

The background features four decorative geometric patterns in the corners. The top-left corner has a series of parallel diagonal lines in a light blue-grey color, with a thin curved line segment extending from the top-right. The top-right corner contains a cluster of overlapping semi-circles in yellow, red, teal, and dark blue. The bottom-left corner features a similar cluster of overlapping semi-circles in red, teal, and dark blue. The bottom-right corner has a series of parallel diagonal lines in a light blue-grey color, with a thin curved line segment extending from the top-left.

PCBWIZ



ELECTRICAL COMPONENT DETECTION

USING OPENCV AND MACHINE LEARNING



TEAM RSA

SHARON P SHAJAHAN

ATHUL ANOOP

ROSHIN R

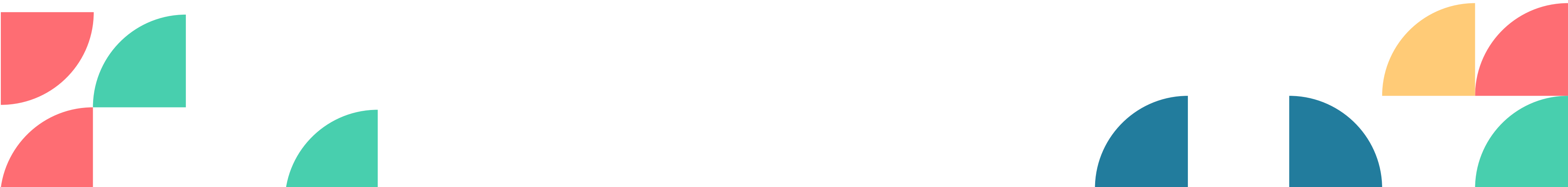


The background features four decorative geometric patterns in the corners. The top-left corner has a series of parallel diagonal lines. The top-right corner contains a cluster of overlapping semi-circles in yellow, red, teal, and blue. The bottom-left corner also features a cluster of overlapping semi-circles in red, teal, blue, and red. The bottom-right corner has a large, faint semi-circle outline with several parallel diagonal lines inside it.

PROJECT INTRODUCTION

Welcome to the "Electrical Component Detection" project presentation. In this project, we explore the application of computer vision techniques to identify and label various electrical components in real-time using OpenCV and ML.

This project is to showcase our approach in leveraging OpenCV, a popular computer vision library, to analyze webcam frames and identify electrical components through contour analysis and ML. We will delve into the methodology, technologies used, and provide a demonstration of the project's capabilities.



OBJECTIVE

The identification of electrical components is crucial in various fields, including electronics maintenance, robotics, and automation. Rapid and accurate recognition of components such as capacitors, LEDs, chips, resistors, and transistors etc. can streamline troubleshooting processes and enhance the efficiency of electronic systems.



Technologies Used

- OpenCV for image processing
- Python programming language
- Webcam for capturing frames

1. Capturing Webcam Frames

The project starts by opening a connection to the webcam using OpenCV.

The `cv2.VideoCapture` function is used to access the default camera (represented by 0).

2. Image Processing

Each frame captured from the webcam is converted to grayscale to simplify processing.

Gaussian blur is applied to reduce noise in the image, improving the accuracy of subsequent operations.

Adaptive thresholding is employed to convert the blurred image into a binary image. Adaptive thresholding is particularly useful in handling varying lighting conditions.

3. Edge Detection

The Canny edge detector is used to identify edges in the binary image. This step helps in highlighting the boundaries of objects within the image.

4. Contour Detection

Contours are identified in the edge-detected image using the `cv2.findContours` function.

Contours represent the boundaries of connected components in the image.

5. Component Identification

The project defines a function (`get_component_label`) to analyze the contours and determine the type of electrical component based on its aspect ratio.

The aspect ratio is calculated from the bounding box of each contour.

6. Displaying Results

For contours with an area greater than a specified threshold, a bounding box is drawn around the identified electrical component on the original frame.

The component label is displayed near the bounding box.

7. Real-time Presentation

The processed frames are continuously displayed in real-time using OpenCV.

The user can see the webcam feed with identified electrical components and their labels.

The background features four decorative geometric patterns in the corners. The top-left corner has a series of parallel diagonal lines in a light blue-grey color. The top-right corner contains a cluster of overlapping semi-circles in yellow, red, teal, and dark blue. The bottom-left corner also features a cluster of overlapping semi-circles in red, teal, and dark blue. The bottom-right corner has a series of parallel diagonal lines in a light blue-grey color, mirroring the top-left pattern.

THANK YOU