RGB-D Object Classification System for Overhead Power Line Maintenance

A Machine Learning Approach

José Mário Nishihara, Heitor Silvério Lopes, Thiago Henrique Silva,
André Eugenio Lazzaretti, Andre Schneider de Oliveira, Ronnier Frates Rohrich

Universidade Tecnológica Federal do Paraná
Curitiba, Brazil

IARIA Conference 2025

Presentation Overview

Introduction & Motivation

- The importance and challenges of high-voltage line inspection

System Architecture

- An overview of the proposed solution

Methodology

- How data was collected, processed and used

Results

- Performance comparison in simulated vs. real environments

Discussion & Conclusions

- Key takeaways and project feasibility

Introduction & Motivation

Problem Statement

High-voltage power line inspection is critical for electrical grid safety and efficiency

Limitations of Traditional Methods

Manual climbing: precise but costly and hazardous

Drone-based monitoring: agile but constrained by battery life and weather conditions

Line-traveling robots

Promising solution for safer and more efficient inspections

The Challenge of Autonomy

Robotic Automation

For a robot to be truly effective, it needs autonomy to traverse an entire power line

Challenge for autonomy

The ability to intelligently identify and classify obstacles (components) on the line

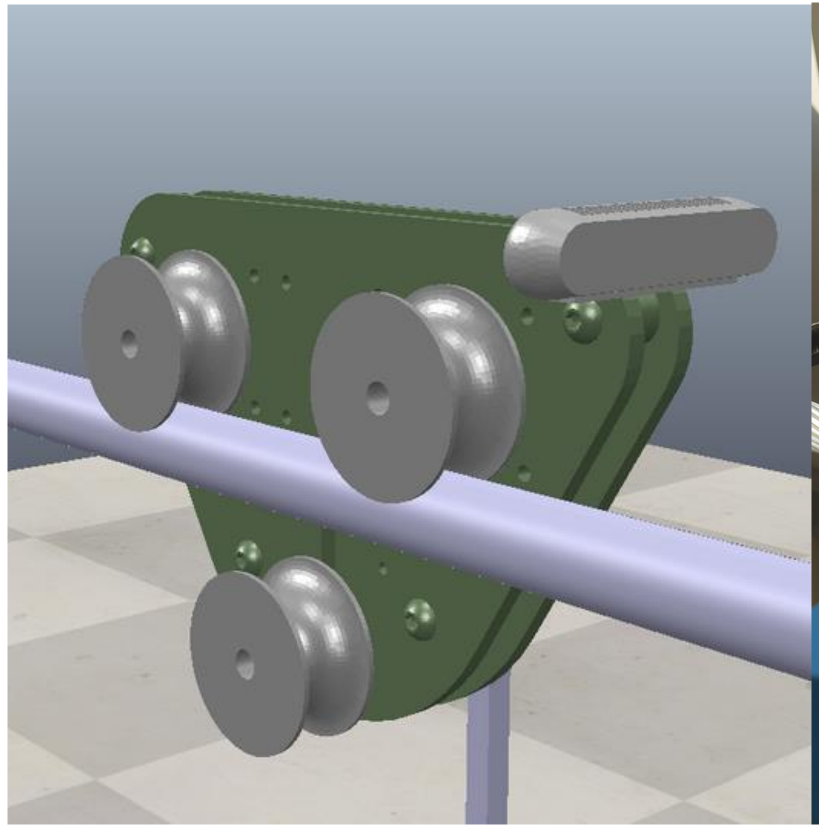
Our Goal

To develop and evaluate a real-time object classification system using depth data from a RealSense D415 camera to guide the robot's actions

