*UTILITY POLES IDENTIFICATION USING GOOGLE STREET VIEW IMAGES*

1S. Vasavi, 2 \* Chathurya Sunkara, 3Aarya Sri Gullapalli**,** 4 Jayadeep Sai Bolla, 5 Sudeer Tiwari

Department of Artificial Intelligence and Data Science

1,2,3,4Velagapudi Ramakrishna Siddhartha Engineering College, Andhra Pradesh, India

5 Andhra Pradesh Space Applications Centre (APSAC)

[1vasavi\_movva@vrsiddhartha.ac.in](mailto:1vasavi_movva@vrsiddhartha.ac.in), [2sunkarachathurya@gmail.com , 3gullapalliaarya@gmail.com,](mailto:2sunkarachathurya@gmail.com%20,%203gullapalliaarya@gmail.com,)

4[jayadeep0304@gmail.com](mailto:jayadeep0304@gmail.com), [5sudeer.apsac@ap.gov.in](mailto:5sudeer.apsac@ap.gov.in)

*Abstract*— Accurate detection of utility poles is essential for effective infrastructure management in power systems, but traditional methods often fall short in delivering the needed precision. Previous versions of You Only Look Once (YOLO) showed promise but lacked the accuracy required for reliable utility pole identification. To address these shortcomings, an enhanced YOLOv9 model was developed to improve detection performance in street view images. By incorporating advanced feature extraction techniques, the YOLOv9 model achieves better detection accuracy. Tested on Google Street View images, the model reached a 95% detection accuracy while reducing the False Positive Rate to 9.01%. These results demonstrate the model’s ability to accurately identify utility poles with fewer false alarms, offering a reliable tool for infrastructure supervision. The enhanced YOLOv9 marks a significant improvement, making maintenance and management more efficient.

***Keywords— Utility Poles, YOLOv9(You Only Look Once version 9), Satellite images, Data augmentation, Image processing***

**I. INTRODUCTION**

Utility poles are integral to urban infrastructure, supporting electrical lines, communication cables, and various other services. Accurately identifying utility poles is essential for maintenance, urban planning, and managing infrastructure. Traditional methods often depend on manual inspections, which are not only time-consuming but also labor-intensive. With advancements in machine learning, automated detection using computer vision has emerged as a promising alternative. Previous models, including earlier versions of YOLO and other convolutional neural networks (CNNs), have successfully detected utility poles [2-7]. However, these models have faced several challenges, including reduced detection accuracy in complex urban environments, where cluttered backgrounds make it difficult to differentiate poles from other objects; handling occlusions, where trees, vehicles, or buildings partially block poles; and inconsistent performance under varying lighting conditions, such as shadows or night-time imagery, which affected detection reliability. Furthermore, the real-time processing speed of earlier models was often insufficient, limiting their practical use. This paper explores the application of YOLOv9 (You Only Look Once, Version 9) [9] for detecting utility poles in Google Street View images, offering improvements in feature extraction, detection accuracy, and real-time processing.

**A. MOTIVATION**

As the demand for electricity increases, it is uncertain whether the existing poles are sufficient for the population, and as the number of poles increases daily, Inspection and identification through manual techniques have hindrances like labor-intensive, time-consuming, and the possibility of mistakes. Hence there is a need for an automated approach for the detection of electric poles which can be effective and able to tackle the disadvantages in the manual process that is present being implemented.

**B. PROBLEM STATEMENT**

The research aims to leverage the YOLOv9 model to

improve the detection accuracy and efficiency of

utility pole identification. Research includes

training the modified YOLOv9 model on a comprehensive dataset of utility poles, evaluating its

performance, and assessing its effectiveness across

diverse urban settings. The findings illustrate the

the potential of advanced deep learning techniques for

efficient and scalable infrastructure management,

particularly in the automated detection of utility poles

**C. OBJECTIVES**

* To compile a comprehensive dataset of utility poles.
* To evaluate existing identification techniques in terms of accuracy and performance.
* To develop an enhanced YOLO model for utility pole detection that gives high performance.

