RGB-D Object Classification System for Overhead Power Line Maintenance

José Mário Nishihara, Heitor Silvério Lopes, Thiago Henrique Silva,
André Eugenio Lazzaretti, Andre Schneider de Oliveira and Ronnier Frates Rohrich
Graduate Program in Electrical and Computer Engineering
Program in Computer Science
Universidade Tecnológica Federal do Paraná
Curitiba, Brazil
e-mail: {josalb}@alunos.utfpr.edu.br
{hslopes | thiagoh | lazzaretti | andreoliveira | rohrich}@utfpr.edu.br

***Abstract*—This paper presents the development and evaluation of different machine-learning models applied to classify objects in high-voltage transmission lines using depth data captured by a RealSense D415 camera. Four models, k-Nearest Neighbors (kNN), Decision Tree, Neural Network (NN), and AdaBoost (AB), were tested using simulated and real data collected in a laboratory environment. The results show that the kNN and NN models achieved robust performance, while the Decision Tree model faced significant limitations due to excessive nodes and the AB model struggled with the real-world data. Moreover, tests with real data revealed noise in the images, which affected model performance. This study also highlights the feasibility of using depth cameras for autonomous inspection tasks, potentially reducing costs and enhancing safety in high-voltage environments.**

***Keywords*-RealSense; Machine Learning; Object Classification; Transmission Lines; Autonomous Inspection.**

I. Introduction

Inspecting high-voltage power lines is critical for ensuring the safety and efficiency of electrical grids, as these structures carry large amounts of energy over long distances and are exposed to extreme weather conditions. Traditional inspection methods, such as manual climbing and drone-based monitoring, have significant limitations: while drones offer agility, they are constrained by battery life and weather conditions, whereas manual inspections, though precise, are costly and hazardous. Key challenges include monitoring cable temperature, detecting nearby obstacles, and assessing structural wear to prevent failures and reduce maintenance costs.

Robotic automation has emerged as a promising solution, enabling safer and more efficient inspections. However, a robot must overcome obstacles such as support towers and irregular structures to traverse an entire power line. Various approaches have been proposed, including modular robots with specialized locomotion units [1], transposition mechanisms [2], and caterpillar-based robots capable of climbing jumpers at 80° inclines [3]. Despite these advances, human operator intervention remains necessary, highlighting the need for greater autonomy.

Furthermore, a reliable electricity supply, directly impacted by Transmission Line (TL) maintenance, is essential for socioeconomic development. Current inspections rely predominantly on visual and manual methods, which are prone to human error and subjectivity, leading to increased service interruptions and inefficient asset management. Thus, this study proposes a predictive aerial inspection system combining advanced technologies—such as thermal, spatial, and reflectance sensing—with artificial intelligence to optimize TL monitoring. The central hypothesis is that this multimodal approach will improve the detection of critical issues—such as cable wear, vegetation encroachment, and structural anomalies—reducing operational costs and preventing power outages.

This paper explores innovative solutions for autonomous power line inspection, discussing technical challenges, recent advancements, and the feasibility of an Artificial Intelligence (AI)-supported multimodal system to overcome the limitations of traditional methods. The second section reviews the state of the art and identifies research gaps in this field. The third section describes the developed system architecture. The fourth section outlines the requirements for experimental evaluation. The fifth section analyzes feature extraction methods. The sixth section presents discussions and conclusions.

II. Related Work

Autonomous inspection requires accurate detection and classification of components using depth sensors and computer vision. Pouliot et al. [4] validated the performance of the UTM-30LX Light Detection and Ranging (LiDAR) sensor for object identification and diameter estimation. The sensor was mounted at a 45° angle under the robot, collecting 49 measurements per scan with a minimum detection distance of 0.9 meters. Their approach identified object edges and estimated diameter and distance, though no classification model was implemented.

Qin et al. [5] employed a LiDAR sensor to generate a 3D point cloud of transmission lines, isolating a single cable and using 3D region-growing segmentation for object classification. Their method achieved 90.6% classification accuracy with 98.2% precision.

Vision-based approaches have also been explored. Song et al. [6] detected broken spacers using an Red, Green, Blue (RGB) camera and morphological operations, segmenting the spacer region to determine structural integrity. Zhu et al. [7] classified dampers, spacers, and clamps using a structured

Support Vector Machine (SVM) model, achieving an accuracy of 96% for clamps and over 92% for other components.

