2023 Latin American Robotics Symposium (LARS), 2023 Brazilian Symposium on Robotics (SBR), and 2023 Workshop on Robotics in Education (WRE) | 979-8-3503-1538-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/LARS/SBR/WRE59448.2023.10333013

Recognition, location, and depth estimation of objects in electrical substations

1st Gilberto Lexinoski
CPGEI-UTFPR, LACTEC
Curitiba, Brazil
gilberto.2015@alunos.utfpr.edu.br

2nd Marco A. S. Teixeira
PPGIa -PUCPR
Curitiba, Brazil
simoes.marco@ppgia.pucpr.br

3rd André E. Lazzaretti
CPGEI-UTFPR
Curitiba, Brazil
lazzaretti@utfpr.edu.br

4th Luiz F. R. B. Toledo
LACTEC
Curitiba, Brazil
luiz.toledo@lactec.com.br

5th Heitor S. Lopes
CPGEI-UTFPR
Curitiba, Brazil
hslopes@utfpr.edu.br

6th Gustavo Travassos
NEOENERGIA
Curitiba, Brazil
gtravassos@neoenergia.com

7th Mario Jambeiro Fonseca
NEOENERGIA
Curitiba, Brazil
mjfonseca@neoenergia.com

***Abstract*—Considering the increase in using Robotic Systems with the capacity to recognize and locate objects in different areas of industry, this work presents an object recognition and location in an Electrical Energy substation (SE) developed in a virtual environment, with a strategy for depth estimation by PointCloud (PC). In order to develop the environment, the CoppeliaSim software of Coppelia Robotics was used, allowing the implementation of a unit Robotic System for displacement and data acquisition in the virtual SE environment developed. Using the Robot Operating System (ROS) platform, we send commands to the UVS displacement and export the data acquired by the stereo RGB sensor to data processing in a Python environment. A YOLOv7 (You Only Look Once) model was used for object recognition and location tasks, trained to recognize and locate six types of objects present in the virtual SE environment with Bounding Boxes (BB). During the recognition and location of objects predictions, good results of the mean Average Precision (mAP) were achieved. Next, the depth estimations are performed by correlating the coordinates of the points related to the BB regions. Finally, the algorithm was implemented to predict during operation to evaluate this development's advantages and disadvantages, looking to improve the object recognition and location task during robot navigation.**

***Index Terms*—Depth Estimation, Electrical Substation, Object Recognition, Object Location, Point Cloud, Robot Operating System, Stereo Vision, Unmanned Vehicle System, YOLOv7.**

I. INTRODUCTION

The use of Machine Learning (ML) Algorithms and robotics systems for different areas of technology has increased significantly in recent years. State-of-the-art (SOTA) approaches present that using robot systems in electrical power systems also has led to a more Reliable and Safe Grid [1].

It is possible to diagnose some installation issues preventively using different ML strategies. This sometimes allows the operator to execute inspections without submitting to the safety risks in these environments. Therefore, the operator does not have to transit to the SE, often installed in places far from urban areas. Unmanned Vehicle Systems (UVS) are often used in SE environments. Recently, in [2], the authors proposed a UVS to monitor an SE. The unit's navigation used a dual antenna Real-time Kinematic Positioning (RTK) as the primary device. In addition, a Light Detection and Ranging (LIDAR) set was also used to support the navigation when the RTK was unavailable. Finally, an Inertial Measurement Unit (IMU) was used to measure the roll and pitch angles of the robot. The system offered a reliable strategy for autonomous navigation in the SE environment, but the authors did not use an RGB sensor for the autonomous navigation. The RGB sensor was used only for support operating in remote control mode.

While RTK, LIDAR, and IMU systems are still generally used as primary sensors for the navigation of a UVS, some researchers also considered the RGB sensor as support during obstacle avoidance by a UVS. In [3], a segmentation algorithm for obstacle detection on the road in a SE environment is proposed, with a pixel-level matching of obstacles segmentation mask and depth data. Although this work brought a comparison between Faster Recursive Convolutional Neural Network (R-CNN) and the You Only Look Once version 3 (YOLOv3) to estimate the obstacle distances with a pixel-level matching, it is still limited at obstacles present only in the route of the UVS.