**D. CONTRIBUTIONS**

* Created a dataset for utility pole identification belonging to the Boston city
* Proposed enhanced YOLOv9 model that offers improved feature extraction and more robust performance in detecting utility poles across varied environments compared to the original model. It also enhances adaptability and reliability in practical applications

**E. ORGANIZATION OF PAPER**

* **Section 1** introduces the problem, and the motivation behind utility pole detection, and outlines the contributions of our work.
* **Section 2** reviews related works in utility pole detection and object detection using deep learning models.
* **Section 3** describes the proposed method, including the use of the YOLOv9 model and advanced feature extraction techniques.
* **Section 4** presents the experimental results, discusses the performance of the proposed model, and provides a comparison with existing approaches.
* **Section 5** concludes the paper and suggests future directions for improvement.

**II. RELATED WORKS**

Recent advancements in utility pole detection have been explored from various perspectives. Sharma (2024) presents an image analysis-based method for automatic utility pole detection in remote surveillance [1]. Gomes et al. (2020) enhance detection accuracy by mapping utility poles in aerial orthoimages using the ATSS deep learning method [5]. Marty et al. (2024) use a CNN approach to classify high-voltage transmission poles from satellite images, focusing on rural areas in Thailand [4]. Li et al. (2024) improve YOLOv9 for object detection under severe weather conditions from a drone perspective [12]. Wang et al. (2024) discuss YOLOv9’s programmable gradient information to optimize learning strategies [9]. Additionally, a 2021 article addresses automating utility pole recognition and inspection with computer vision techniques [8], while Wang et al. (2022) explore network design strategies through gradient path analysis [10]. Yin et al. (2023) introduce the Pyramid Enhancement Network (PE-YOLO) for dark object detection, which can be beneficial for detecting utility poles in challenging visual conditions [11]. These contributions reflect the ongoing efforts to refine utility pole detection using various advanced technologies and methodologies. Lou Peeples emphasizes that utility poles are one of the most valuable and vulnerable assets for utility companies and highlights necessary tagging and identification practices [13].

**A. RESEARCH GAP**

Current research on the identification of utility poles from street view imagery has primarily focused on the application of deep learning models such as YOLOv3 and CNN in urban settings [3,7]. There remains a notable gap in comprehensive datasets that encompass diverse environmental conditions beyond urban landscapes, including suburban and rural areas. These environments pose distinct challenges such as varying lighting conditions, varied pole designs, and occlusions by vegetation and infrastructure. Additionally, existing studies often rely on single-source data, limiting the integration of multi-modal data sources like LiDAR or high-resolution satellite imagery, which could enhance pole detection accuracy [2,6]. Addressing these gaps is essential to develop robust and adaptable models capable of accurately identifying utility poles across varied geographical and environmental contexts.

**B. STUDY AREA AND DATA PREPARATION**

Fig.1 contains a Sample input image. Data preparation stands as the foundational step in any deep learning endeavor, involving the collection of raw data essential for training the model. The dataset that is available in [14] is used in the proposed system. The dataset comprises 2,094 images, with 80% allocated for training, 10% for testing, and 10% for validation.

A bus driving down a road

Description automatically generated

Fig.1 Sample input image

**III. PROPOSED METHOD**

A diagram of a software flow

Description automatically generated with medium confidence

Fig 2 enhanced model architecture

The architecture of modified yolov9 comprises three primary components: The Backbone, Neck, and Head components. Fig 2 presents the enhanced model architecture.