These studies highlight the feasibility of autonomous inspection through depth sensing and computer vision. Therefore, this project develops a real-time object classification model for transmission lines using depth camera data from a *RealSense D415*. The model must operate efficiently within the robot's embedded system constraints while managing concurrent motion and sensor control.

III. System Architecture

This work is part of a broader project focused on developing a fully autonomous robot for transmission line inspection. The robot can traverse lines, overcome obstacles, and efficiently collect data. In this context, object classification is essential for enabling the robot to recognize and appropriately respond to various components of the transmission line infrastructure, such as insulators, dampers, and markers.

The camera was mounted on the robot, with objects positioned in front of it for data collection. The acquired data was collected through direct communication with the Robot Operating System (ROS), an open-source platform providing tools and libraries to streamline robotic system development, facilitating flexible integration of hardware, sensors, and control algorithms. The robot in development features two claws for controlling speed and a body responsible for executing obstacle-overcoming maneuvers. The robot's specifications are detailed in [8]. The overall operation of the system is illustrated in Figure 1.

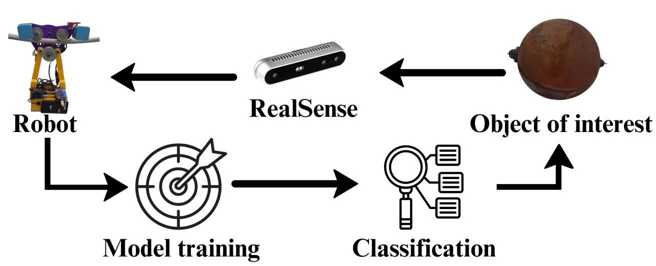

Figure 1. Operation diagram.

A. RealSense D415

The Intel RealSense D415 is a depth camera with stereoscopic infrared sensors for depth detection, widely used in robotics and automation. It captures depth maps with a resolution of 1280 × 720 and a field of view of 65° × 40°. The camera has a depth accuracy of less than 2% at 2 meters and a frame rate of up to 90 fps. Its balance of resolution and accuracy makes it suitable for object classification in transmission lines.

B. Object Classes to Be Detected

In this project, the object classes to be detected include:

- **Polymeric Insulators**: Devices used to isolate conductors in high-voltage lines are essential for ensuring the safety and efficiency of electrical systems.
- **High-Voltage Line Markers**: Visual markers placed on high-voltage lines to improve visibility and reduce accident risks.
- **Dampers**: Devices designed to mitigate vibrations and shocks in transmission systems and line supports.
- **No Obstacles**: Scenarios where no objects are detected in front of the robot, a key condition for its operation.

The comprehensive set of objects analyzed and detected within the scope of this study is illustrated in Figure 2.

Figure 2. Objects to Be Detected.

IV. Requirements for experimental evaluation

Two primary data analyses were conducted for the development of this project. Each analysis took place in different environments and had specific objectives to evaluate the feasibility and performance of the object classification model for transmission lines using the RealSense camera. The procedures and environments used in each step are detailed below.

A. Simulation-Based Problem Modeling and Analysis

The simulation aimed to replicate the realistic operating conditions of the robot on a high-voltage transmission line as closely as possible. Accordingly, the RealSense camera was positioned identically to its final deployment setup—mounted atop the robot, which was fixed to the simulated cable. The objects in the simulation were modeled with high fidelity to their real-world counterparts, matching both dimensions and shapes (see Figure 3).

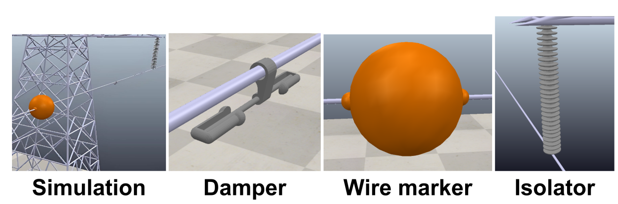

Figure 3. Components of the transmission line used in the simulation.

During the simulation, the robot was moved along the cable linearly and constantly, simulating the scenario where the robot traverses the transmission line under real-world conditions. The camera capture rate was set to 10 Hz. The RealSense camera was configured to capture depth information up to 5 meters away, using a resolution of 1280x720 pixels, returning grayscale images to the code, where each pixel represented the depth measured for that position, as shown in Figure 4.