Although the RGB sensor can offer support during obstacle detection for a UVS, during its use in an operated system in an SE environment, especially when near live lines, an ML algorithm for processing RGB sensor data can enhance the safety of the operator, considering that a computer vision algorithm can alert in case of risks of live line contact. In order to evaluate ML strategies in SE environments without having a real UVS, nor the need to submit to the safety risks present in SEs, which require proper training for access, we can use CoppeliaSim software from Coppelia Robots. Hence, developing a virtual SE environment to implement a virtual UVS to evaluate different strategies is possible. Furthermore,

The Authors appreciate the financial support from the project PeD_2289_SE_NEOENERGIA_00047-0094-2022_SENSOR-CAMINHÃO-LV of the Research and Development program of the ANEEL.

979-8-3503-1538-7/23/$31.00©2023 IEEE

the CoppeliaSim software allows the Robot Operating System (ROS) platform to send commands to the UVS to export data from UVS sensors during the operation.

The research [4] brings an example using the CoppeliaSim software to implement a cooperative robotic system between a UVS unit and an Unmanned Aerial System (UAS) in virtual agricultural environments. The algorithm was evaluated with a UAS operating in a real environment, recognizing and locating insect traps installed in olive trees using a YOLOv7 model. Although the focus of the work was the robotic systems optimizations, the algorithm can bring significant innovation for automation in agriculture. For example, with data collection, a classification offers support when exchanging for a new trap is necessary.

Other researches to before are also available using RGB sensors for the location of meters in an SE environment. In [5], an algorithm was implemented using a YOLOv3. On the other hand, in [6], an algorithm using the Scale Invariant Features Transform (SIFT) was implemented.

In complementing the related works and others that also consider UVS units for navigation em SE environments, such as [7], [8], and [9], this paper evaluates the use of a Stereo RGB sensor for depth estimation of equipment in the SE environment. In addition, the algorithm will implement in a virtual SE environment. It allows evaluating strategies and, next, the behavior of the UVS in the face of different situations that can correspond to a situation in the real environment. This evaluation will support route optimizations in a real UVS about trench present in SE and generate safety alerts when close to equipment with live lines exposed.

In order to build the dataset, we use the online resource RoboFlow. This tool offers support when building datasets, as it indicates the objects of interest by Bounding Boxes (BB). Then, we can export the data with the position of the pixels that delimit the objects in the images used in model training. Using the mean Average Precision (mAP) metric is possible to evaluate the results quantitatively and, through the Confusion Matrix, understand the results and possible limitations.

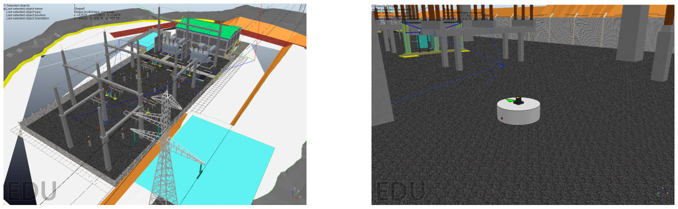
After training a model for recognition and location to objects of interest, it is integrated with the UVS developed in the virtual environment to execute this task during operation for recognition and location of equipment from image samples acquired in the virtual SE environment. Next, processing the Point Cloud (PC) generated from the RGB stereo camera implements a depth estimation task. Thus, it is possible to evaluate the feasibility of integration with other resources, using the recognition, location, and depth estimation of objects in route optimization to UVS in the face of obstacles or even improving the safety when operating manned systems in the SE environment.

The next caption will present the following items: Methods, which will present the building of the scenario, preparing the dataset, training the model, performance metrics, and, finally, depth estimation; Results, which will present as quantitative evaluations with scores for the mAP and the confusion matrix, to qualitative evaluations will be presented predictions results, the depth estimation, and computational complexity also will be presented; Conclusions and Next Steps.