System Architecture

Core Components

Inspection Robot



Simulation

Real

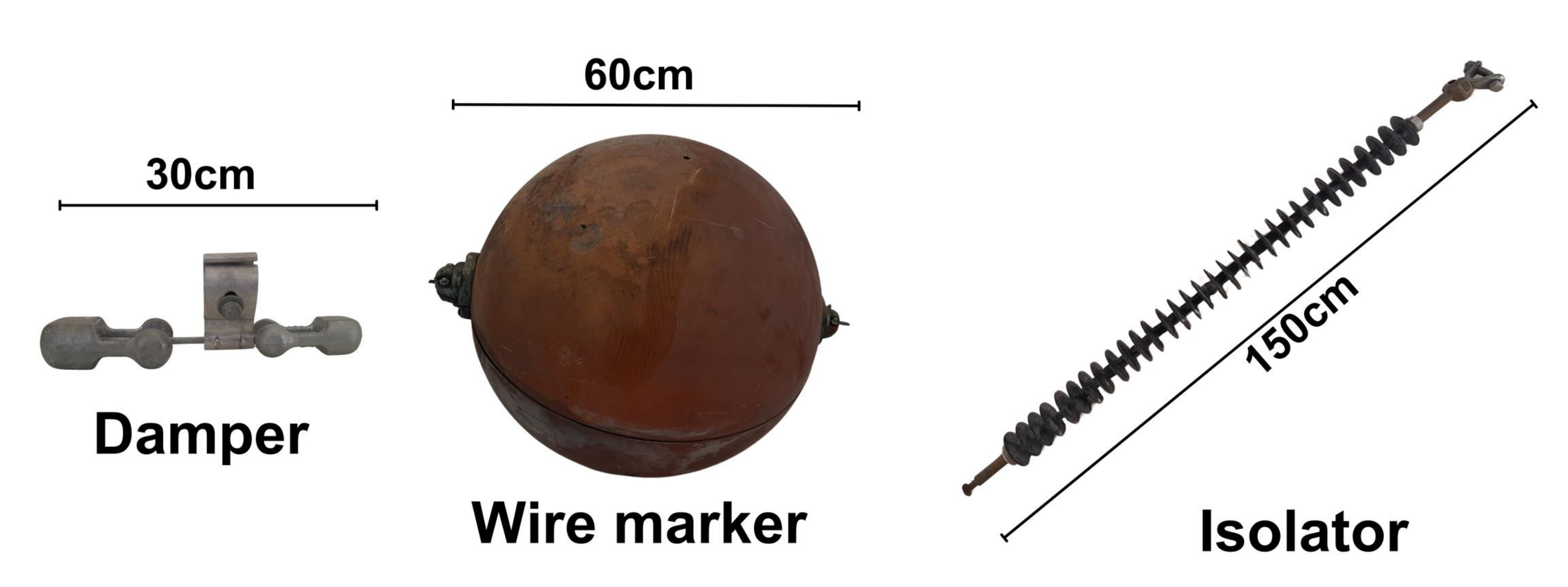
Intel RealSense D415 Camera

Specifications

- Resolution: 1280 × 720
- FOV: 65° × 40°
- Accuracy: <2% at 2m
- Frame Rate: up to 90 fps