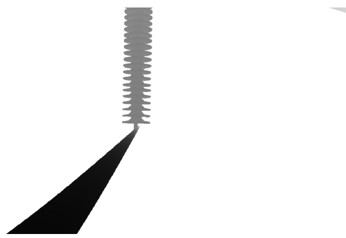

Figure 4. Depth image captured by the RealSense camera during the simulation.

The depth data collected by RealSense was saved in Portable Network Graphics (PNG) format due to the high fidelity this format offers in preserving the visual details necessary for subsequent analyses. The images generated during the simulation were used to feed the machine learning model, serving as the basis for training and evaluating the system.

B. Real-System Validation and Performance Analysis

The second stage of the project was conducted in a laboratory environment. For this experiment, a section of a transmission line was set up and divided into two segments, each 5 meters long, where typical high-voltage line objects such as insulators, markers, and dampers were fixed. These objects were arranged along the segments to closely resemble their placement in real lines, with the aim of maintaining as much fidelity as possible with the field conditions, as shown in Figure 5.

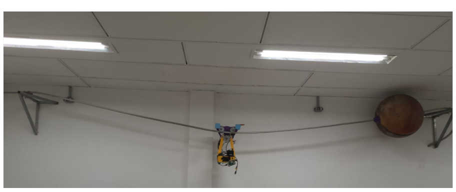

Figure 5. Setup of the experiment in the laboratory with the RealSense camera.

Due to space limitations in the laboratory, an adaptation was necessary for the position of the RealSense camera. Instead of being positioned above the line, as it would be in the real scenario, the camera was mounted on the bottom of the robot, and the data was collected as if the line were upside down. This adaptation allowed the camera to capture the objects like it would in the field, albeit with the orientation inverted. As in the simulation, the robot was controlled linearly and constantly, ensuring uniform data collection along the line segments. For this experiment, the RealSense camera was configured to operate at 15 Hz, returning grayscale images to the code, as illustrated in Figure 6.

As in the simulation, the depth data captured by the RealSense camera was stored in PNG format, preserving the necessary details for subsequent analysis and machine learning applications.

Figure 6. RGB (for reference) and depth image captured by the RealSense camera in the laboratory.

C. Simulated and Real Experimental Data Processing

The data was divided into two categories: **raw data**, which represented the originally captured images, and **processed data**, which underwent preprocessing using a simple edge detection algorithm, as illustrated in Figures 7-12.

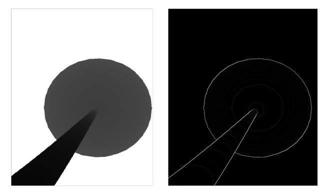

Figure 7. Algorithm applied to the simulated marker image.

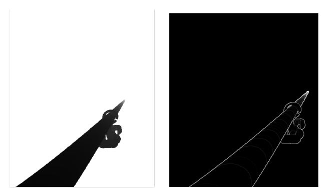

Figure 8. Algorithm applied to the simulated damper image.

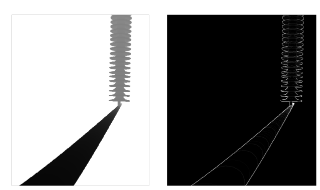

Figure 9. Algorithm applied to the simulated insulator image.

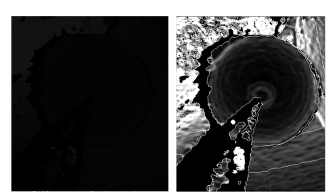

Figure 10. Algorithm applied to the real marker image.

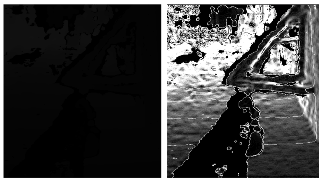

Figure 11. Algorithm applied to the real damper image.

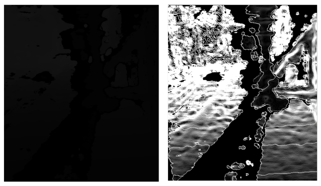

Figure 12. Algorithm applied to the real insulator image.

Data processing aimed to rapidly and efficiently simplify the images, eliminating the requirement to execute a more complex model to accomplish this task. For this purpose, an edge detection Algorithm 1 was used. This algorithm is efficient and fast, capable of highlighting the leading edges in the grayscale depth images. It calculates the depth intensity difference between adjacent pixels horizontally and vertically. Extreme values are not wished; the edge detection result is limited to a maximum value of 255.