II. Methods

A. Building the scenario

Using the CoppeliaSim software, a virtual environment of a SE was developed to implement the UVS unit then. Basic actuators and sensors were used, such as two motor wheels (left and right) and a stereo RGB camera. Fig. 1a presents an overview of the virtual SE model developed in the CoppeliaSim software. On the other hand, Fig. 1b presents the UVS unit developed for displacement and image acquisition from the virtual SE environment.



(a) SE

(b) UVS

Fig. 1: Simulated model of the SE and the UVS.

For data acquisition in the virtual SE environment, the following topics were used to interface with the UVS unit through the ROS platform: *joy*, *cmd_vel*, and *kinect\rgb*. The *joy* and *cmd_vel* topics were used to send commands to UVS displacement in the virtual SE environment. At the same time, the *kinect\rgb* topic was used to export the RGB image data from the stereo RGB camera installed onboard the virtual UVS unit.

B. Preparing the dataset

For this work, a model was trained to recognize and locate the following objects: (i) transformer; (ii) circuit-break; (iii) oil-level; (iv) disconnector; (v) trench; and (vi) robot-ramp. To compose the dataset, 600 acquisitions were made from the RGB sensor installed in the UVS, where we have 100 samples with a focus on each one of the objects, varying the viewing angle and distances to the object. Using the official Roboflow website, the BB indications were made for each object present in each of the 600 samples. Subsequently, the data was exported using the YOLO v7 PyTorch format, which generates the code lines necessary to access the dataset by the Roboflow Application Programming Interface (API). This allows a simple interface with the training model using YOLOv7 layers, available in Python.

Table I summarizes the number of instances for each one of the classes and the total. There are more significant numbers of instances than the number of in-focus images because samples that focus on one object generally have other objects too.

Finally, the dataset was split into smaller datasets (subsets) for training, validation, and testing. The following proportion for the subsets was used: Training: 70% (420 images); Validation: 20% (120 images); and Testing: 10% (60 images).

TABLE I: Instances for each class and total

Class	Focused images	Total of instances
transformer	100	265
circuit breaker	100	452
oil level	100	230
disconnector	100	399
trench	100	359
robot ramp	100	394
Total	**600**	**2099**

C. Training the model

In order to train the model, the How to Train YOLOv7 on a Custom Dataset procedure was used, also available on the official Roboflow website. This algorithm uses a pre-trained YOLOv7 model with the COCO dataset as a starting point and then executes the retraining of the last layers of the model. Therefore, the frozen layers of the YOLOv7 model were used as feature extractors to train the last layers for recognition and location of objects of interest.

D. Performance metrics

The mean Average Precision (mAP) metric was used to evaluate the model's performance. In addition, the Confusion Matrix was used to support the visualization of results. The mAP can be calculated according to Eq.(1):

$$mAP@\alpha = \sum_{i=1}^{n} AP_i. \tag{1}$$

According to [10], the mAP metric is the mean from values to Average Precision (AP) for each class. In comparison, each class's AP is equivalent to the Area Under the Precision Recall-Curve (AUC-PR), calculated as the own name under the Precision-Recall Curve (PR-Curve). Finally, all these variables are directly dependent on the Intercession over Union (IoU) Metric, which corresponds to the intersection of the predicted and actual Area over the union of both areas, according to Eq. (2):

$$IoU = \frac{A_{truth} \cup A_{pred}}{A_{truth} \cap A_{pred}}. \tag{2}$$

The IoU metric needs a threshold (α), through this parameter is defined whether a prediction is classified as:

- True Positive (TP) when IoU $\geq \alpha$;
- False Positive (FP) when $0 <$ IoU $< \alpha$;
- False Negative (FN) when $\alpha = 0$ or when there is an object that was not detected.

True Negative (TN) is not used in object recognition and location tasks, as it corresponds to the background, which is not demarcated in the prediction. A model is considered fitting when it results in mAP values close to 1, and the higher is the threshold value (α), which also varies between 0 to 1. This work used the mAPs with α = 50% (mAP@[.50]). In addition, the average of 10 values mAP was calculated, with the value of α varying from 50% to 95%, with a step of 5%. This strategy is generally denoted as mAP@[.50:.5:.95]. In order to present the Confusion Matrix, the α = 50% was used.