**A. ANNOTATION**

Bounding box techniques are essential in object detection, as they enable precise localization of objects within images. Two common representations are the corner points format Eq (4) and the center-size format Eq (2). The corner points format defines a bounding box using the coordinates of the top-left (xmin, ymin) and bottom-right (xmax, ymax) corners, directly correlating with the image pixel grid. However, it is more sensitive to variations in object scale and aspect ratio, and prediction errors can propagate across multiple coordinates, leading to inaccuracies, particularly for small or occluded objects. In contrast, the center-size format uses the center (Xcenter, Ycenter) and dimensions (width, height) of the bounding box, offering more stability by reducing variables and isolating the object's central position. This often results in more accurate predictions, especially in complex environments. Both formats impact the Intersection over Union (IoU) Eq (3), which measures the overlap between the predicted and ground truth boxes, making IoU a critical metric for evaluating detection performance and overall accuracy.

(1)

(2)

(3)

**B. PRE-PROCESSING**

Image preprocessing is a critical component in optimizing object detection models. As part of the preprocessing pipeline, images undergo auto-orientation to correct any orientation inconsistencies that may arise from different capture devices or formats, ensuring uniform orientation for consistent processing. Following auto-orientation, each image is resized to a standard dimension of 640x640 pixels, defined in Eq (4). The resizing operation as given in Eq (5) ensures that the model receives consistent inputs, which is important for improving the accuracy and reliability of object localization and classification tasks. By minimizing variability in image dimensions, this preprocessing step contributes significantly to optimizing model training and inference performance.

(4)

(5)

**C. AUGMENTATION**

Data augmentation is crucial for enhancing the robustness and generalization capability of deep learning models, especially in tasks like object detection. By applying various transformations to the original training data, such as rotations, translations, flips, and adjustments in brightness and contrast, augmented datasets introduce variability that aids the model in learning invariant features. Additionally, augmentations include introducing noise, such as Gaussian noise with a standard deviation of 0.1%, to simulate real-world imperfections Eq (6), where ϵ∼N(0, σ2). These techniques, like horizontal flips Eq (7) and random rotations Eq (8), expand the dataset by providing diverse perspectives of objects and scenarios. This variability helps the model generalize better to unseen data and reduces overfitting by exposing it to a broader range of conditions than those in the original dataset.

(6)

(7)

(8)

**D. MODEL CREATION**

**Input:** The input layer serves as the initial stage of the architecture and is tasked with receiving Google Street View images that may contain utility poles. The input layer handles images with dimensions of 640x640 pixels. Any image not meeting these dimensions requires preprocessing.

**Backbone:** The backbone of the YOLO model serves as the main feature extractor, processing input images to extract key features for further analysis. Initial convolutional layers reduce the input size and increase feature depth, capturing detailed information. Custom RepNCSPELAN4 blocks refine these features through convolutions, batch normalization, and activations. Adown layers downsample feature maps at different stages for effective multi-scale feature extraction. The model's head uses these features to generate bounding boxes and class probabilities. SPPELAN blocks incorporate spatial pyramid pooling for multi-scale feature capture, while upsample layers improve feature map resolution. Concatenation layers merge feature maps from different stages, enriching detection information. Lastly, the DDetect layer predicts bounding boxes and class probabilities with the refined features, ensuring accurate object detection. The activation function that is used is Leaky ReLU (Rectified Linear Unit) and is defined as

(9)

Table 1 compares the proposed model with the existing model w.r.t parameters. The proposed model offers several advantages over YOLOv9. It incorporates a higher number of initial filters (80 vs. 32) and more layers (390 vs. 105), potentially enhancing its ability to extract detailed features and learn complex representations. Despite the increased depth, it maintains fewer parameters (55M vs. 60M), suggesting improved parameter efficiency and reduced risk of overfitting. Consistent input and output dimensions (640x640x3) streamline the pre-processing and training processes. These factors collectively indicate potential improvements in feature extraction, model complexity handling, and computational efficiency compared to the YOLOv9 baseline.

