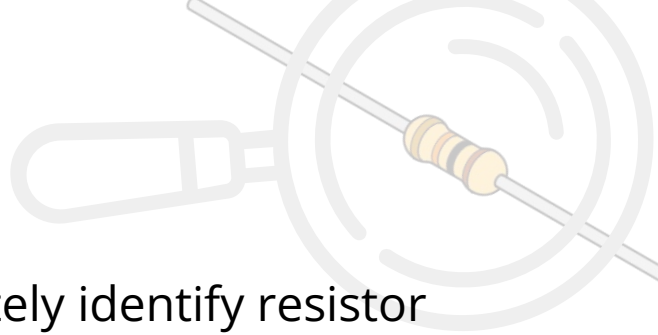
Resistor Recognition

By:
Eitan Klass,
Roey Mevorach,
Brandon Pragasa
and
Nathan Jaggers

Introduction

- Electronic circuits are an integral part of everyday life, and resistors are one of the key components in such circuits
- With the use of computer vision technology, it can be leveraged to automate the resistor value detection process and enable us to be more efficient

The Problem and Motivation



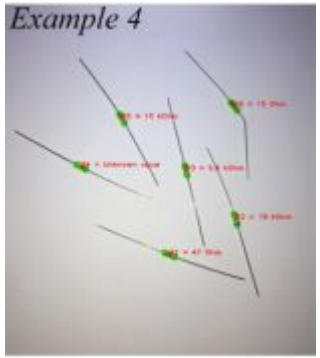
- As EEs and CPEs, we have felt the pain of accurately identify resistor values
- Traditional methods are time consuming and prone to human error
- As circuits become more complex, the demand for higher precision and accuracy increases

Data Set

- Resistors with different band colors in different order
- Resistors with same orientation
  - Evolve to different orientations
- Resistors at different distances
- Resistor environment
  - High contrast background
  - Evolve to more complicated backgrounds
- Resistors with four bands
  - Evolve to more bands

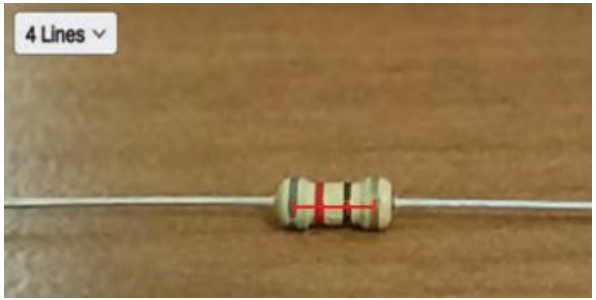
Previous Work

- Segmentation and classification of resistors from an image using openCV [1]
- Using industrial black and white cameras with a neural network to determine resistor nominal value [2]
- Real-time video analysis resistor band decoder android app [3]



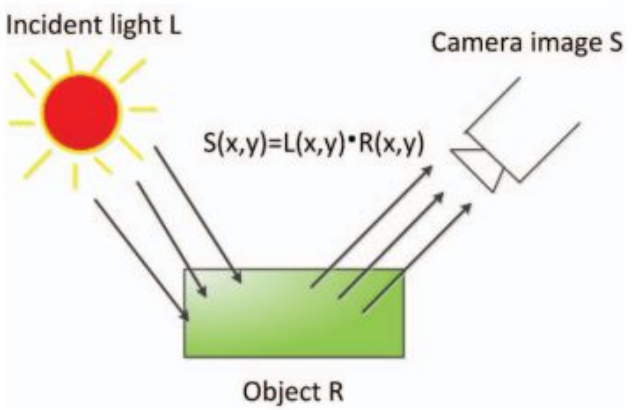
Results from [1] on left

Config from [3] on right

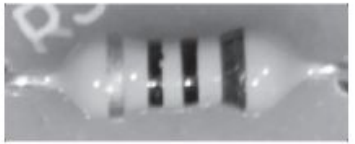


Techniques Used

- Image segmentation, masking, thresholding, Support Vector Machines (SVM) [1]
- Image preprocessing, Retinex algorithm, Back Propagation Neural Network Classifier [2]
- Nonlinear Image filtering, Photometric Invariants (RGB->HSV), Euclidean distance based clustering strategy