E. Depth estimation

In order to depth estimation, data from the stereo sensor installed onboard UVS was used, which was published by a float32MultiArray message. A node was used to convert from float32MultiArray message to the PointCloud2 message, to then calculate the depth of each pixel in the Depth estimation by point cloud module. The Fig. 2 presents the communication diagram during the depth estimation task, the labels in the connections arrows represents the topics used in ROS plataform.

The Depth estimation by the point cloud node receives the PointCloud2 message and detection message containing the BB locations. Through the position of the BB, it is possible to know the depth of the pixels in the same region of the BB detection. Hence, it is enough to calculate the module from the x, y, and z coordinates of the points corresponding to these pixels.

Then, the Depth estimation by the point cloud module returns the depth estimation data to be inserted into the result of predictions by the YOLOv7 model node. As a result, we add depth estimation to the predictions made in the YOLOv7 model, thus presenting recognition, location, and depth estimation of objects.

During the depth estimation by PointClound (PC), a parameter to downsample (ds) was used to improve the execution time in this task by reduction of points in the PC. Fig. 3a presents a Depth image for the region of a BB in its original size. In contrast, Fig. 3b presents a Depth image of the same region with $ds = 8$. In other words, one every eight rows and one every eight columns were considered allowing a reduction of 64 times the PC, bringing a significant improvement in the execution time during depth estimation without relevant losses to the depth information.

During the depth estimation for an object, a strategy based on the mean for a certain percentage of the nearest points was used. Therefore, verifying that the floor biased the depth estimation was possible because it concentrated various nearest points. Thus, a center rate (CR) parameter was implemented to delimit the central region of the BB to be considered in the mean calculation, which allows noise reduction during the depth estimation, as the region floor was discarded. Fig. 4a presents a depth image of a region of a BB in the original size, while Fig. 4b presents the region from the same BB applying $CR = 0.70$.

Finally, it was possible to perform the depth estimation for each BB based on the mean of the 10% nearest points. After defining the CR delimitation at 70%, it allowed performing the depth estimation with less noise, as the floor region was eliminated. In addition, with the reduction of the PC by 64 times, it was possible to perform the recognition, location, and depth estimation in a reasonable time.

III. Results

A. Quantitative evaluation

In Fig. 5, the mAP@[.50] and the mAP@[.50:.5:.95] values are presented during the progress of the training epochs

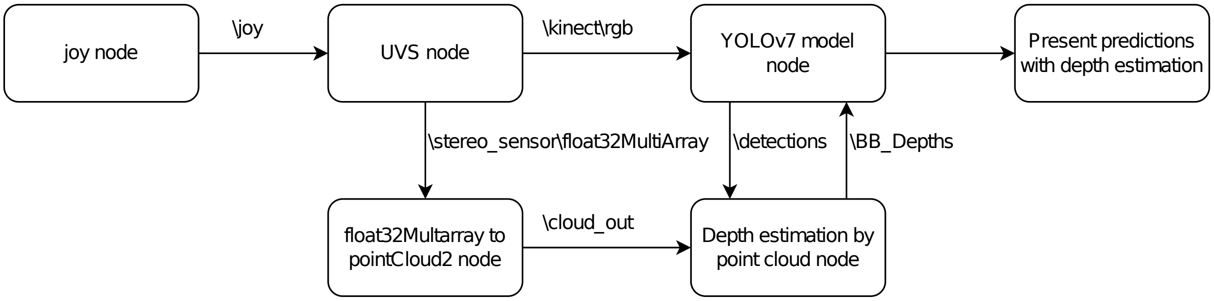
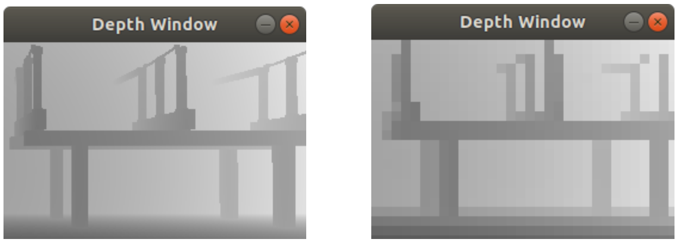

joy node
\joy
UVS node
\kinect\rgb
YOLOv7 model node
Present predictions with depth estimation
\stereo_sensor\float32MultiArray
\detections
\BB_Depths
float32Multarray to pointCloud2 node
\cloud_out
Depth estimation by point cloud node

