Object Classification with YOLOv5 for Electric Utility Asset Inspection using UAVs

Hen-Geul Yeh, Richard Nguyen, and Seok-Chul Kwon
Electrical Engineering Department, California State University Long Beach, Long Beach, CA 90840, USA

***Abstract* — The inspection of electric utility assets is an important procedure for repair and hazard prevention such as wildfires. Traditionally, human workers inspect power lines manually which is time consuming and potentially dangerous due to high elevation and high voltage. Manual inspection requires power to be shut off during the procedure and results in inconvenient blackouts for residents and businesses which can be exacerbated by time spent diagnosing faults and repair equipment. With recent developments in computer vision and artificial intelligence (AI) using machine learning, the process of inspecting electric utility assets can be both expedited and made safer using unmanned aerial vehicles (UAVs) in conjunction with sustainable and resilient federated learning communication networks. This work aims to train object classification models for deployment on UAVs which will detect common electric utility assets during inspection processes. The models are trained on a large dataset of approximately 30,000 high resolution images capturing five class objects: crossarms, cutouts, insulators, poles, and transformers. Eight different model configurations of the YOLOv5 algorithm are trained using the dataset and scored against each other to assess performance and computational cost for deployment on UAVs.**

***Index Terms*— computer vision, machine learning, power line inspection, object detection, YOLOv5, artificial intelligence, UAV**

I. Introduction

Modern society is heavily dependent on electrical power being delivered to homes and workplaces across the nation. Decades of transmission and distribution infrastructure have been built up for urban and suburban everyday living. When electric utilities need to be maintained, repaired, or turned off due to unexpected weather conditions, power delivery must be temporarily shut off. Power shutoffs can prove to be a major inconvenience leaving people unable to do their work and preventing businesses from operating. The length of a power shutoff can vary depending on whether it occurs for inspection and maintenance of something more severe such as wildfire prevention in dangerous weather conditions. In the case of inspections, electric utility assets must be manually inspected by human workers which can be a slow and tedious process.

The research presented in this work is part of a larger unmanned aerial vehicles (UAVs) project. Each UAV can hover at different locations and obtain images. Inspection can be both expedited and made safer through UAVs for the electric utility industry. Therefore, it is an efficient methodology that each UAV fulfills different levels of learning in a distributed and collaborative fashion, which is a new paradigm raised by federated learning (FL). The main concept of this larger project is to deploy a FL communication network of drones to inspect a power distribution or transmission circuit. Automatic flight path will allow the drones to fly along the power lines, object classification algorithms will allow them to detect and diagnose problems along the circuit. A sustainable and resilient FL network will allow the drones to relay information to each other and the electric utility base station of industrial operations for a completely streamlined and automatic inspection procedure efficiently. In this paper, we focus on developing a machine learning (ML) model capable of detecting and classifying electric utility assets, specifically, before and after wildfires or other disasters. This paper only considers computing of ML model using YOLOv5 algorithm as the baseline, meanwhile the relationship of this work and communications by FL has been presented at a different paper [1].

While similar computer vision on object detection research has been performed in literature, previous works have been performed with much smaller datasets. By using a much larger and higher quality dataset with *multiple objects*, this project aims to develop a *robust baseline object classification* model which can be further developed in future stages of the larger project to detect faulty and damaged equipment. Our contribution is provided as follows. (1) We train object classification models for deployment on UAVs which detected common electric utility assets during inspection processes with the specific steps and resized parameters involved in implementing YOLOv5 for object classification. (2) The models were trained on a significant large dataset of approximately 30,000 high resolution images capturing five class objects: crossarms, cutouts, insulators, poles, and transformers. (3) Eight different model configurations of the YOLOv5 algorithm were trained using the dataset and scored against each other to assess performance and computational cost for deployment on UAVs.

II. Related Works

The following works employed UAVs for aerial inspection of electric utility assets. Takaya et al. present a proof of concept using a quadrotor helicopter drone equipped with sensors and a camera to follow transmission lines and record clear footage of electric utilities and surrounding vegetation [2]. The success of this research demonstrated the benefits of using a UAV to inspect areas which make manual inspection difficult such as rocky terrain and dense vegetation.

Chen et al. implemented the YOLOv3 algorithm for detecting and counting poles in UAV inspection video footage [3]. The model was trained to differentiate between upright and fallen poles. Due to the scarcity of data, the authors simulated images to represent the desired data and generated a set of 13,000 images. The model was able to achieve about 90% accuracy with this dataset, demonstrating the robustness of the YOLO algorithm.