Algorithm 1 Edge Detection Algorithm

for for each row i of the image, from bottom to top **do**
  for for each column j, from right to left **do**
    $gray\_index = i \times img\_width + j$
    if $i == 0$ or $j == 0$ **then**
      Set $img[gray\_index] = 0$
    else
      Horizontal difference:
      $diff_x = img[gray\_index] - img[gray\_index - 1]$
      Vertical difference:
      $diff_y = img[gray\_index] - img[gray\_index - img\_width]$
      Magnitude of difference:
      $derivative = \sqrt{diff_x^2 + diff_y^2}$
      Set:
      $img[gray\_index] = \min(derivative, 255)$
    end if
  end for
end for

In addition to simplifying the images, this method of deriving the image also helps normalize the data. Since the data represents depth, the distance between elements of the same object is constant, regardless of the distance from the camera to the object. This means that even if the distance between the camera and the object varies, the derivative of these distances will not be affected, keeping the edges consistent. This characteristic makes the method robust against variations in the distance between the robot and the objects, ensuring uniform edge detection independent of the camera's position. Furthermore, the algorithm allows for the visualization of objects hidden in the images, making visible those that would not be perceptible to the naked eye. Figure 13 illustrates how data normalization affects visualization, clearly showing previously visible objects.

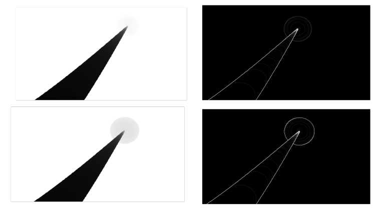

Figure 13. Example of data normalization and visualization of hidden objects (marker).

D. Organization of the Implemented Machine Learning Models

Due to the absence of a benchmark for this project, several machine learning models were tested to determine the most efficient for classifying objects on power lines. The models evaluated were **k-Nearest Neighbors (kNN)**, **Decision Tree**, **Neural Network**, and **AdaBoost**.

The *kNN* model was configured with $k = 6$, Mahalanobis distance, and distance-based weights. The neural network had three hidden layers (128, 64, and 32 neurons), Rectified Linear Unit (ReLU) activation, and used the Adam optimizer. The AdaBoost classifier was implemented with the Samme. *R* variant, suitable for multiclass classification. The decision tree was also tested due to its interpretability and computational efficiency.

1) Feature Extraction Using SqueezeNet: To enhance the representativeness of the depth data, a feature extraction step was implemented using SqueezeNet, a lightweight Convolutional Neural Network (CNN) designed for efficient feature extraction with low computational cost.

SqueezeNet was applied to grayscale depth images to extract compact visual representations, which were then used as input for the supervised learning models. This approach improved classification efficiency by focusing on relevant image features instead of raw data.

2) Feature Extraction Using Mean and Variance: As an alternative to convolutional neural networks, a statistical feature extraction approach using **mean** and **variance** of grayscale depth images was applied.

The mean represents the average depth value in each image, providing an estimate of object distance, while the variance quantifies depth dispersion, capturing surface irregularities. This method offers a computationally efficient way to summarize image characteristics, facilitating classification in resource-constrained environments.

3) Training and Validation: The **Orange** software was used for model training, a machine learning platform that offers a visual interface for creating and evaluating models. The tests were performed using **10-fold cross-validation**.

V. Analysis of Feature Extraction Methods

This section presents the results obtained after training four machine learning models using different feature extraction methods from depth images. The approaches include statistical features (mean and variance) and deep learning-based feature extraction using SqueezeNet, which is applied to raw and derivative images. The models were evaluated with simulation and real sensor measurement data, allowing for a comparative analysis of their performance under different conditions.

A. Evaluation Metrics

To assess model performance, we used several classification metrics, including Area Under the Receiver Operating Characteristic (ROC) Curve (AUC), Class Accuracy (CA), F1-Score, Precision (Prec), Recall, and Matthews Correlation Coefficient (MCC). These metrics provide a comprehensive evaluation by considering different aspects of classification performance, such as class balance, precision-recall trade-offs, and overall correlation with proper labels.

B. Simulation Results

This subsection presents the results obtained for the machine learning models trained using the simulation data. The results are divided based on the different types of images and extracted *features*, including the raw image, the derived image, and the mean and variance *features*.