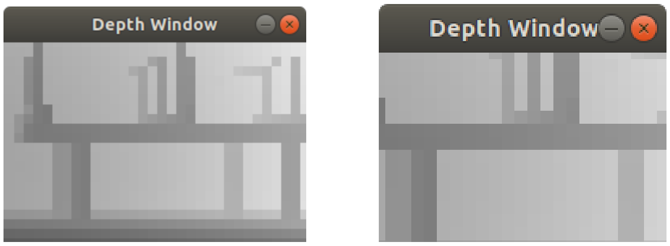
Fig. 2: Communication diagram of the System.



(a) Original Resolution.

(b) Downsample with $ds = 8$.

Fig. 3: PointCloud reduction.



(a) BB with Original Size.

(b) BB with $CR = 0.7$.

Fig. 4: Center Rate of 70%.

using validation subsets. One can observe the model's good performance (mAP > 0.6) in the recognition and location of objects in the virtual SE environment. The mAP@[.5] metric reached values above 0.9, while the mAP[.50:.5:.95] metric reached values above 0.7. Fig. 6 presents the Confusion Matrix that shows the proportion of correct predictions for each one of the classes.

It is possible to observe that a few instances were classified incorrectly; a small proportion of backgrounds were classified as objects; on the other hand, a small proportion of objects were classified as background. This is related to how much BB delimits the objects. A small proportion of the circuit-break instances (0.01) were confused with the disconnector, and the same proportion of the trench instances (0.01) was confused with the robot ramp. This can be explained because the disconnectors are installed near the circuit breaks; many images captured these objects with overlap, and the same occurs between the trench and robot ramp.

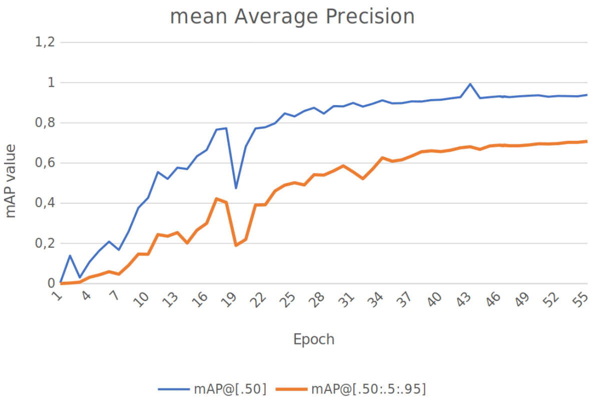
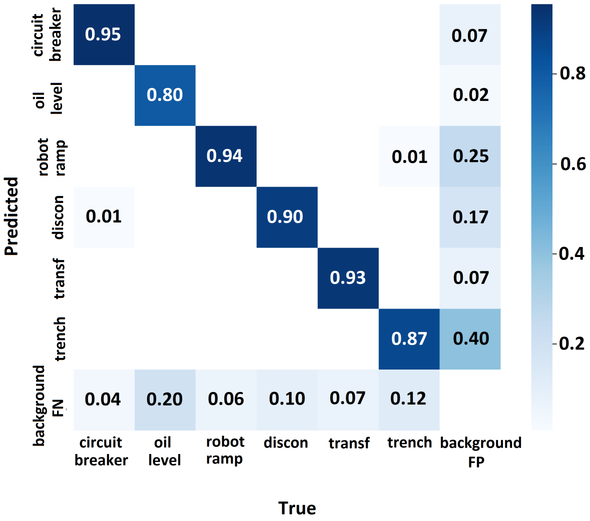


Fig. 5: mAP@[.50] and mAP@[.50:.5:.95] curves.



	circuit breaker	oil level	robot ramp	discon	transf	trench	background FP
circuit breaker	0.95						0.07
oil level		0.80					0.02
robot ramp			0.94			0.01	0.25
discon	0.01			0.90			0.17
transf					0.93		0.07
trench						0.87	0.40
background FN	0.04	0.20	0.06	0.10	0.07	0.12	

Fig. 6: Confusion Matrix with α = 50%.

