Real-time Traffic Signs Detection

- Introduction
- Solution
- Results and Discussion
- Improvement

•Introduction

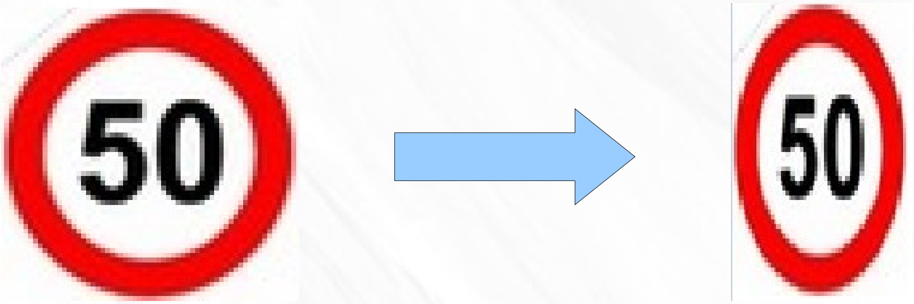
- Traffic signs
- Traffic signs in the real-world

70 m
10 m
2,30 m
3,50 m
30
40
50
60
70
80
6,5 t
www.shutterstock.com · 1327359

•Introduction

- The potential problems:
  - Noisy background.
  - Shape change when observing from side,