(a) The original image

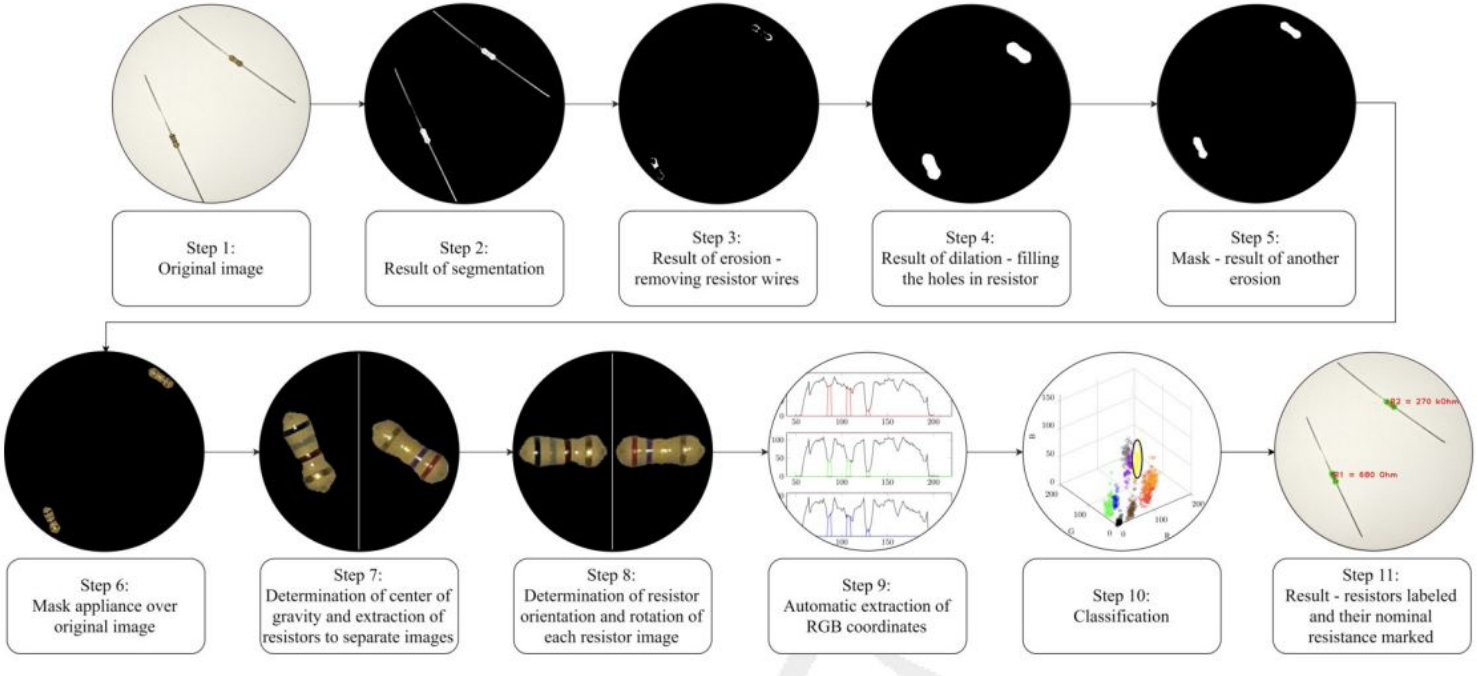


(b) The Retinex processed image

Lessons Learned

- Reduce illumination variance by either using normalized RGB or HSV
  - HSV seems to be a popular color spaces in the papers explored
- Avoid statistical analysis for location of resistor bands
  - Use edge detection for band position estimation
  - Develop other band detector to know location of band prior to image preprocessing

Algorithm from Previous Work [1]



Implementation

Step 1: Image Segmentation

- Find and segment the resistor from the image
- Thresholding v. Edge-based v. Region-based
  - Depends on complexity of the image
- Functions:
  - Bwareopen, imclose, imfill, bwpropfilt

High Contrast Background

original image

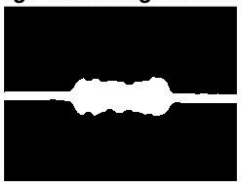
edge-based segmentation

thresholding segmenation

Noisy/Complex Background

original image

edge-based segmentation



thresholding segmenation

Implementation

Step 2: Morphological Operations

- Erosion
  - Morphological erosion removes floating pixels and thin lines so that only substantive objects remain. Remaining lines appear thinner and shapes appear smaller.
  - Use to remove wires
- Dilation
  - Morphological dilation makes objects more visible and fills in small holes in objects. Lines appear thicker, and filled shapes appear larger.
  - Use to fill in any holes in resistor