Sumagayan et al. used YOLOv5 to detect six powerline components from images captured at their university in the Philippines [4]. Again, due to a scarcity of data, the authors opted to obtain images from powerlines on the university campus, manually label the images, and augment the images to produce a dataset of 10,000 images. Their results demonstrated very high scores which will be compared to the YOLOv5 models produced in this paper.

Works attempting to use object detection defect and fault detection on electric utilities have only focused on insulators specifically. Feng at al. used a dataset of only 3000 images to train an insulator detector that can pinpoint defects using images of large insulators in China [5]. Huang et al. also developed a fault detector using YOLOv5 for insulators in distribution networks specifically [6]. This work also used a dataset of 3000 images. Souza et al. created an insulator defect classifier for transmission lines using a model that combines the YOLO algorithm with a ResNet classifier using a dataset of only around 1000 images [7]. Throughout all of the works on object detection for electric utility assets, it becomes very apparent that data scarcity is a major problem. Researchers in this area are forced to procure their own small local datasets and find methods to augment the images to create a dataset suitable for training a ML model. However, the amount of success with the YOLO algorithm with such limited datasets demonstrates the algorithm's robustness and ability to detect electric assets accurately.

III. Method

This work implements object classification using the YOLOv5 algorithm. The original YOLO (You Only Look Once) algorithm was developed by Joseph Redmon and released in 2016 [8]. Prior object detection methods followed a two-stage approach where the first stage extracts information and proposes bounding boxes on an image. The second stage runs a classifier on these proposed bounding boxes and classifies the object within each box. The two-stage method was slow and difficult to optimize. YOLO was a major breakthrough by streamlining the process into a single-stage approach modeled as a regression problem. The YOLO algorithm predicts multiple bounding boxes and class probabilities simultaneously.

In 2020, Glenn Jocher and Ultralytics released the YOLOv5 algorithm on GitHub [9]. YOLOv5 was implemented on the PyTorch framework which was a departure from the previous iterations of YOLO running on the Darknet framework. YOLOv5 also comes in five different configurations: nano, small, medium, large, and extra large. These models are named YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. These configurations refer to the number of layers and depth of the neural network. These standard models are trained on input images of 640 × 640 resolution. YOLOv5 also has variants of all five configurations which use input images of 1280 × 1280 resolution instead which are instead named YOLOv5n6, YOLOv5s6, YOLOv5m6, YOLOv5l6, and YOLOv5x6.

Due to the large size of the dataset, we trained YOLOv5 on local hardware rather than in an online notebook such as Google Colab. The machine used for training was equipped with a AMD Ryzen 5 2600X processor, NVIDIA GeForce 3060 Ti graphics card, and 48 GB of RAM. All configurations of the YOLOv5 algorithm were trained except for YOLOv5l6 and YOLOv5x6 which were too large for the machine to train. All models were trained for 100 epochs using the Auto batch function for batch size to maximize resource utility on our machine. Additionally, we trained the models starting with pretrained weights which were provided by Ultralytics after training each configuration with the MS COCO dataset [10]. Our initial experiments showed that pretrained weights yielded better results rather than training from scratch weights for our dataset.

IV. Dataset

The YOLOv5 object classification models are trained using the Drone-based Distribution Inspection Imagery dataset from the Electric Power Research Institute (EPRI). EPRI produced the dataset and made it publicly available due to the scarcity of available electric utility infrastructure imagery datasets [10]. The dataset consists of 29,705 images in .jpg format with resolution of 5184 × 3888 pixels. All images are overhead imagery of distribution lines captured from a UAV. Due to the nature of aerial imagery capturing sensitive information, EPRI has removed images containing people, cars, and other sensitive background information from the dataset. To further anonymize the images, EPRI has also .exif image-data scrubbed all images in the dataset. On July 2022, EPRI published the labels for the dataset indicating six classes of objects to be detected: poles, crossarms, insulators, cutouts, transformers, and background structures [12].

From EPRI [11], the dataset captures aerial imagery of 16 different distribution circuits is given in Table I. The images vary in terms of lighting condition, angle, and background objects and conditions, which all serve to enrich the dataset to produce a more robust object classification model. Many different daytime lighting conditions are captured, i.e. direct sunlight, sunset, overcast, etc., but fog, rain, and nighttime images are not included. This means the models produced are intended to be deployed on UAVs in mostly clear daytime conditions. To split the data, we aim to use 80% of the data for training, 10% for validation, and 10% for testing. After determining how many images belong to each of the different distribution circuits, images from Circuit 15 are reserved for the validation set and images from Circuit 2 are reserved for the testing set since the number of images from each circuit approximately achieve the split goals. By using complete circuits for validation and testing rather than randomly splitting data, the scores used to evaluate the models are more indicative of the real-world application of a UAV inspecting an unseen distribution circuit.