|  |  |  |
| --- | --- | --- |
| **Parameters** | **YOLOv9 model [9]** | **Proposed**  **model** |
| Number of channels in the input image | 3 | 3 |
| Dimensions | 600x600x3 | 640x640x3 |
| Strides | 2x2 | 2x2 |
| Size of Input Kernel | 3x3 | 3x3 |
| Number of Initial Filters | 32 | 80 |
| Variables | 60M | 55M |
| No. of layers | 105 | 390 |
| Channels in Resulting Image | 3 | 3 |

Table 1: Comparison of the model

**E. MODEL TRAINING**

Load the prepared and augmented dataset into the training pipeline, making sure to use a batch size of 16 for proper batching. Initiate the training process of the enhanced YOLOv9 model with the configured hyperparameters [15]. Monitor the training progress, loss, and accuracy metrics. Use the YOLOv9 loss function, which combines classification loss, localization loss, and objectness loss, to guide model optimization. The loss function typically includes:

(10)

where,

classification loss (LOSSclass) measures the error between the predicted class (utility pole or other objects) and the actual class. For utility pole detection, optimizing this loss ensures that the model accurately distinguishes poles from similar urban objects like streetlights, traffic signs, or trees, which are often visually similar in the dataset.

Bounding box regression loss (LOSSbbox), which evaluates how accurately the predicted bounding box fits the ground truth bounding box for utility poles

objectness loss (LOSSob), measures the confidence of the model that an object (in this case, a utility pole) exists within the predicted bounding box. A well-optimized objectness loss reduces false positives by ensuring the model only focuses on actual utility poles and ignores irrelevant background features or other non-target objects.

**F. MODEL TESTING**

Load the trained YOLOv9 model weights that were obtained after the training phase. Utilize the YOLOv9 model to generate bounding boxes and class probabilities for utility poles in test images. Evaluate performance metrics such as precision, recall, and mean Average Precision (mAP) to determine the model's effectiveness.

**G. EVALUATION METRICS**

The model's performance can be assessed using metrics like Precision (P), Recall (R), and the F1 score (Dice coefficient).

True Positive (TP) represents the proportion of correctly identified pixels as Utility Poles, False Positive (FP) denotes the proportion of incorrectly identified pixels as Utility Poles, True Negative (TN) indicates correctly identified pixels not classified as Utility Poles, and False Negative (FN) signifies incorrectly identified pixels as Utility Poles. Posi

(11)

(12)

(13)

(14)

(15)

(16)

Where,

TP (true positives) Truly detected utility pole objects.

FP (false positive) Falsely detected utility pole objects.

FN (false negative) non-detected utility pole objects.

mAP = 1/C (17)

Where,

AP Average precision

mAP Mean average precision

**IV. RESULTS AND DISCUSSION**

The model is successfully able to predict the utility poles in testing. Fig.3 shows an example output in which utility poles were detected.

The key achievement of this model is that it can achieve high accuracy and shows strong performance. Table 2 presents the comparison of the proposed model with the existing works. The proposed model exhibits superior performance with 95% accuracy and 90.04% precision, significantly surpassing previous methods in reliability and precision. The F-1 score of 94.7% reflects a well-balanced trade-off between precision and recall. The false positive rate is 9.01% and can primarily be attributed to complex urban environments and occlusions. Complex environments with objects like streetlights and trees can lead to misclassification. This can be mitigated by enhancing data augmentation and incorporating contextual information to better differentiate utility poles from similar objects. Occlusions and partial views of poles can also result in false positives; this issue can be addressed by including occluded examples in training or using synthetic occlusions. Additionally, improving the objectness score threshold can help filter out uncertain predictions, further reducing false positives. Addressing these factors through improved training strategies and post-processing adjustments can significantly enhance detection accuracy. Additionally, the use of a diverse dataset of street view images from various regions enhances the model's robustness and adaptability across different environments.