Implementation

Step 3: Object Representation and Description

- Masking the resistor body out of the original image

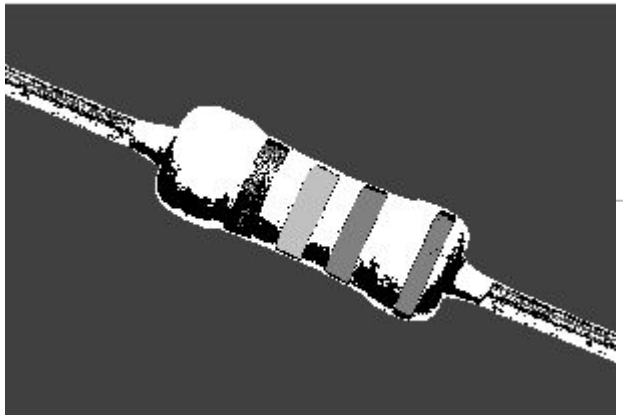
Step 4: Resistor Band Location & Color Extraction

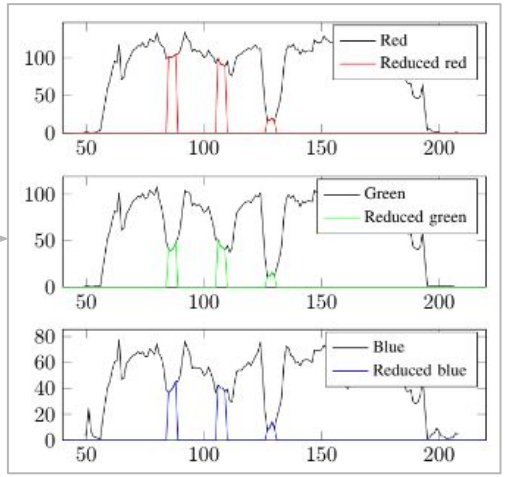
- Left End coordinates
- Right End coordinates
- Band coordinates
- Band Feature Vector [Hue, X, Y]
- Ends Feature Vector [-, X, Y]
- Resistor Feature Array:
  - Band Feature Vector [Hue, X, Y]
  - Ends Feature Vector [-, X, Y]

Band # / End	Left End	1st	2nd	3rd	4th	Right End
Hue	-					-
X pos	x	x	x	x	x	x
Y Pos	y	y	y	y	y	y

Implementation

Step 4: Resistor Band Location & Color Extraction

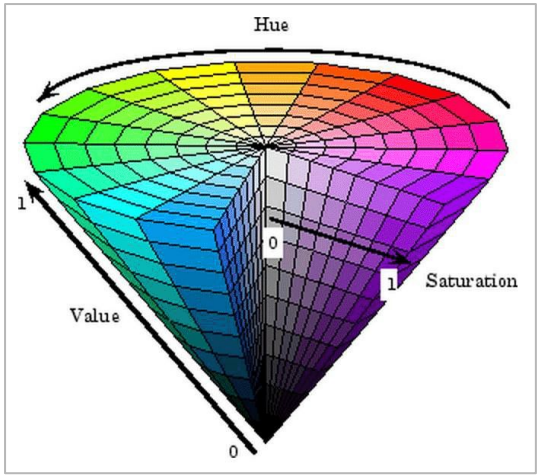




Implementation

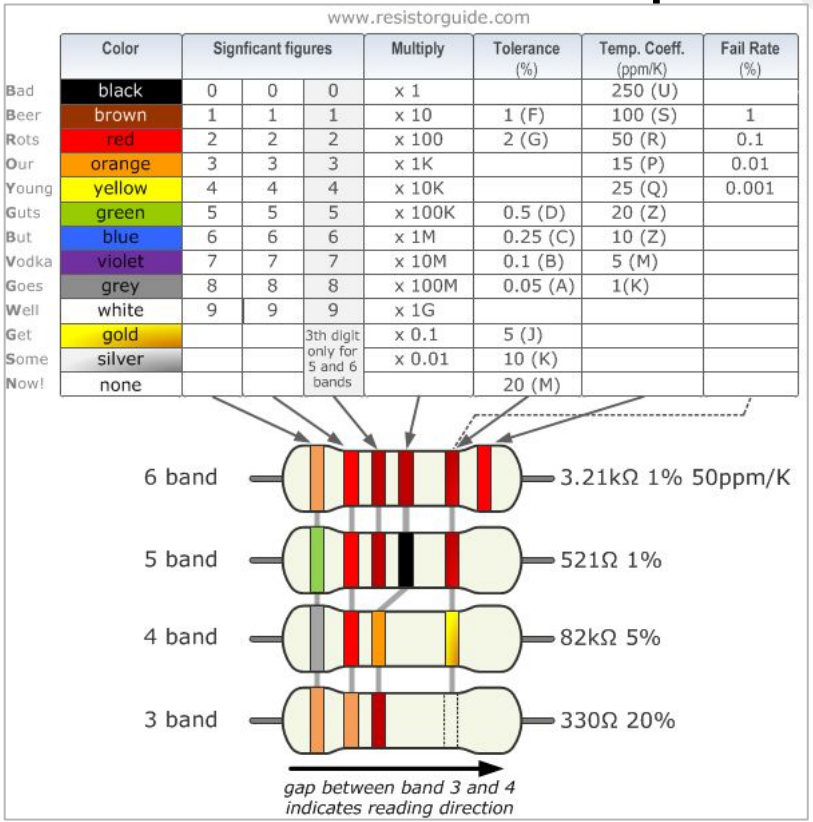
Step 4: Resistor Band Location & Color Extraction