TABLE I
NUMBER OF IMAGES FOR EACH DISTRIBUTION CIRCUIT

Distribution Circuit	Number of Images
1	1909
2	2977
3	954
4	979
5	428
6	642
7	717
8	675
9	1996
10	575
11	4954
12	3008
13	2252
14	1895
15	2673
16	3071
Total	29705

TABLE II
NUMBER OF INSTANCES FOR EACH OBJECT CLASS

	Training Set	Validation Set	Testing Set	Total
Crossarm	12597	9714	757	23068
Cutout	7915	640	929	9484
Insulator	45545	724	3372	49641
Pole	24246	4576	3001	31823
Transformer	7349	808	577	8734
Background Structure	347	33	35	415

Table II shows the distribution of object instances for the dataset. The background structure class shows a significantly lower number of instances compared to the other classes. Ultralytics recommends approximately 10,000 instances for each class for sufficient training results [13]. Examination of these background structures showed that they captured supporting structures that could be confused for poles, but training results showed that accuracy detecting the class was very low and resulted in more errors than helping in detecting poles more accurately. Additionally, background structures are not an electric utility asset and would better be considered part of the background instead. Preliminary experiments excluding the background structure class from training yielded better results in the performance of our models.

The dataset underwent minimal pre-processing due to data augmentation options being present in the hyperparameter settings for YOLOv5. We used default hyperparameters which include hue, saturation, and value augmentation for color space, rotation, translation, flipping across horizontal or vertical axis, and mosaic augmentation. To save computational resources, we resized the images into datasets appropriate for the 640- and 1280-resolution models of YOLOv5. Using default training parameters, the YOLOv5 algorithm will automatically resize images to the appropriate input size. Maintaining the same aspect ratio of the original images means the 640 dataset has images of resolution 640 × 480 while the 1280 resolution dataset has images of resolution 1280 × 960. All images were saved in a .jpg format. Preliminary results comparing the training time using the original, 1280, and 640 for training the YOLOv5s model showed that training time for each epoch is drastically reduced while learning is not affected. On our hardware, the original resolution dataset took 50 minutes for one epoch, the 1280-resolution dataset took 6 minutes, and the 640-resolution dataset took 3.5 minutes. Resizing the images to the appropriate input size before training saves a large amount of computation time compared to allowing the YOLOv5 algorithm to resize images during training. Fig. 1 shows the bounding boxes of Ground Truth from Dataset while Fig. 2 shows the predictions from YOLOv5.

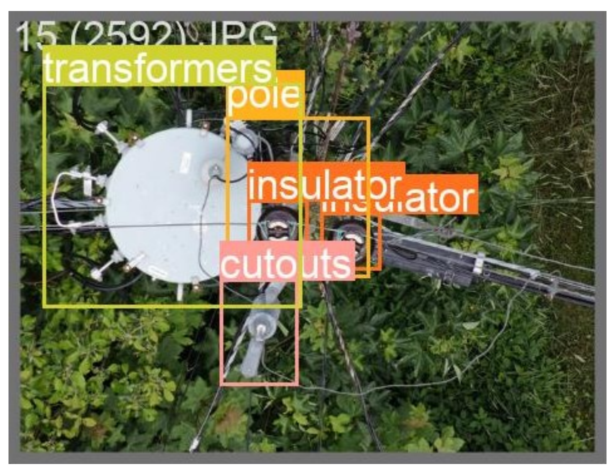

Fig. 1. Example of Ground Truth from Dataset.

V. EVALUATION METRICS

The standard metrics for evaluating the performance of object classification models are precision, recall, and mean average precision (mAP). The formulas for calculating precision and recall are as follows:

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

where TP stands for true positive, FP stands for false positive, and FN stands for false negative. For object classification, precision indicates the accuracy of objects detected by the model. Recall represents how well the model can detect our objects of interest. Note that true negative (TN) is assumed to never be applicable for object classification since it is assumed that there are always objects to detect. Any objects we are not interested in are simply detected as part of the background.

Fig. 2. Example of Predictions from YOLOv5.

