

٢٠٢٥

Bar-Code Detection and Decoding

BAR-CODE DETECTION AND DECODING USING PURE
PYTHON REPORT
HAMZA MOHAMMED

Name: Hamza Obaid
ID: 120211251
Subject: Digital Image processing Project

Contains

Introduction	2
Methodology	2
Image Preprocessing	2
Implementation.....	2
Programming Language and Librarie	2
Modular Design	2
Results and Testing	3
Test Cases	3
Observations	3
Limitations	4
Conclusion and Future Work	4
Conclusion	4
This project successfully demonstrates fundamental image processing techniques using Python. The combination of grayscale conversion and median filtering effectively enhances image quality.....	4
Future Improvements	4

Introduction

Barcodes are widely used for encoding information in a scannable format. This report presents an approach to detect and extract Barcodes from images using image processing techniques.

Methodology

Image Preprocessing

Conversion to Grayscale:

The system converts the input image into grayscale to simplify the processing steps.

Median Filter:

A median filter is applied to reduce noise while preserving image details.

Implementation

Programming Language and Librarie

The project is implemented in Python using:

- PIL (Pillow): For image handling and manipulation.
- NumPy for numerical operations.
- Matplotlib for image visualization.

Modular Design

The code is structured into functions for each major step:

- `load_and_display_image()`: Loads the input image and displays it.

- `convert_to_grayscale()`: Converts the image to grayscale.
- `median_filter ()`: Applies a median filter to reduce noise.
- `Threshold ()`: applies Otsu's thresholding method to convert it into a binary image.
- `find_barcode_region ()`: Extracts the detected Barcode region from the image.
- `detect_barcode ()`: Orchestrates the full Barcode detection and extraction process.

Results and Testing

Test Cases

The implementation was tested on multiple images using the following workflow:

1. Load and display the image.
2. Convert the image to grayscale.
3. Apply the median filter.
4. Detect Bar codes.
5. Extract and display the detected Bar code region.

Observations

The detection function successfully identified and decoded Bar codes from test images.

Limitations

- The performance may vary depending on the type and quality of images.
- Some high-noise images may still retain artifacts after filtering.

Conclusion and Future Work

Conclusion

This project successfully demonstrates fundamental image processing techniques using Python. The combination of grayscale conversion and median filtering effectively enhances image quality.

Future Improvements

Potential areas for enhancement include:

- Implementing additional filtering techniques for better noise reduction.
- Enhancing robustness against varying image conditions.
- Exploring advanced edge detection and feature extraction methods.