- HSV color space
- Histogram for value in the HSV color space (Value vs. X coordinate)



Band # / End	Left End	1st	2nd	3rd	4th	Right End
Hue	-	0.04	0.15	0.03	0.06	-
X pos	x	x	x	x	x	x
Y Pos	y	y	y	y	y	y

Resistor Color Guide For Classification



Implementation

Step 5: Classification

- Use a data set
  - 12 classifiers (12 for band colors)
- Specify Hue range for each index

Band # / End	Left End	1st	2nd	3rd	4th	Right End
Hue	-	2	3	1	7	-
X pos	x	x	x	x	x	x
Y Pos	y	y	y	y	y	y

Color	Black	Brown	Red	Orange	Yellow	Green	Blue	Violet	Grey	White	Gold	Silver
Index	0	1	2	3	4	5	6	7	8	9	10	11

Implementation

Step 5: Classification

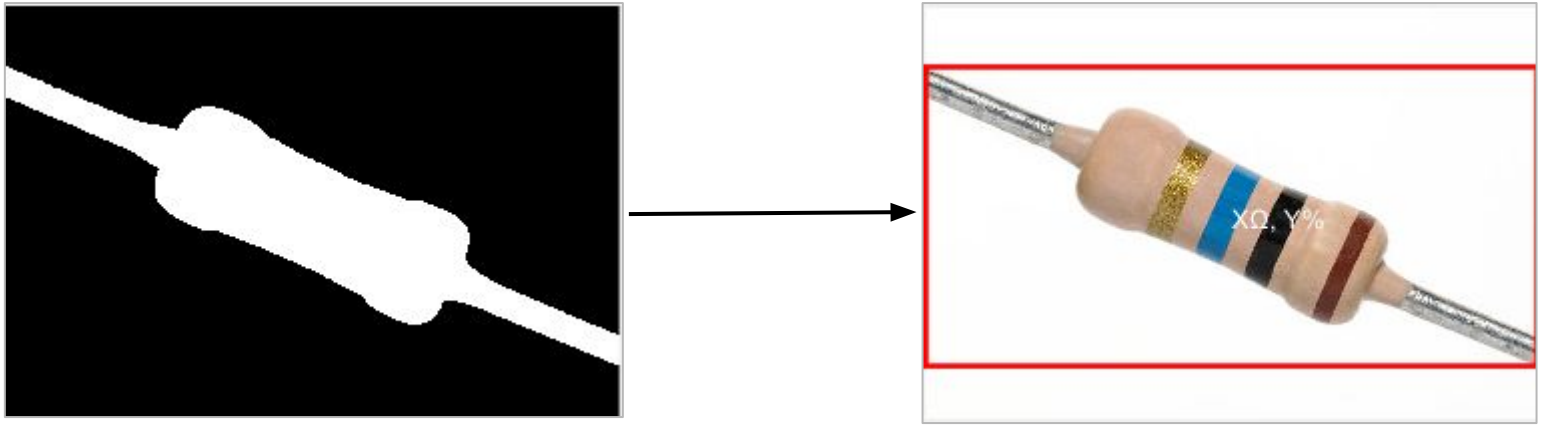
Index	0	1	2	3	4	5	6	7	8	9
Multiplier	x1	x10	x100	x1K	x10K	x100K	x1M	x10M	x100M	x1G