While applications such as image classification have straightforward methods to differentiating between TP, FP, and FN, object classification requires another metric to determine these values due to the nature of classifying sections of an image rather than the image as a whole. To this end, the concept of Intersection over Union (IoU) is used as a threshold to determine which guess made by the algorithm is TP, FP, or FN. For the YOLOv5 algorithm, the bounding box for the ground truth is compared to the bounding box drawn for the prediction. IoU is calculated by dividing the area of overlap of the two boxes by the area of the union of the boxes. By setting IoU to different threshold values, we can produce different scores according to our target localization goals. The standard threshold when comparing object classification models against each other is IoU = 0.5. If a prediction box has IoU $\geq$0.5 and has the correct object classification, it is considered a TP. If a prediction box has IoU < 0.5 or duplicate prediction boxes are generated, it is considered a FP. Finally, if there is no classification or IoU $\geq$ 0.5 with incorrect object classification, it is considered a FN.

Now that a threshold is set for precision and recall being calculated, mAP can be calculated using the following formula:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \quad (3)$$

where N is the number of classes of objects and AP is average precision for a class. To calculate AP, we graph the Precision-Recall curve for each class and take the area under the curve. As depicted in Fig. 3, the Precision-Recall curve is generated by graphing the trade-off between precision and recall at decreasing confidence thresholds. Since both precision and recall scores are between 0 and 1, the highest score for AP is also 1, with a higher score indicating high precision and recall. mAP is calculated by taking the average of the AP for each class.

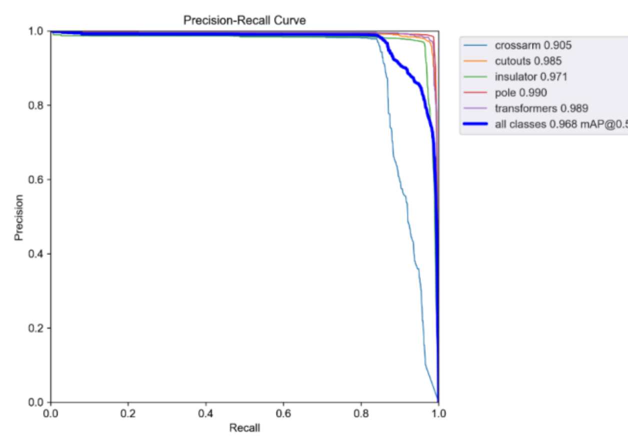

Fig. 3. Precision-Recall Curve After Training for crossarm, cutouts, insulator, pole, transformer, and all classes 0.968 mAP@0.5.

Since different mAP scores can be generated using different values for IoU threshold, YOLOv5 provides two scores called mAP_{50} and $mAP_{50\text{-}95}$. mAP_{50} is calculated using the standard IoU threshold of 0.5 and is meant to serve as the standard mAP score for evaluation. $mAP_{50\text{-}95}$ calculates AP at thresholds from 0.5 to 0.95 in increasing intervals of 0.05 and averages them to generate the AP for each class. By using ten different IoU thresholds of increasing value,

mAP_{50-95} serves as a challenge metric where a higher score is obtained through high localization.

VI. RESULTS

The performance of our YOLOv5 object classification models perform very well as indicated in Tables III and IV. Except for the YOLOv5n model, all models achieved a mAP_{50} score of 0.95 or higher with the validation set. Despite being the lowest scoring model, the YOLOv5n model manages to only fall behind by a small margin compared to YOLOv5s which is the second lowest. We observe that as model size increases, precision typically stays within the same range while the other scores improve for the validation set. We also see that the 1280-resolution models always perform better than their 640-resolution counterparts at the cost of significantly longer training time.

When we compare just the 640-resolution models against each other, the scores provide us with insight on which metrics benefit the most with increasing network depth. The precision, recall, and mAP_{50} scores using validation set for the YOLOv5m, YOLOv5l, and YOLOv5x models stay roughly the same with only minor differences. However, the mAP_{50-95} score increases with increasing network depth. This indicates that at the medium model the algorithm has reached a limit on how accurately it can detect objects using our custom dataset at the 640 resolutions, but increasing network depth allows the algorithm to draw bounding boxes closer to the ground truth boxes resulting in better localization of object classification. This is further supported when we observe the same behavior with the testing set scores. This helps in narrowing what model to select for an aerial inspection application. Within the constraints of a 640-resolution model, YOLOv5m provides the best object classification accuracy but choosing a larger model will provide better localization if that is a desired trait.