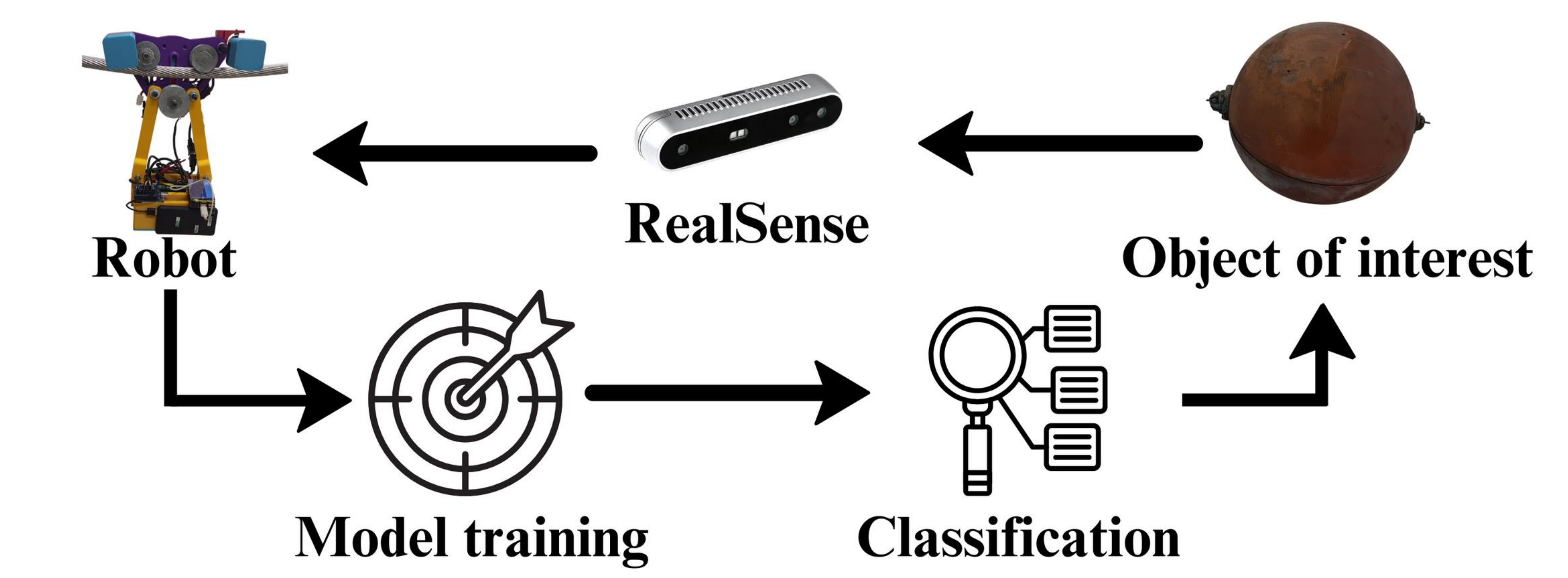
System Architecture

Objects to Be Detected



Methodology

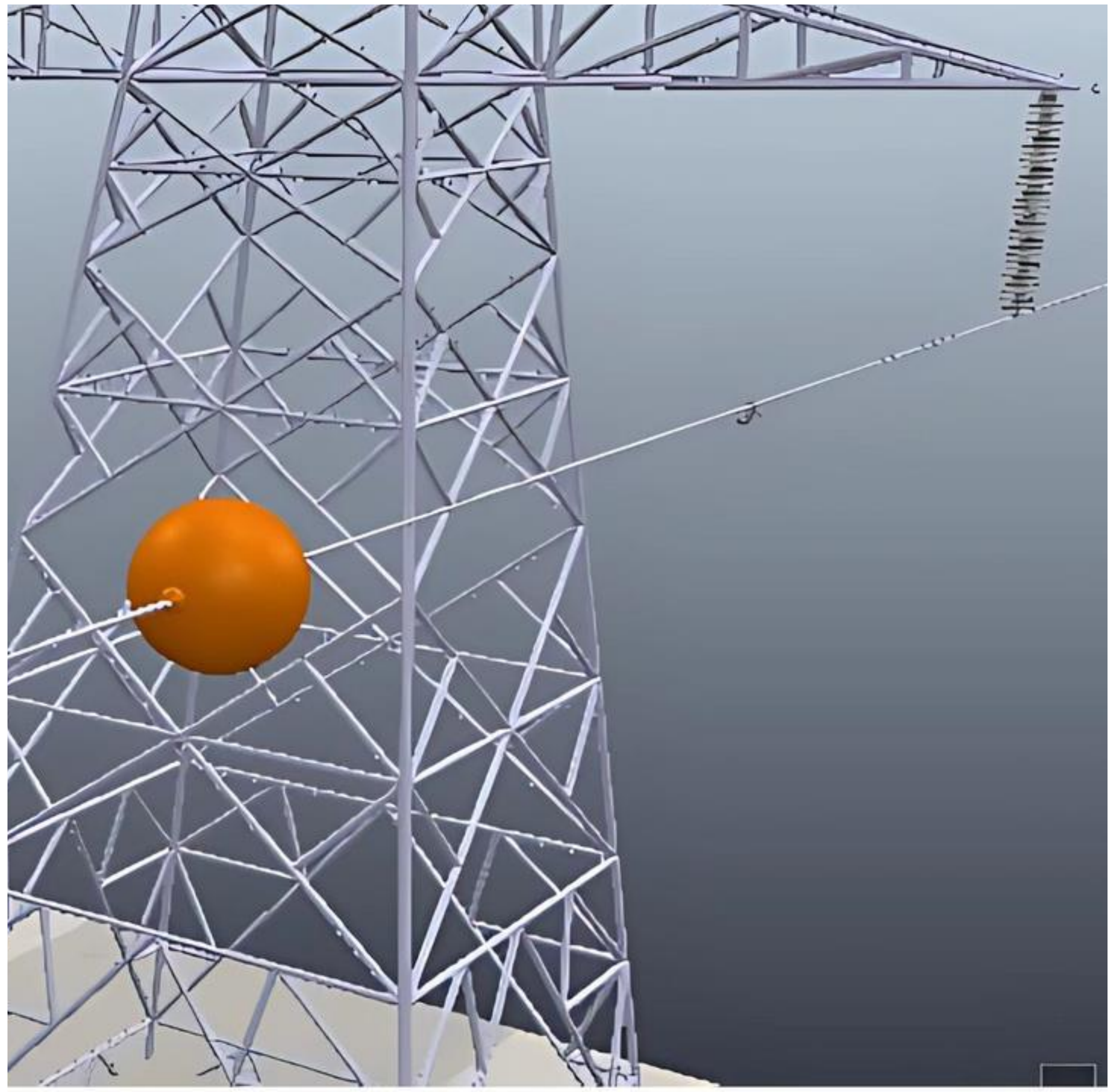
Operation diagram



Methodology

Two Data Collection Environments

Simulation

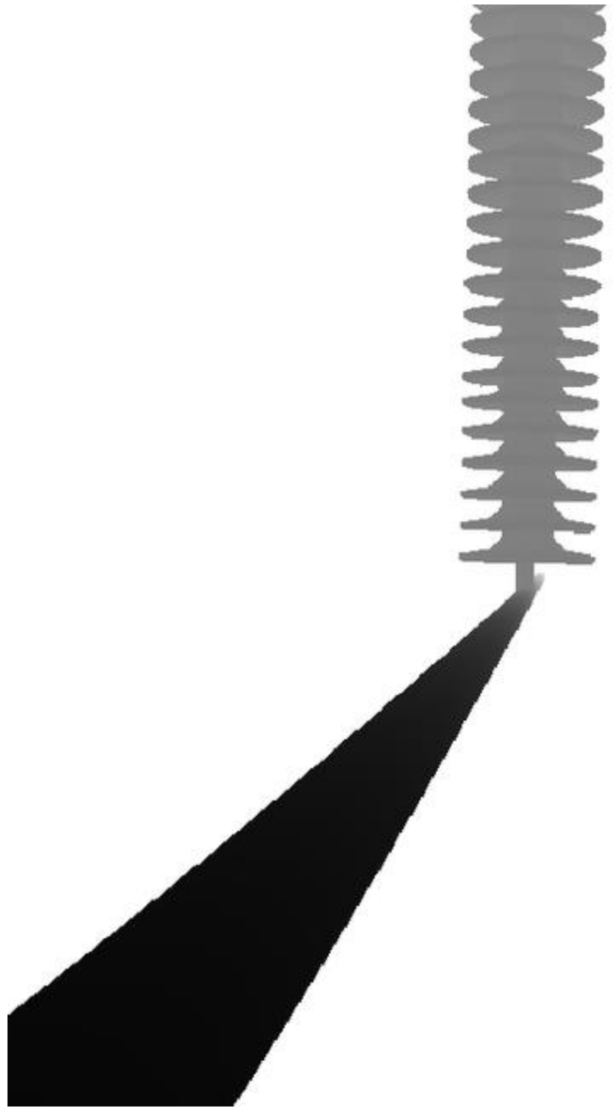


Real-System

Methodology

Depth Image Captured

Simulation



Real-System

Real-System (RGB)

Data Processing

Raw Images

The original grayscale depth images captured by the camera

Derived Images

Images processed by an edge detection algorithm

Feature Extraction

SqueezeNet: A lightweight Convolutional Neural Network (CNN) to extract complex visual features

Mean and Variance: A computationally cheaper, statistical approach to summarize depth and surface irregularities