    like circle → ellipse

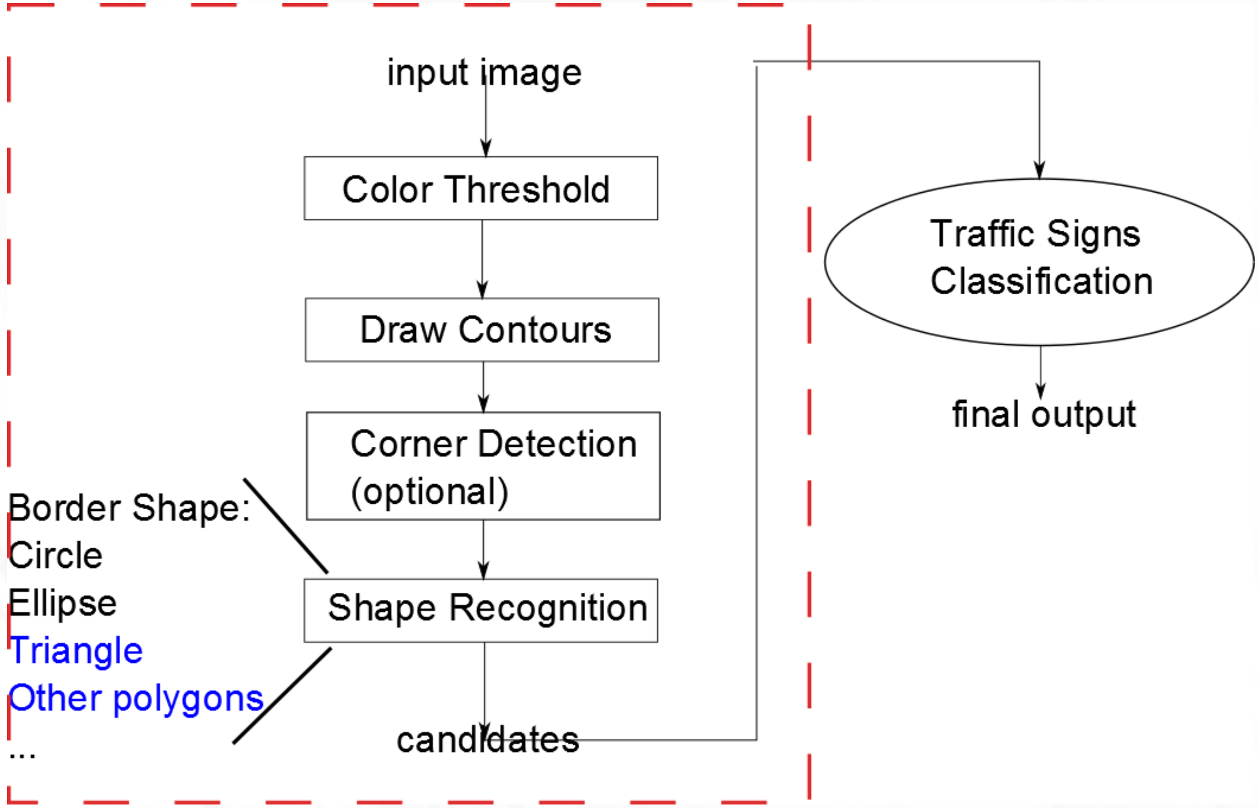


  - Low cost, so it could be deployed in real-time systems.

•Solution

- Need 4 steps to compute result.
  - Color Threshold
  - Draw Contours
  - Corner Detection(optional)
  - Shape Recognition
    - Circle
    - Ellipse
    - Triangle
    - Other polygons ...

•Solution



input image

Color Threshold

Draw Contours

Corner Detection (optional)

Shape Recognition

candidates

Traffic Signs Classification

final output

Border Shape:
Circle
Ellipse
Triangle
Other polygons
...

•Color Threshold

- The common characteristics for most traffic signs
  - Conspicuous and constant color, like red, yellow, orange...

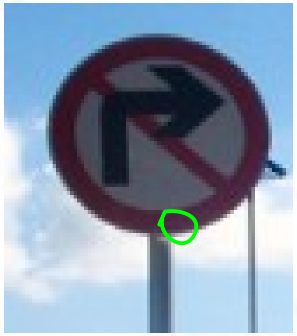
  - Regular shape and wide border

•Color Threshold

- Two techniques to separate traffic signs from other irrelevant signals
  - Color threshold
  - Shape detection
- In this project, the color threshold will be applied first
  - Reason: it could filter more signals from raw image.

•Color Threshold

- Problem: RGB metric system could be easily impacted by amphibians light change, like shadow or sunshine



Sample a point of border

RGB = [49, 22, 37]

covered by shadow

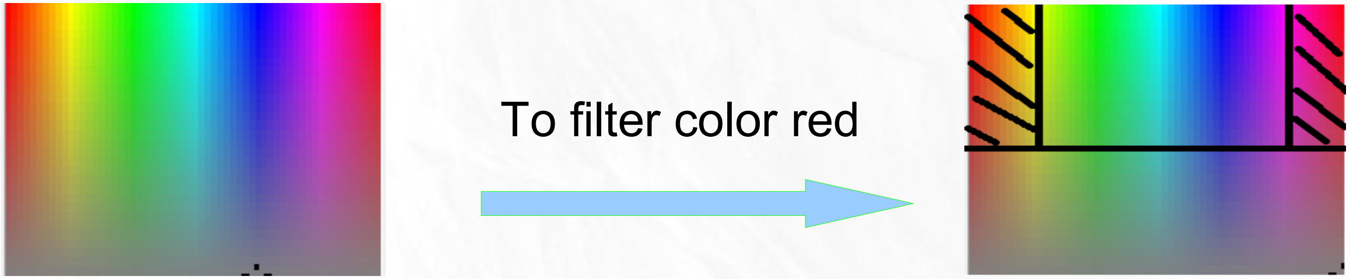
- To solve it, convert it to HSV or HSL metric.

•HSV/HSL metric

- Composed by 3 elements, fewer impact from light change than RGB.
  - Hue
  - Saturation
  - Lightness
- How to set the color threshold in HSV/HSL?