TABLE III
YOLOv5 VALIDATION SET SCORES

	Precision	Recall	mAP_{50}	mAP_{50-95}	Training hrs
YOLOv5n	0.957	0.897	0.941	0.669	4.3
YOLOv5s	0.958	0.919	0.95	0.706	7
YOLOv5m	0.962	0.932	0.957	0.727	12
YOLOv5l	0.959	0.935	0.957	0.733	20.3
YOLOv5x	0.962	0.939	0.957	0.738	36.5
YOLOv5n6	0.961	0.925	0.959	0.719	16.7
YOLOv5s6	0.959	0.942	0.964	0.74	23.1
YOLOv5m6	0.956	0.949	0.965	0.751	50.3

Comparing the 1280-resolution models to the 640-resolution models, we can see that the higher resolution model can surpass the accuracy limits observed in the lower resolution model at the cost of significantly more training time. The increased training time is largely because a 1280-resolution image is quadruple the size of a 640-resolution image. More computational resources are required to process and augment the data during training. The larger images also slow down training time even more in situations where a smaller batch size is selected due to the limited computational resources of the hardware. The benefit to using a larger resolution model is the algorithm can extract more information from the increased number of pixels. Although precision scores remained in the same range as the 640-resolution models, recall and both mAP scores of the YOLOv5s6 and YOLOv5m6 models surpassed those of the YOLOv5x model by a noticeable margin. The only outlier is the precision for the YOLOv5m6 model for the validation set which is the lowest achieved value. However, this can be due to training variances or differences in batch size. Using testing results as a more practical evaluation demonstrates clearer increases in performance across all models. Although the YOLOv5n6 model performs better than its 640-resolution counterpart on every metric, it does not yield higher scores than the YOLOv5m model which achieves similar scores with less training time.

TABLE IV
YOLOv5 TESTING SET SCORES

	Precision	Recall	mAP_{50}	mAP_{50-95}
YOLOv5n	0.964	0.914	0.954	0.712
YOLOv5s	0.966	0.932	0.962	0.749
YOLOv5m	0.972	0.942	0.968	0.773
YOLOv5l	0.971	0.949	0.967	0.779
YOLOv5x	0.971	0.949	0.968	0.781
YOLOv5n6	0.973	0.946	0.974	0.769
YOLOv5s6	0.973	0.959	0.979	0.788
YOLOv5m6	0.972	0.966	0.98	0.797

Although 100 epochs were used to train all of our models, some models converge to their best validation scores much earlier. The same sized models for both resolution configurations are observed to converge at approximately the same rate, i.e. YOLOv5s converges at approximately the same rate as YOLOv5s6. Once again observing the 640-resolution models, the small and nano models converge at approximately epoch 100 while the larger models converge within the range of epoch 40 to 60 with larger models converging earlier. This makes perfect sense since a deeper network will be able to extract more information and converge faster. Although 100 epochs were chosen for training all models for an equal training environment, training for a reduced number of epochs on larger models would reduce training time proportionally to the reduction of epochs. If we are to assume the medium, large, and extra-large models can converge at approximately epoch 50, training time will be reduced by half. The YOLOv5m model was previously signified as the most accurate model ignoring improved localization with the large and extra-large models. If training time is reduced by half to 6 hours, the medium model would achieve noticeably improved scores compared to the small model for approximately the same training time.

Finally, we can compare the metrics of our YOLOv5 object classification models with those produced by the work of others. Although the work is similar in scope to the research performed in this paper, the dataset and labels used are completely different. The authors chose to label six classes of objects: transformer banks, high voltage bushings, low voltage bushings, arresters, radiator fins, and cutoff fuses. Their dataset was also obtained in one university campus, which will mean background conditions in images will not differ greatly. The authors also performed data augmentation to achieve a dataset of 10,400 images, which is almost a third of the size of the dataset used in this paper without augmentation. The top scores achieved by their work across all models with a testing set were 0.9735 for precision, 0.949 for recall, and 0.9686 for mAP_{50}. No scores for mAP_{50-95} were provided. When comparing the scores with our testing set scores, we can see that all scores are matched by YOLOv5m and larger. Recall and mAP_{50} are surpassed by a noticeable margin with our 1280-resolution models. Although classes detected between their work and ours are different, these observations demonstrate how a larger more varied dataset can achieve superior results with the same algorithm.