1) Raw Image: The models were trained using the raw depth images without any additional processing. The results for the four tested models are presented in Table I.

TABLE I. Results of raw simulation images.

Model	AUC	CA	F1	Prec	Recall	MCC
KNN	1.000	0.998	**0.998**	0.998	0.998	0.997
NN	1.000	0.997	0.997	0.997	0.997	0.996
Tree	0.995	0.993	**0.993**	0.993	0.993	0.990
AB	0.996	0.994	0.994	0.994	0.994	0.991

2) Derived Image: The models were trained using the derived depth images, applying the edge detection technique described earlier. The results are shown in Table II.

TABLE II. Results of derived simulation images.

Model	AUC	CA	F1	Prec	Recall	MCC
KNN	1.000	1.000	**1.000**	1.000	1.000	1.000
NN	1.000	1.000	1.000	1.000	1.000	1.000
Tree	0.998	0.997	**0.997**	0.997	0.997	0.997
AB	0.999	0.999	0.999	0.999	0.999	0.998

TABLE III. Results of raw simulation images features.

Model	AUC	CA	F1	Prec	Recall	MCC
KNN	0.985	0.927	**0.926**	0.927	0.927	0.902
NN	0.985	0.906	0.904	0.914	0.906	0.877
Tree	0.923	0.867	**0.865**	0.865	0.867	0.820
AB	0.913	0.870	0.870	0.870	0.870	0.825

3) Mean and Variance of Raw Image: The models were trained using the mean and variance *features* extracted from the raw images, and the results are presented in Table III.

4) Mean and Variance of Derived Image: The models were trained using the mean and variance *features* extracted from the derived images. The results for the four models tested are presented in Table IV.

TABLE IV. Results of derived simulation images features.

Model	AUC	CA	F1	Prec	Recall	MCC
KNN	0.990	0.963	**0.963**	0.964	0.963	0.950
NN	0.988	0.878	**0.872**	0.881	0.878	0.838
Tree	0.968	0.932	0.932	0.932	0.932	0.908
AB	0.957	0.936	0.936	0.936	0.936	0.913

C. Real Data Results

In this subsection, we present the results obtained for the machine learning models trained using real data collected by the sensor. The results are divided based on different types of images and extracted *features*, including raw image, derived image, and mean and variance *features*.

1) Raw Image: The models were trained using raw-depth images without any additional processing. The results are represented in Table V.

TABLE V. Results of raw real images.

Model	AUC	CA	F1	Prec	Recall	MCC
KNN	0.992	0.940	0.940	0.940	0.940	0.920
NN	0.996	0.958	**0.958**	0.958	0.958	0.943
Tree	0.848	0.747	0.747	0.748	0.747	0.661
AB	0.825	0.738	**0.739**	0.740	0.738	0.649

2) Derived Image: The models were trained with the derived depth images, utilizing the previously described edge detection technique, as presented in Table VI.

TABLE VI. Results of derived real images.

Model	AUC	CA	F1	Prec	Recall	MCC
KNN	0.991	0.948	0.948	0.948	0.948	0.930
NN	0.996	0.952	**0.952**	0.952	0.952	0.936
Tree	0.877	0.803	**0.804**	0.805	0.803	0.737
AB	0.875	0.813	0.814	0.814	0.813	0.749

3) Mean and Variance of Raw Image: The models were trained based on the mean and variance *features* obtained from the raw images. The results for the four tested models are displayed in Table VII.

TABLE VII. RESULTS OF RAW REAL IMAGES FEATURES.

Model	AUC	CA	F1	Prec	Recall	MCC
KNN	0.926	0.752	**0.751**	0.752	0.752	0.667
NN	0.929	0.722	0.721	0.722	0.722	0.627
Tree	0.832	0.689	0.689	0.690	0.689	0.584
AB	0.785	0.680	**0.679**	0.678	0.680	0.570

4) Mean and Variance of Derived Image: The models were trained using the mean and variance *features* extracted from the derived images. It is showed in the Table VIII.

TABLE VIII. RESULTS OF RAW DERIVED IMAGES FEATURES.