•HSV/HSL metric

- Color spectrum

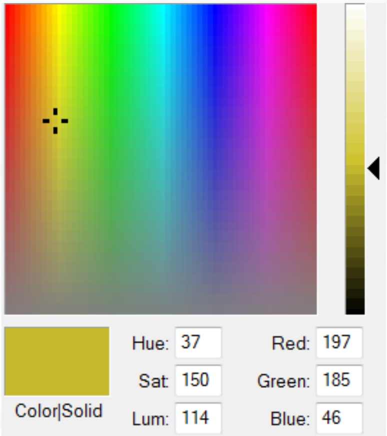


To filter color red

- What's the specific value of this range in HSV/HSL metric? Use some color tools:
- Range:

[H,S,V] = [0~20, 50~255, 50~255] and

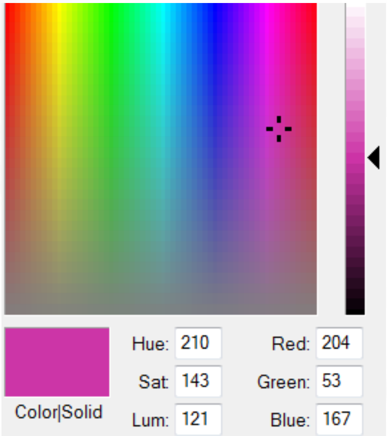
[150~179, 50~255, 50~255]



Color|Solid

Hue: 37 Sat: 150 Lum: 114

Red: 197 Green: 185 Blue: 46



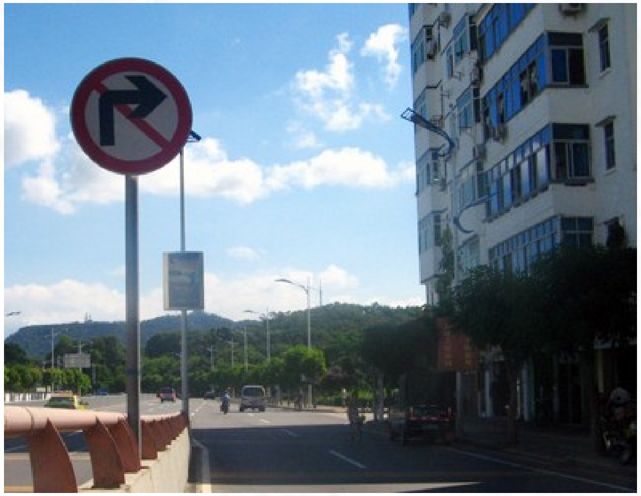
Color|Solid

Hue: 210 Sat: 143 Lum: 121

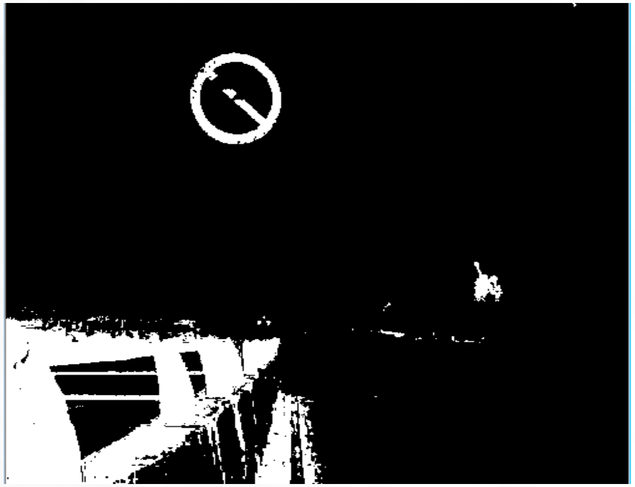
Red: 204 Green: 53 Blue: 167

•Color Threshold

- Demonstration:



Color threshold

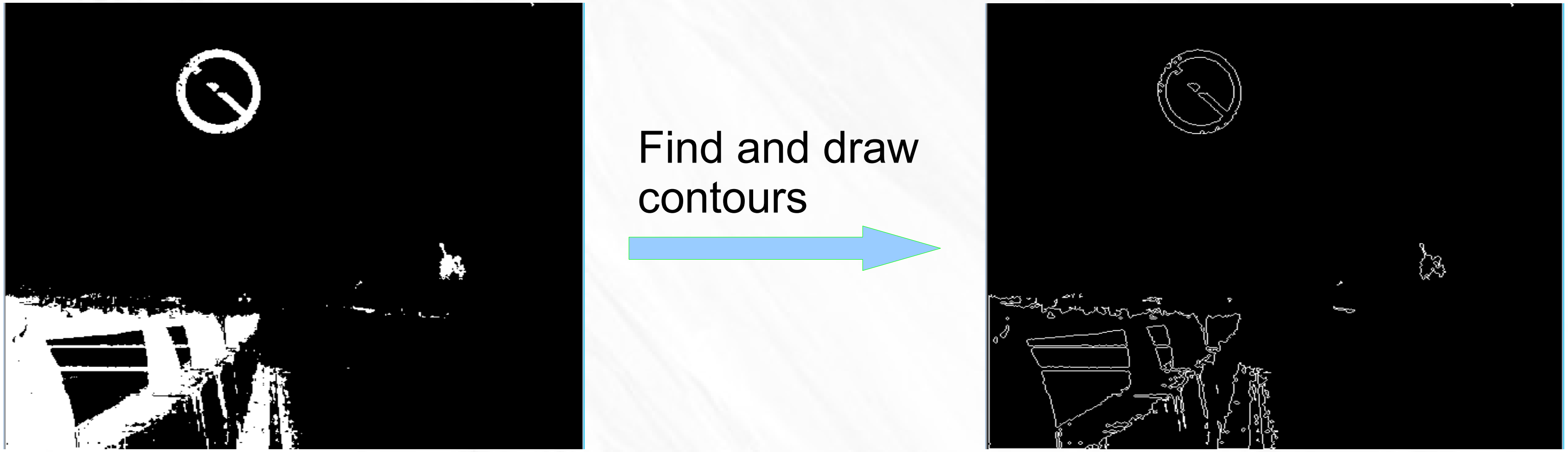


raw image

output

Draw Contours

- Motivation: we can not deploy shape recognition based on separated points.
- Demonstration:



Find and draw contours

- Output: a set of objects, and the object is composed by several points

Shape Detection

- Sorts of shape detections
  - Circle detection
  - Ellipse detection
  - Triangle detection

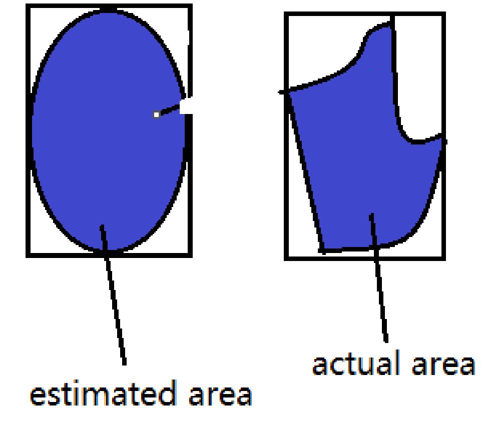
    …
- In this case, we will use only circle and ellipse, and circle could be treated as special case of ellipse, so we will deploy Ellipse detection.

Ellipse Detection

- 3 Techniques to fit ellipse:
  - Compare actual area, to estimated area
  - Actual area(aa) could b

    computed based on

    contour



  - Estimated area(ea):

    ea = PI * rect.width * rect.height
  - If |aa- ea| > error_threshod

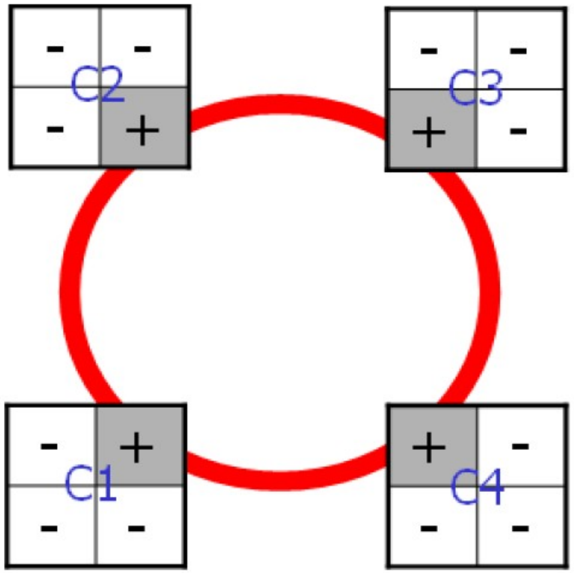
    we say this shape is not a ellipse.