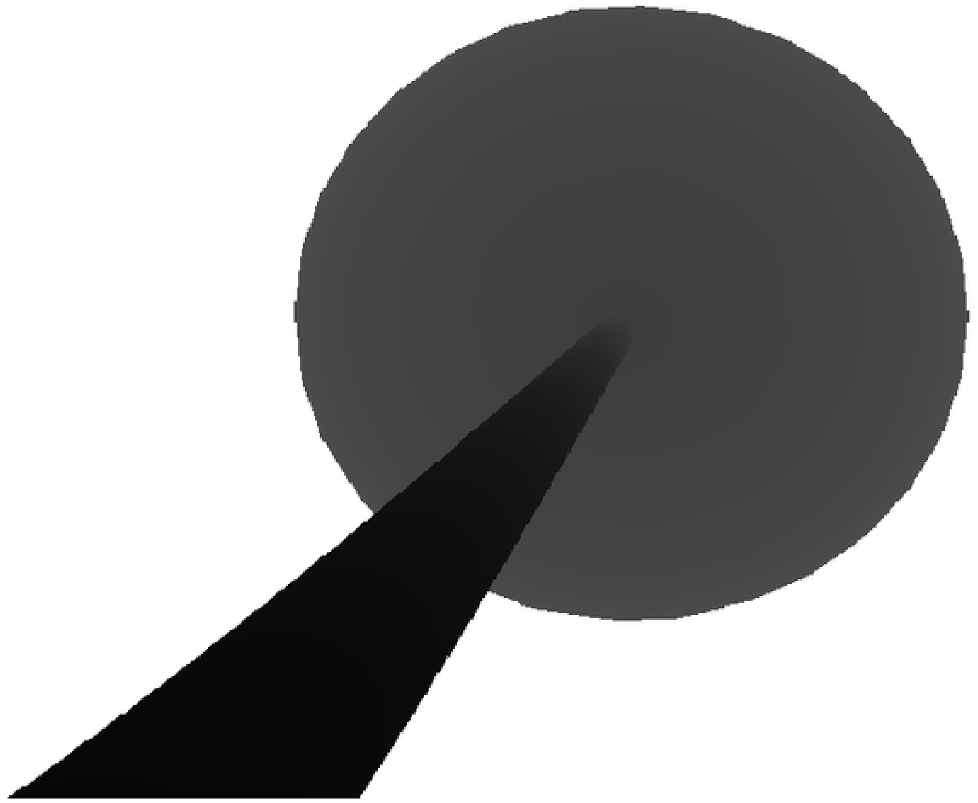
Edge Detection Algorithm

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

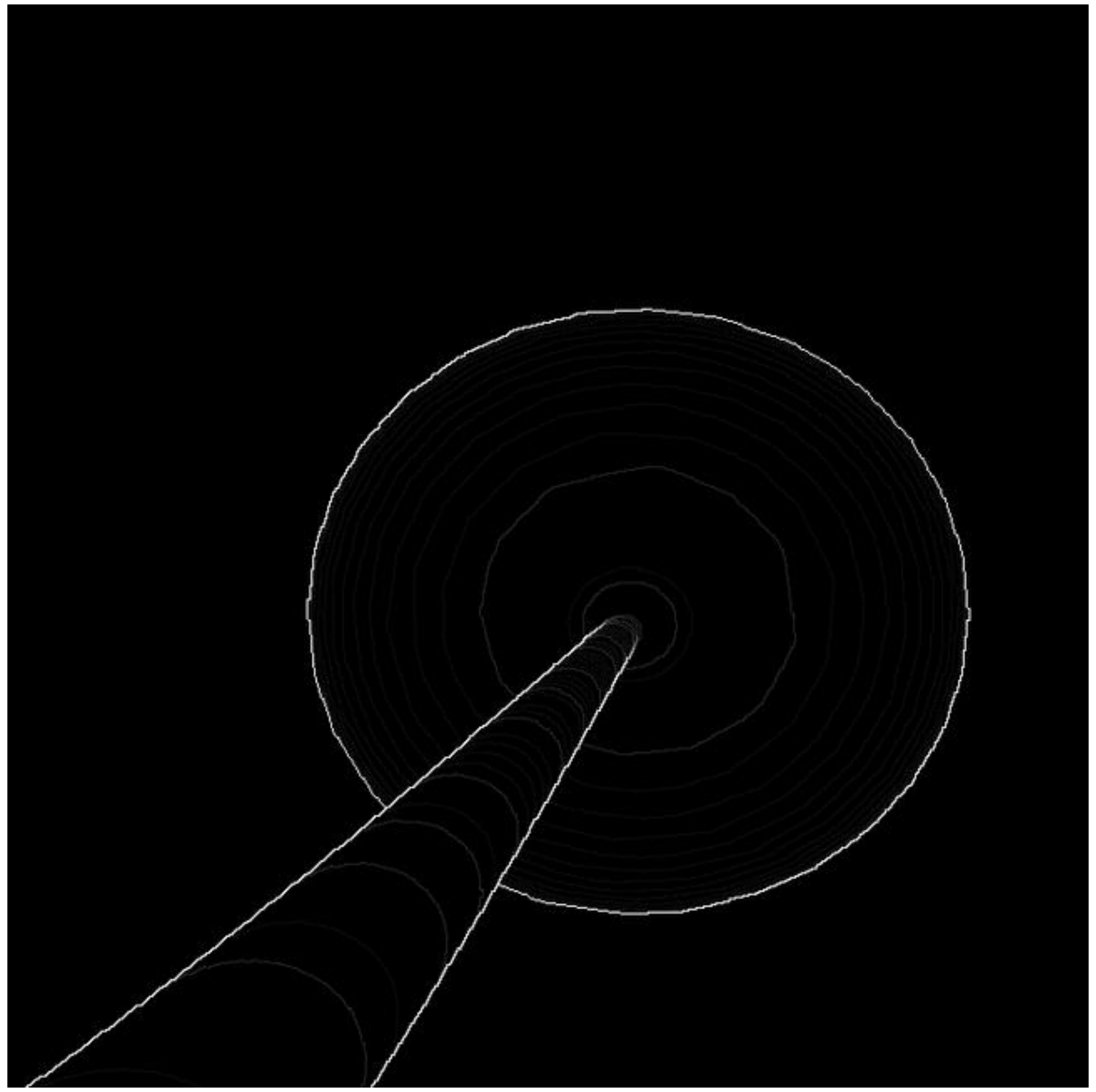
Edge Detection Algorithm

Simulated Depth Image Before and After the Algorithm

Before



After

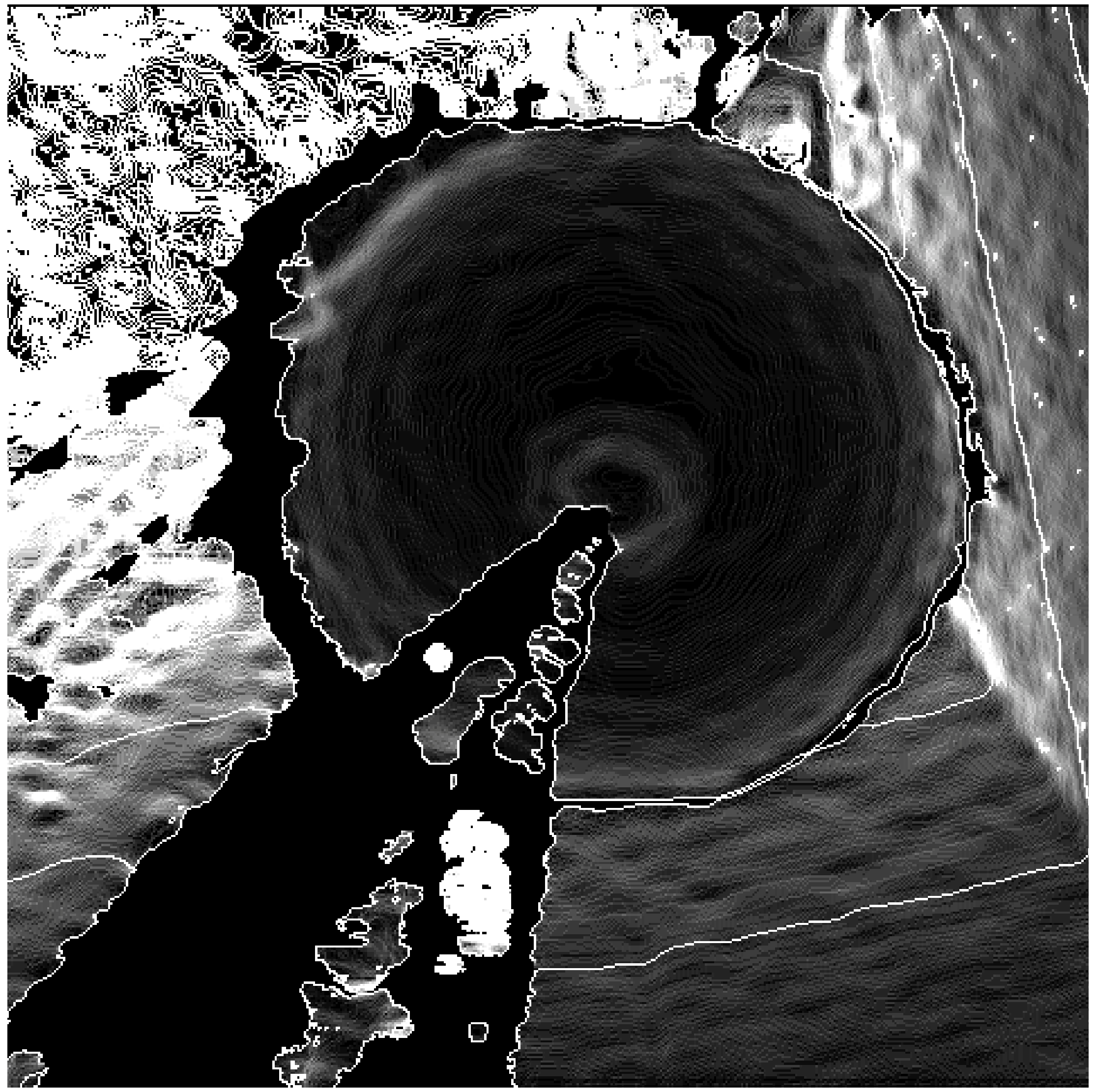


Edge Detection Algorithm

Real Depth Image Before and After the Algorithm

Before

After



Machine Learning Models

Models Evaluated

k-Nearest Neighbors (kNN)

An instance-based model that classifies based on similarity

k = 6, Mahalanobis distance, distance-based weights

Neural Network

A model capable of learning complex patterns

3 hidden layers (128, 64, 32 neurons), ReLU activation, Adam optimizer

Decision Tree

An interpretable model based on a set of decision rules

AdaBoost

An ensemble model that combines multiple weak classifiers to create a stronger one

Samme.R variant for multiclass classification

Training and validation were performed using 10-fold cross-validation

Results: Simulation Data

With Raw Images

Near-perfect performance across all models

kNN and the Neural Network achieved perfect performance

With Derived Images

Performance slightly decreased compared to using raw images

Conclusion

In an ideal, noise-free environment, raw depth data is sufficient, and pre-processing is not necessary

