an improved Hough transform. Mach Vis Appl. 2010;21:677–686. doi:10.1007/s00138-009-0206-y
[23] Yan G, Li C, Zhou G, et al. Automatic extraction of power lines from aerial images. IEEE Geosci Remote Sens Lett. 2007;4:387–391. doi:10.1109/LGRS.2007.895714
[24] Jalil B, Leone GR, Martinelli M, et al. Fault detection in power equipment via an unmanned aerial system using multi modal data. Sensors. 2019;19:1–15. doi:10.3390/s19133014
[25] Li Y, Xiao Z, Zhen X, et al. Attentional information fusion networks for cross-scene power line detection. IEEE Geosci Remote Sens Lett. 2019;16:1635–1639. doi:10.1109/LGRS.2019.2903217
[26] Shaha M, Pawar M. Transfer learning for image classification, In Second international conference on electronics, communication and aerospace technology, IEEE, pp. 656–660, 2018. doi:10.1109/ICECA.2018.8474802
[27] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition, 3rd Int Conf Learn Represent ICLR 2015 - Conf Track Proc, pp. 1–14, 2015. doi:10.48550/arXiv.1409.1556
[28] Watanabe JI, Ren S, Zhao Y, et al. Power line-tree conflict detection and 3D mapping using aerial images taken from UAV, autonomous systems: sensors, vehicles, security, and the internet of everything, pp. 10643:106430U, 2018. doi:10.1117/12.2303480
[29] A. Cerón IF, Mondragón B, Prieto F. Power line detection using a circle based search with UAV images, International Conference on Unmanned Aircraft Systems, pp. 632-639, 2014. doi:10.1109/ICUAS.2014.6842307
[30] Nguyen VN, Jenssen R, Roverso D. Automatic autono mous vision-based power line inspection: A review of current status and the potential role of deep learning. Int J Electr Power Energy Syst. 2018;99:107–120. doi:10.1016/j.ijepes.2017.12.016
[31] Baker L, Mills S, Langlotz T, et al. Power line detection using Hough transform and line tracing techniques, International Conference on Image and Vision Computing, pp. 1–6, 2016. doi:10.1109/IVCNZ.2016.7804438
[32] Zou K, Zhao S, Jiang Z. Power line scene recognition based on convolutional capsule network with image enhancement. Electronics (Basel). 2022;11(18):2834), doi:10.3390/electronics11182834
[33] Yetgin ÖE, Gerek ÖN. Automatic recognition of scenes with power line wires in real life aerial images using DCT-based features. Digit Signal Process. 2018;77:102–119. doi:10.1016/j.dsp.2017.10.012
[34] Gatos B, Pratikakis I, Perantonis SJ. Improved document image binarization by using a combination of multiple binarization techniques and adapted edge information. 19th International Conference on Pattern Recognition: 1–4. 2008. doi:10.1109/ICPR.2008.4761534
[35] Gonzalez RC. Digital image processing. Pearson Education India; 2009.
[36] Smith P, Reid DB, Environment C, Palo L, Alto P, Smith PL. NOBUYUKI OTSU, A threshold selection method from gray-level histograms, IEEE Trans Syst Man Cybern, vol. 20, pp. 62–66, 1979.
[37] Villán AF. Mastering OpenCV 4 with python: a practical guide covering topics from image processing, augmented reality to deep learning with OpenCV 4 and python 3.7. Packt Publishing Ltd, Birmingham; 2019.
[38] Lim JS. Two-dimensional signal and image processing. Englewood Cliffs, New Jersey; 1990.
[39] Canny J. A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell PAMI. 1986;8:679–698. doi:10.1109/TPAMI.1986.4767851
[40] Kubrikov MV, Paulin IA, Saramud MV, et al. Application of sequential processing of computer vision methods for solving the problem of detecting the edges of a honeycomb block. J Phys Conf Ser. 2020: 1679), doi:10.1088/1742-6596/1679/4/042098
[41] Redington RW, Berninger WH. Medical imaging systems, pp. 34, 1981. doi:10.1063/1.2914691
[42] Ma Y, Ma H, Chu P. Demonstration of quantum image edge extraction enhancement through improved sobel operator. IEEE Access. 2020. doi:10.1109/ACCESS.2020.3038891
[43] Guo Y, Pang Z, Du J, et al. An improved AlexNet for power edge transmission line anomaly detection. IEEE Access. 2020;8:97830–97838. doi:10.1109/ACCESS.2020.2995910
[44] Qu Z, Mei J, Liu L, et al. Crack detection of concrete pavement with cross-entropy loss function and improved VGG16 network model. IEEE Access. 2020;8:54564–54573. doi:10.1109/ACCESS.2020.2981561
[45] Han J, Yang Z, Zhang Q, et al. A method of insulator faults detection in aerial images for high-voltage transmission lines inspection. Appl Sci. 2019;9(10):2009), doi:10.3390/app9102009
[46] Garbin C, Zhu X, Marques O. Dropout vs. batch normalization: an empirical study of their impact to deep learning. Multimed Tools Appl. 2020;79:12777–12815. doi:10.1007/s11042-019-08453-9
[47] Gao Z, Yang G, Li E, et al. Insulator segmentation for power line inspection based on modified conditional generative adversarial network. J Sensors. 2019: 1–8. doi:10.1155/2019/4245329
[48] Tschopp F, Von Einem C, Cramariuc A, Hug D, Palmer AW, Siegwart R, et al., Hough^{2}Map-Iterative event-based hough transform for high-speed railway mapping, IEEE Robot Autom Lett, 6, pp. 2745–2752, 2021. doi:10.1109/LRA.2021.3061404
[49] Shahi TB, Dahal S, Sitaula C, et al. Deep learning-based weed detection using UAV images, a comparative study. Drones. 2023;7(10):624), doi:10.3390/drones7100624
[50] [Dataset] Yetgin, ÖE., & Gerek, ÖN. PL Image Dataset (Infrared-IR and Visible Light-VL), *Mendeley Data*, pp. 7, 2017.