Model	AUC	CA	F1	Prec	Recall	MCC
KNN	0.942	0.780	**0.779**	0.778	0.780	0.704
NN	0.933	0.728	0.728	0.729	0.728	0.637
Tree	0.846	0.717	**ee**	0.717	0.717	0.621
AB	0.813	0.721	0.721	0.721	0.721	0.626

VI. DISCUSSIONS AND CONCLUSIONS

The results highlight significant differences in the model's performance between simulated and real-world data, mainly due to variations in image capture conditions.

A. Difference Between Simulation and Real-World Data

In the simulation, the controlled environment with clean-depth images led to near-perfect model performance, with kNN and Neural Networks achieving an AUC of 1.000. However, real-world data from the RealSense camera introduced noise from lighting, reflections, and depth variations, reducing the model's accuracy. This discrepancy underscores the challenge of adapting models trained in idealized conditions to real-world scenarios, where sensor limitations and environmental factors impact classification performance.

B. Laboratory Environment Limitations

Unlike those of a real power transmission line, the laboratory's spatial and lighting constraints led to considerable noise in the images captured by the RealSense camera, complicating object identification. Additionally, the camera's position at the bottom of the robot, capturing data as if the transmission line were upside down, introduced further discrepancies that would not occur in a real-world inspection, potentially affecting model performance.

C. Model Performance

While simple and interpretable, the Decision Tree model became excessively large and complex in this project due to the variations in simulated and real-world depth images. It generated an impractical structure, losing its main advantage of clear decision rules, especially when exposed to noise in real-world data.

The k-Nearest Neighbors (kNN) model showed strong consistency in both simulated and real-world data, with perfect AUC (1.000) and a slight drop to 0.991 and 0.948 for real-world data. kNN effectively handled noise, especially in derived images, thanks to the Mahalanobis distance metric.

The Neural Network excelled in simulated data with an AUC of 1.000 but also performed well on real-world data (AUC of 0.996) despite noise. However, its higher computational cost compared to kNN could be a limitation for embedded systems.

AdaBoost performed well on simulated data (AUC of 0.996 and 0.999 for raw and derived images) but struggled with real-world data, with AUCs of 0.825 and 0.875. The model's performance was compromised by noise, leading to overfitting and reduced generalization ability.

ACKNOWLEDGEMENTS

The project is supported by the National Council for Scientific and Technological Development (CNPq) under grant number 407984/2022-4; the Fund for Scientific and Technological Development (FNDCT); the Ministry of Science, Technology and Innovations (MCTI) of Brazil; the Araucaria Foundation; the General Superintendence of Science, Technology and Higher Education (SETI); and NAPI Robotics.

REFERENCES

[1] Z. Qing *et al.*, "Mechanical design and research of a novel power lines inspection robot," in *2016 international conference on integrated circuits and microsystems (ICICM)*, IEEE, 2016, pp. 363–366.

[2] B. Tong, "Research status and development trend of obstacle crossing mechanism of hv transmission line inspection robot," *Journal of Engineering Research and Reports*, vol. 26, no. 5, pp. 83–92, 2024.

[3] X. Yue, H. Wang, and Y. Jiang, "A novel 110 kv power line inspection robot and its climbing ability analysis," *International Journal of Advanced Robotic Systems*, vol. 14, no. 3, p. 1 729 881 417 710 461, 2017.

[4] N. Pouliot, P.-L. Richard, and S. Montambault, "Linescout power line robot: Characterization of a utm-30lx lidar system for obstacle detection," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2012, pp. 4327–4334.

[5] X. Qin, G. Wu, J. Lei, F. Fan, and X. Ye, "Detecting inspection objects of power line from cable inspection robot lidar data," *Sensors*, vol. 18, no. 4, p. 1284, 2018.

[6] Y. Song *et al.*, "A vision-based method for the broken spacer detection," in *2015 IEEE international conference on cyber technology in automation, control, and intelligent systems (CYBER)*, IEEE, 2015, pp. 715–719.

[7] Y. Zhu, X. Wang, and B. Xu, "Design of vision-based obstacle crossing of high-voltage line inspection robot," in *2016 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, IEEE, 2016, pp. 506–511.

[8] A. Domingues *et al.*, "A robotic cable-gripper for reliable inspection of transmission lines," in *Robot 2023: Sixth Iberian Robotics Conference*, L. Marques, C. Santos, J. L. Lima, D. Tardioli, and M. Ferre, Eds., Cham: Springer Nature Switzerland, 2024, pp. 519–530, ISBN: 978-3-031-58676-7.