Ellipse Detection

- 2nd technique: Ratio of width/height of the contour
- Shortcoming of previous 2 methods:
  - Not always reliable
  - Inaccurate
- Advantages for these 2 methods:
  - Fast, few computation is required
  - Easy to deploy.

Ellipse Detection

- 3rd technique: corner detection.
- Principle: design 4 different masks to match 4 corners of the object.

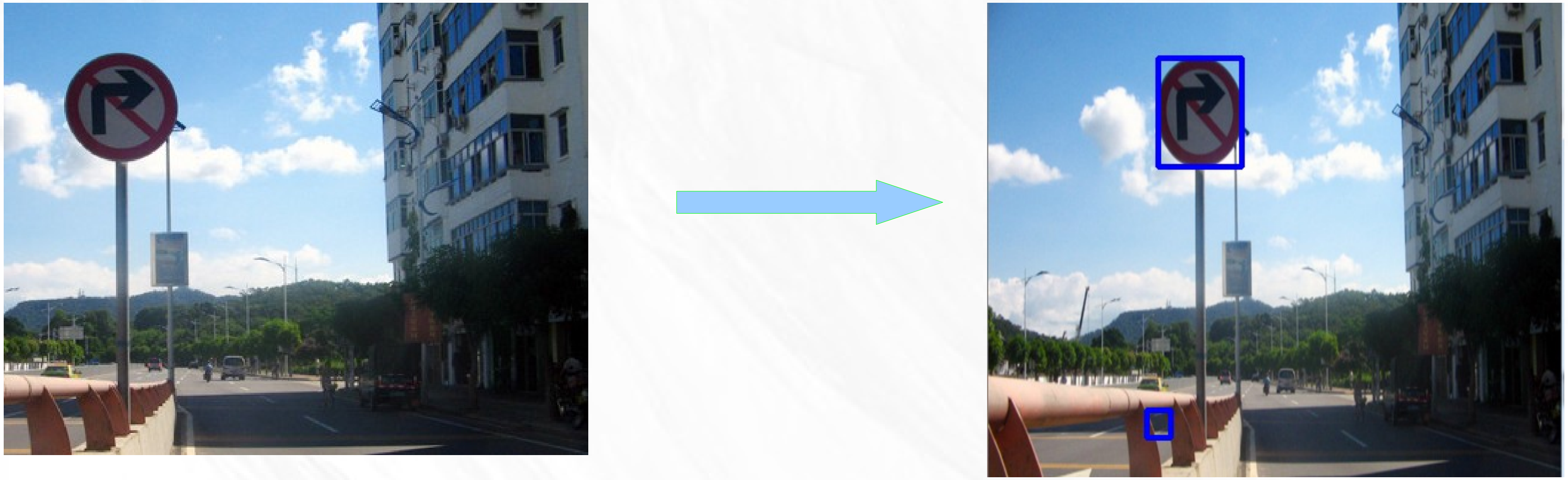


- Advantages:
  - Accurate
  - Robust
- Drawback:
  - More calculation required

    than previous 2 method, need to be adapted before it is deployed in real-time system.

Result

- Demonstration



- Incorrect detection could hardly be prevented if based on current algorithm

Improvement

- Run shape detection before color threshold to prevent rough contour edges
- Adapting the algorithm of ellipse detection, one available solution is check if the object is symmetric
- Improving color threshold algorithm to prevent isolated points.

- Thank you!