B. *Qualitative evaluation*

Implementing the trained YOLOv7 model for recognition and location of objects of interest with the algorithm for depth estimation in the UVS allows performing the displacement in the virtual SE environment while recognition, location, and depth estimation of objects occurs. Fig. 7 shows a prediction where five objects were observed. The transformer was recognized with the lowest confidence value of 0.72, and it is possible to observe that this equipment is the most distant at 4.39m. On the other hand, one of the robot-ramp was recognized with the best confidence value of 0.90 and corresponded to the nearest object with 1.02m.

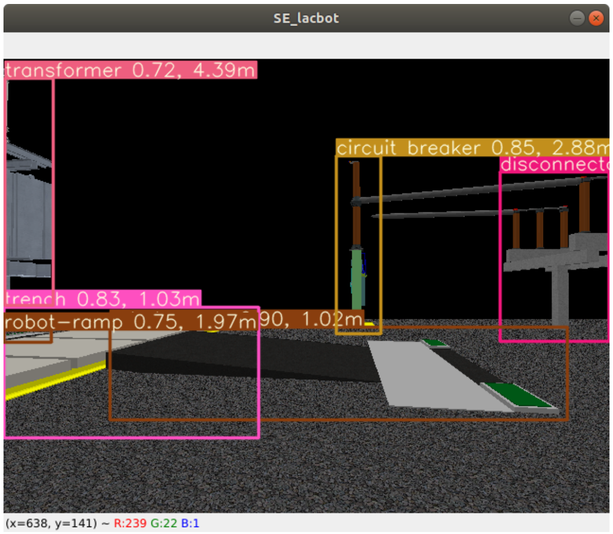


Fig. 7: Recognition, location and depth estimation of objects.

In Fig. 8, a case helps explain the results presented in the confusion matrix, as an overlap between circuit break and disconnector occurs. In this case, all objects presented high confidence values. Furthermore, this case presents a situation in which the Center Rate Parameter supports the estimating of the disconnector depth because this adjustment allowed to discard of the region in the right of the BB, which would be a very high noise considering that the circuit breaker is a distance of 0.90m even so still a distance of 1.52m was estimated for the disconnector.

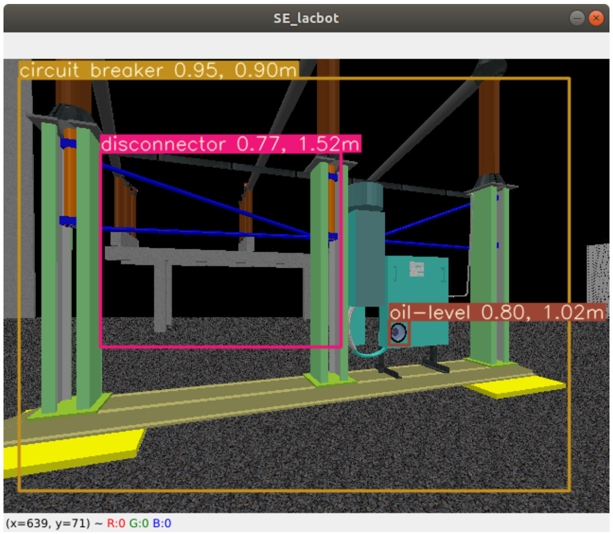


Fig. 8: Recognition, location and depth estimation of objects.

On the other hand, Fig. 9 brings a clear case that the nearest disconnector is present in a large proportion of the other BBs. Thus, both the robot ramp and the circuit breaker presented depth estimation affected by the distance of the nearest disconnector.