Fig.4 shows the accuracy graphs. The model is learning and improving its object detection ability as it processes more data (more epochs). The model's performance has plateaued, potentially indicating it has reached its peak performance or is stuck in a local minimum. The decrease in loss can be observed as the number of epochs increases, as shown in the graph above. Fig.5 shows loss graphs at the time of training and testing. The graphs show separate lines for the training loss and validation loss. The training loss curves tend to decrease steadily as the model is exposed to more training data. This suggests the model is learning to fit.

The experiments were conducted using Google Colab, leveraging an NVIDIA T4 GPU with 16 GB VRAM for model training and inference. The implementation was done in Python 3. x, utilizing the PyTorch framework. Google Colab's environment included pre-installed CUDA 11. x and cuDNN 8. x for GPU acceleration. The YOLOv9 model was implemented using a modified version of the Ultralytics YOLOv5 repository, adapted for YOLOv9 architecture.

**A. DISCUSSION**:

The results of this study on Utility pole identification Using Google Street View Images demonstrate that the YOLOv9 model performs effectively, achieving high accuracy in detecting utility poles within diverse and complex urban environments. YOLOv9’s advanced feature extraction and architecture enable robust detection, even with challenges such as varying lighting, pole appearances, and cluttered backgrounds. However, occasional false positives were observed, particularly with objects resembling poles, like street lamps or trees. Despite these challenges, the model shows great potential for real-world applications in infrastructure monitoring and urban planning. Future work could explore integrating additional data sources or ensemble methods to further enhance detection accuracy and reduce false positives.

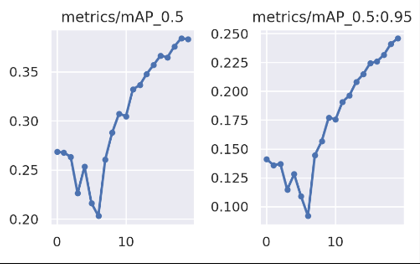
A collage of a street view

Description automatically generated

Fig.3 Sample output image detected by the model

|  |  |  |
| --- | --- | --- |
| Work | Performance | Dataset |
| [1] | Accuracy: 94%, Precision: 92% | UAV Images, China |
| [3] | Accuracy: 88%, Precision: 85% | street-level view in urban and suburban areas of China |
| [5] | Accuracy: 88%, Precision: 85% | Aerial orthoimages, rural and urban regions |
| [6] | Accuracy: 85%, Precision: 82% | Mobile LiDAR data, urban areas with dense infrastructure |
| [7] | Accuracy: 87%, Precision: 84% | Aerial stereo imagery, power line corridors |
| Proposed model | Accuracy: 95%  Precision: 90.04  Recall: 100  F-1:94.7  FPR:9.01 | Street view images |

Table 2: Comparison of model performance

Fig.4-AccuracygraphsA graph of loss and results

Description automatically generated with medium confidence Fig.5 Loss graph

**V. CONCLUSIONS**

An upgraded YOLOv9-based utility pole detection system using Google Street View imagery has been explored in this piece of work. The model has been trained on extensive datasets, tested, and validated with a sufficient number of epochs, adhering to the YOLOv9 architecture. The robust feature extraction capabilities of the model have enabled accurate detection of utility poles. However, limitations remain in the detection process, such as the challenge of detecting poles under varying environmental conditions, occlusions, and complex backgrounds, which can still impact performance. Additionally, the model could benefit from enhancements to better handle issues like overlapping objects or extreme weather conditions. Future work aims to improve the detection accuracy by incorporating additional feature extraction techniques, such as pole shape and size, pole texture, contextual information, and shadow analysis. These enhancements could lead to a more comprehensive and reliable detection system. The overall methodology ensures the continued development of advanced solutions for automating utility pole detection by integrating systematic data collection, preprocessing, model training, and evaluation strategies.