VII. CONCLUSION AND FUTURE WORKS

With recent developments in computer vision and machine learning, the process of inspecting electric utility assets can be both expedited and made safer using UAVs in conjunction with sustainable and resilient communication networks. The object classification models produced in this paper serve as a baseline for training with the Drone-based Distribution Inspection Imagery dataset from EPRI. The dataset is the largest dataset to capture electric utility assets to date. Future works aim to compare the baselines set by this paper with other algorithms such as YOLOv4- Tiny, SSD, and other popular and successful object classification models and algorithms. Such comparisons can help evaluate which method is most appropriate for deployment on UAVs. Additionally, future research will attempt to develop object detectors which will also be able to detect faulty equipment using transfer learning. As previously stated, data capturing faulty equipment is scarce and inadequate for training a robust model. However, by employing transfer learning, we can build off a strong foundation of models trained using the large dataset from EPRI and adjust weights using a smaller dataset to adapt them for fault detection.

Moreover, together with federated learning, efficient communication and UAV networks can be developed as part of future works for inspecting electric utility assets. Furthermore, this scheme has practical applications for enhancing wildfire surveillance and prevention.

ACKNOWLEDGMENT

The authors would like to thank Aerospace Corporation for the funding of this research. The authors also wish to acknowledge Mr. Juan Castaneda and Mr. Ricson Chude of Southern California Edison for their support for this project.

REFERENCES

[1] A. Hoffy, S.-C. Kwon, and H.-G. Yeh, “Federated/Deep Learning in UAV Networks for Wildfire Surveillance,” in *Proc. Wireless Telecommunications Symposium,* Boston, MA, April 2013, pp. 1-10.

[2] Takaya, Kenta, Hiroshi Ohta, Valeri Kroumov, Keishi Shibayama, and Masanao Nakamura. ”Development of UAV system for autonomous power line inspection.” In 2019 23rd International Conference on System Theory, Control and Computing (ICSTCC), pp. 762-767. IEEE, 2019.

[3] Chen, Binghuang, and Xiren Miao. ”Distribution line pole detection and counting based on YOLO using UAV inspection line video.” Journal of Electrical Engineering & Technology 15 (2020): 441-448.

[4] Sumagayan, Moheddin U., Earl Ryan M. Aleluya, Christian Y. Cahig, Lester G. Librado, Rohanni B. Mangorsi, Margie S. Arda, Noel R. Estoperez, and Carl John O. Salaan. ”You Only Look Once on Power Line Components: An Object Detection Using Unmanned Aerial Vehicle.” In 2021 IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), pp. 1-6. IEEE, 2021.

[5] Feng, Zhiqiang, Li Guo, Darong Huang, and Runze Li. ”Electrical insulator defects detection method based on yolov5.” In 2021 IEEE 10th Data Driven Control and Learning Systems Conference (DDCLS), pp. 979-984. IEEE, 2021.

[6] Huang, Zengrui, Shilin Hu, and Lei Zhang. ”Fault Detection of insulator in distribution network Based on YOLOv5s Neural Network.” In 2022 International Conference on Artificial Intelligence and Computer Information Technology (AICIT), pp. 1-5. IEEE, 2022.

[7] Souza, Bruno Jose´, Stefano Frizzo Stefenon, Gurmail Singh, and Roberto Zanetti Freire. ”Hybrid-YOLO for classification of insulators defects in transmission lines based on UAV.” International Journal of Electrical Power & Energy Systems 148 (2023): 108982.

[8] Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. ”You only look once: Unified, real-time object detection.” In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779-788. 2016.

[9] Ultralytics. “Yolov5/Readme.md at Master Ul-tralytics/yolov5.” https://github.com/ultralytics/yolov5/blob/master/README.md GitHub, Feb 2023.

[10] Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Per- ona, Deva Ramanan, Piotr Dolla´r, and C. Lawrence Zitnick. ”Microsoft coco: Common objects in context.” In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pp. 740-755. Springer International Publishing, 2014.

[11] Lewis, Dexter. “EPRI Distribution Inspection Imagery.” IEEE Data- Port. IEEE, July 20, 2022. https://ieee-dataport.org/open-access/epri- distribution-inspection-imagery.

[12] Lewis, Dexter. “EPRI Distribution Inspection Imagery.” Kaggle, June 15, 2022. https://www.kaggle.com/datasets/dexterlewis/epri-distribution- inspection-imagery.

[13] Ultralytics. “Tips for Best Training Results.” GitHub. Accessed March 16, 2023. https://github.com/ultralytics/yolov5/wiki/Tips-for- Best-Training-Results.