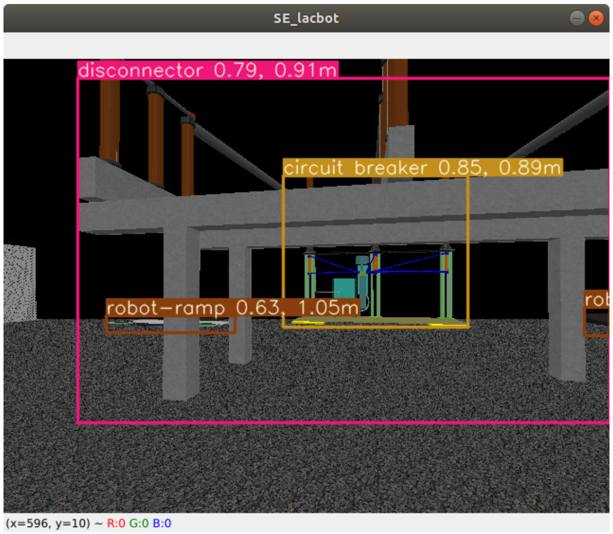
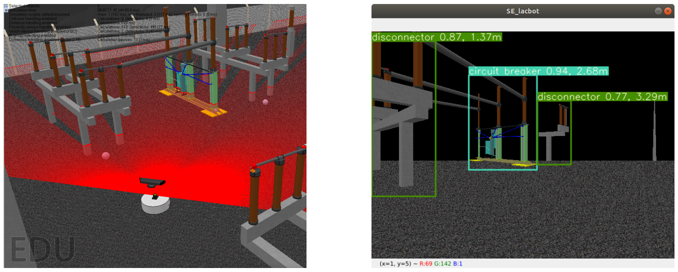


Fig. 9: Recognition, location, and depth estimation of objects.

This strategy demonstrates an excellent way to estimate the characterization of an object, which can lead to considerable advances in the route optimization of robotic systems. On the other hand, the strategy used for depth estimation by PC can be affected when there are BB overlaps so that new strategies can be evaluated.

C. Depth estimation evaluation

In order to evaluate the depth estimation, the distances between the object and the stereo camera positions in the virtual SE environment were used. Fig. 10a presents the UVS positioned to calculate the distances. Fig. 10b shows the predictions performed by the algorithm when the actual positions were annotated.



(a) UVS positioned.

(b) Depth estimation.

Fig. 10: Evaluation of the depth estimation.

Table II presents the error in percentage between the actual depth of the objects in the virtual SE environment and the predicted depth performed by the algorithm. Although the error increases as the distance increases, it is still possible to observe that these errors have small values even for greater distances.

TABLE II: Depth Analysis

Object	Real Depth (m)	Predic. Depth (m)	Error (%)
Disconnector 1	1.338	1.380	0.607
Circuit Break	2.736	2.680	2.063
Disconnector 2	3.403	3.290	3.326

D. Computational Complexity

In order to compare the computational complexity, Table III presents values to the elapsed time for fifty iterations considering three situations. First, the values for recognition and location task performed by the YOLOv7 model trained are presented. Next, the values considering the YOLOv7 and depth estimation tasks are detailed without the downsampling. Finally, the values for YOLOv7 and depth estimation tasks are presented with a downsampling of $DS = 8$.

TABLE III: Computational Complexity Comparation

Computing Cost	Min (s)	Max(s)	Mean (s)	P95 (s)
YOLOv7	0.018	0.023	0.020	0.023
YOLOv7, Depth	2.940	3.497	3.139	3.273
YOLOv7, Depth, (DS=8)	0.702	1.059	0.921	1.021

It is possible to observe that the predictions performed by the YOLOv7 model are very relatively fast, where 95% of the iterations were performed in less than 0.023 seconds. On the other hand, performing the depth estimation increased the elapsed time, where 95% of the iteration was performed in less than 3.273 seconds.

Finally, performing the downsample with $DS = 8$, the elapsed time decreased three times, where 95% of the iterations were performed in less than 1.021 seconds, a reasonable time for applications in robotic systems. Notably, these elapsed times were achieved running the virtual SE environment simultaneously in the same processor. With a dedicated processor, improving the elapsed time by iteration is possible. However, an analysis focused on the embedded system used should be performed.