However, combining the fast edge detection algorithm with statistical features (mean & variance) is a computationally cheaper approach than using a CNN like SqueezeNet, making it suitable for resource-constrained embedded systems

Results: Real Data

With Raw Images

Performance dropped due to sensor noise and lab physical conditions

The Neural Network and kNN maintained the best performance, while Decision Tree and AdaBoost struggled

With Derived Images

All four models achieved perfect or near-perfect scores

Conclusion

The edge detection algorithm was extremely effective at filtering noise and highlighting essential object features, proving crucial for success in a real-world scenario

Model Performance Comparison

Models

k-Nearest Neighbors (kNN)

Proved to be robust and consistent in both simulated and real-world data. It is an excellent choice due to its strong performance and lower computational cost compared to the NN

Neural Network

Achieved the best or second-best performance in nearly all scenarios but requires more computational resources

Decision Tree

Became excessively large and complex, losing its main advantage of interpretability, making it impractical for this application

AdaBoost

Performed well on simulated data but its performance was compromised by noise in the real-world data

Discussion & Conclusions

Feasibility

This study demonstrates the feasibility of using depth cameras for the autonomous classification of objects on transmission lines

The Importance of Pre-processing

The key takeaway is that a simple, efficient image pre-processing step (edge detection) was more impactful for success than the choice of ML model, when dealing with noisy, real-world data

Best Solution

The combination of the RealSense camera, the edge detection algorithm, and a robust classifier like kNN represents an effective, accurate, and computationally viable solution for embedded systems on inspection robots

Limitations & Future Work

Validation was conducted in a laboratory setting. The next step is to test the system in a real outdoor environment, overcoming challenges like natural lighting and other weather variables and also test different approaches for image pre-processing.

Acknowledgements

This project is supported by the National Council for Scientific and Technological Development (CNPq), 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

Thank you!

Contact Information

José Mário Nishihara — josalb@alunos.utfpr.edu.br
Heitor Silvério Lopes — hslopes@utfpr.edu.br
Thiago Henrique Silva — thiagoh@utfpr.edu.br
André Eugenio Lazzaretti — lazzaretti@utfpr.edu.br
Andre Schneider de Oliveira — andreoliveira@utfpr.edu.br
Ronnier Frates Rohrich — rohrich@utfpr.edu.br

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