**REFERENCES**

[1] Hrishikesh Sharma. (2024). *Image Analysis-Based Automatic Utility Pole Detection for Remote Surveillance | Semantic Scholar*. semanticscholar.org. https://www.semanticscholar.org/paper/Image-Analysis-Based-Automatic-Utility-Pole-for-Sharma-Vellaiappan/aba4ffb5e2526229c1adc3903821ba23f14cd736

**[2] Shokri, D., Rastiveis, H., Shams, A., and Sarasua, W. A.: Utility Poles Extraction from Mobile Lidar Data in Urban Area Based on Density Information, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLII-4/W18, 1001–1007, https://doi.org/10.5194/isprs-archives-XLII-4-W18-1001-2019, 2019**

[3] Zhang, Y., & Alshaykh, O, 5G Utility Pole Planner Using Google Street View and Mask R-CNN. 2020 IEEE International Conference on Electro Information Technology (EIT), 309-312, https://arxiv.org/pdf/2008.11689

[4] Marty, B., Gaudin, R., Piperno, T., Rouquette, D., Schwob, C., & Mezeix, L. (2024). Methodology to classify high voltage transmission poles using CNN approach from satellite images for safety public regulation application: A study case of rural area in Thailand. *Systems and Soft Computing*. https://doi.org/10.1016/j.sasc.2024.200080

**[5] Gomes M, Silva J, Gonçalves D, Zamboni P, Perez J, Batista E, Ramos A, Osco L, Matsubara E, Li J, et al. Mapping Utility Poles in Aerial Orthoimages Using ATSS Deep Learning Method. Sensors. 2020; 20(21):6070. https://doi.org/10.3390/s20216070**

[6] Talebi Nahr, S., & Saadatseresht, M. (2021). Utility‐pole Detection Based on Interwoven Column Generation from Terrestrial Mobile Laser Scanner Data. *The Photogrammetric Record*. https://doi.org/10.1111/phor.12394.

**[7] Abdul QayyumImran RazzakAamir MalikAamir MalikSajid AnwarSajid Anwar,** **Fusion of CNN and sparse representation for threat estimation near power lines and poles infrastructure using aerial stereo imagery,** T**echnological Forecasting and Social Change, 168, 2021, DOI: 10.1016/j.techfore.2021.120762**

**[8] Automating Utility Pole Recognition & Inspection with Computer Vision, 2021,** <https://utilityanalytics.com/2021/05/automating-utility-pole-recognition-inspection-with-computer-vision/> **Last accessed on 12-06-2024**

[9] Chien-Yao Wang, I-Hau Yeh, Hong-Yuan Mark Liao, YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information, arXiv:2402.13616,2024

[10] Wang, Chien-Yao, Hong-Yuan Mark Liao, and I-Hau Yeh. “Designing network design strategies through gradient path analysis.”, arXiv:2211.04800 2022

[11] Yin, X., Yu, Z., Fei, Z., Lv, W., Gao, X. (2023). PE-YOLO: Pyramid Enhancement Network for Dark Object Detection. In: Artificial Neural Networks and Machine Learning – ICANN 2023. ICANN 2023. Lecture Notes in Computer Science, vol 14260. Springer, Cham. https://doi.org/10.1007/978-3-031-44195-0\_14

[12] Li J, Feng Y, Shao Y, Liu F. IDP-YOLOV9: Improvement of Object Detection Model in Severe Weather Scenarios from Drone Perspective. Applied Sciences. 2024; 14(12):5277. https://doi.org/10.3390/app14125277

[13] Lou Peeples. (2024). *Utility Pole Identification and Numbering Systems: 10 Expert Resources - Camcode*. Camcode. https://www.camcode.com/blog/utility-pole-identification-and-numbering-systems-resources/

[14] Chathurya, Utility poles identification Dataset. Roboflow Universe. Roboflow. **Last accessed on 12-06-2024**

[15] James Gallagher, Piotr Skalski, How to Train YOLOv9 on a Custom Dataset. Roboflow Blog: <https://blog.roboflow.com/train-yolov9-model/>, **Last accessed on 12-06-2024**