IV. Conclusions and Next Steps

This paper presented a strategy for object recognition, location, and depth estimation in a virtual SE environment. It can collaborate to improve the efficiency and safety of systems considering that the algorithm is performed in a reasonable time. It can be implemented in robotic systems, offering support during a navigation task so that the system can follow the best route or keep the distance of equipment with a live line present, for example.

The recognition and location of objects task presented good scores for the mAP metrics, reaching values above 0.9 for mAP@[.50] and above 0,7 for the mAP@[.50:.5:.95] metrics. In addition, during the depth estimation, the worst prediction presented a small error 3.326% even for a distance of 3,403m.

On the other hand, the depth estimation strategy used is biased when different BB overlap in the image because it considers the closest points present in the regions of the BB that refer to the nearest equipment. Therefore, the depth estimation is reliable only when no BB overlay exists. In addition, new analyses can performed to optimize depth estimation through PC processing when BB overlaps exist.

The subsequent work is the development of a dataset by sampling in a real SE environment. This step will acquire more images focused per class, considering different view angles and luminosity conditions. Next, train a new YOLOv7 model for the recognition and location of the real equipment in the SE, making it possible to evaluate the algorithm's behavior and difficulty of the implementation in facing the variation in real environment or even variations in physical conditions between equipment of the same model and based on the results we can propose improvements for this algorithm. Finally, the depth estimation can implemented as presented in this work.

Still using the virtual SE environment developed, it will be possible to evaluate new strategies to process the depth estimation, offering support to the mean of the percentage of the nearest points. In addition, depth estimations during navigation tasks for UVS or even for generating safety alerting when operating manned systems that need to operate close to equipment with exposed live lines will be evaluated.

References

[1] O. Menendez, F. A. Auat Cheein, M. Perez, and S. Kouro, "Robotics in power systems: Enabling a more reliable and safe grid," *IEEE Industrial Electronics Magazine*, vol. 11, no. 2, pp. 22–34, 2017.

[2] P. Dandurand, J. Beaudry, C. Hébert, P. Mongenot, J. Bourque, and S. Hovington, "All-weather autonomous inspection robot for electrical substations," in *2022 IEEE/SICE International Symposium on System Integration (SII)*, pp. 303–308, 2022.

[3] X. Hongsheng, C. Tianyu, Z. Qipei, L. Jixiang, and Y. Zhihong, "A deep learning and depth image based obstacle detection and distance measurement method for substation patrol robot," *IOP Conference Series: Earth and Environmental Science*, vol. 582, p. 012002, sep 2020.

[4] G. S. Berger, M. Teixeira, A. Cantieri, J. Lima, A. I. Pereira, A. Valente, G. G. R. d. Castro, and M. F. Pinto, "Cooperative heterogeneous robots for autonomous insects trap monitoring system in a precision agriculture scenario," *Agriculture*, vol. 13, no. 2, 2023.

[5] C. Wang, L. Yin, Q. Zhao, W. Wang, C. Li, and B. Luo, "An intelligent robot for indoor substation inspection," *Industrial Robot: the international journal of robotics research and application*, vol. 47, pp. 705–712, Jan 2020.

[6] H. Fang, X. Cui, L. Cui, and Y. Chen, "An adapted visual servo algorithm for substation equipment inspection robot," in *2016 4th International Conference on Applied Robotics for the Power Industry (CARPI)*, pp. 1–5, 2016.

[7] B. Su, H. Zhang, and H. Meng, "Development and implementation of a robotic inspection system for power substations," *Industrial Robot: An International Journal*, vol. 44, pp. 333–342, 05 2017.

[8] H. Wang, J. Li, Y. Zhou, M. Fu, and S. Yang, "Research on the technology of indoor and outdoor integration robot inspection in substation," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 2366–2369, 2019.

[9] B. Wang, R. Guo, B. Li, L. Han, Y. Sun, and M. Wang, "Smartguard: An autonomous robotic system for inspecting substation equipment," *Journal of Field Robotics*, vol. 29, no. 1, pp. 123–137, 2012.

[10] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. da Silva, "A comparative analysis of object detection metrics with a companion open-source toolkit," *Electronics*, vol. 10, no. 3, 2021.