Index	0	1	2	3	4	5	6	7	8	9
Tolerance (%)	-	1	2	-	-	0.5	0.25	0.1	0.05	-

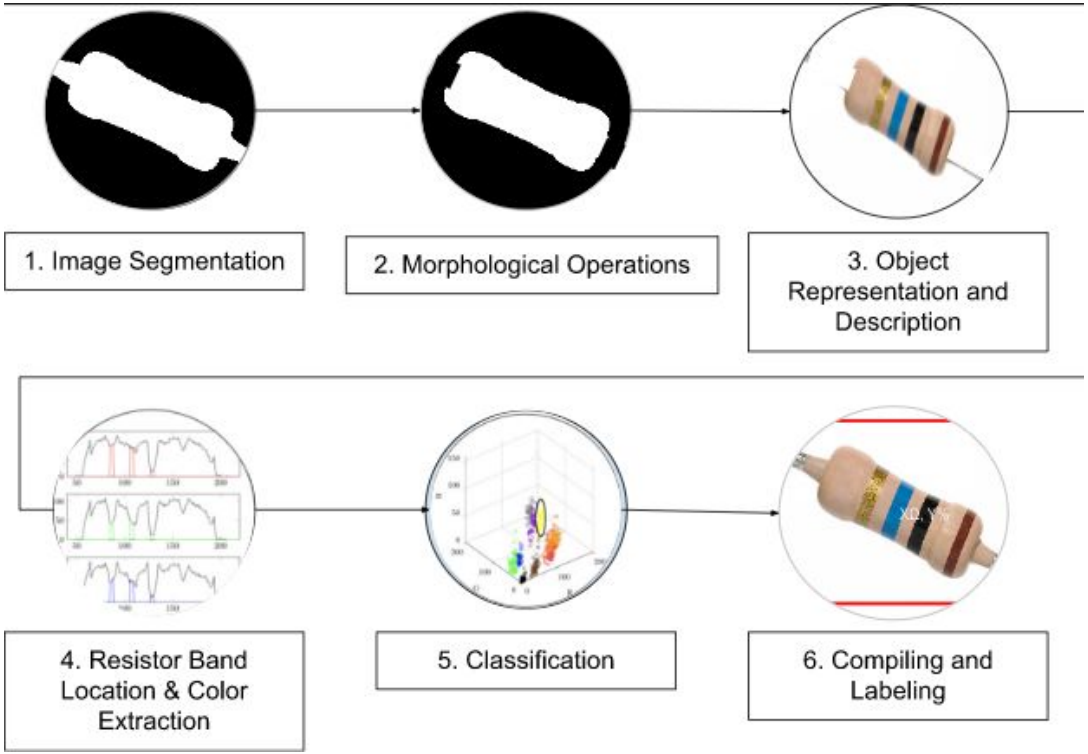
Implementation

Step 6: Compiling and Labeling

- Binary Image
- Regionprops(, ‘BoundingBox’)
- Label with the resistance value (Ω) and tolerance (%)

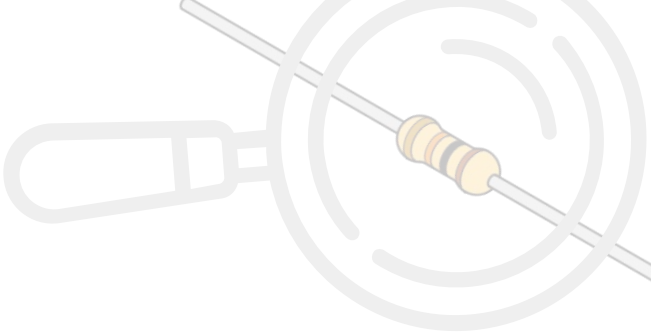


Implementation



Thank You!

Questions?



References

[1]M. Muminovic and E. Sokic, “Automatic Segmentation and Classification of Resistors in Digital Images,” *2019 XXVII International Conference on Information, Communication and Automation Technologies (ICAT)*, Oct. 2019, doi: https://doi.org/10.1109/icat47117.2019.8939034.

[2]X. Li, Z. Zeng, M. Chen, and S. Che, “A new method of resistor’s color rings detection based on machine vision,” *IEEE Xplore*, Oct. 01, 2017. https://ieeexplore.ieee.org/document/8242770/ (accessed Mar. 16, 2023).

[3]M. F. Demir, A. Cankirli, B. Karabatak, A. Yavariabdi, E. Mendi, and H. Kusetogullari, “Real-Time Resistor Color Code Recognition using Image Processing in Mobile Devices,” *2018 International Conference on Intelligent Systems (IS)*, Sep. 2018, doi: https://doi.org/10.1109/is.2018.8710533.