Lane Detection for Intelligent Driver Assistance System

Group #22
Yawen Deng, Yuzhe Sheng, Min Zheng
NetID: yd298, ys766, mz474

Dec.5th 2017

Problem Statement

- Based on single image
- For the center lane: Segment lane mark pixels and extract single-pixel-width line (separately for left and right)

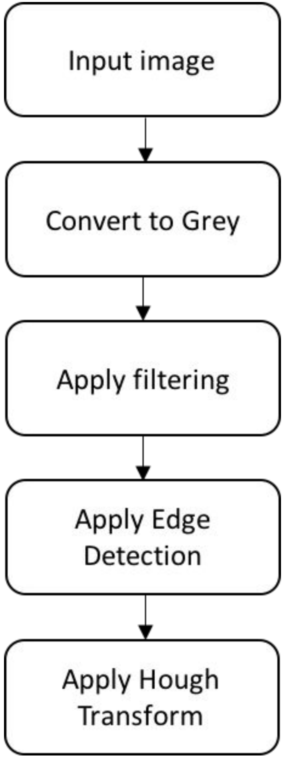
- ❑ Input Image
- ❑ output Image

Motivation and Issues

- **Significance:**
  - 1.24 million people die every year from automobile accidents[1]
  - Human error is the most critical reason for 93% of crashes[1]
  - Solution: Applications using Computer vision based lane detection
  - Vision enhancement System, Lane Departure Warning in Advanced driver-assistance systems (ADAS)
  - Applied by Nissan, Toyota, Honda, BMW
- **Issues:**
  - Curved lane
  - Sensitive to noise and change of lighting (shadows, incomplete landmarks)

[1]"Global Status Report on Road Safety 2013: Supporting a Decade of Action", 2013.World Health Organization.

Previous Works Review and Issues



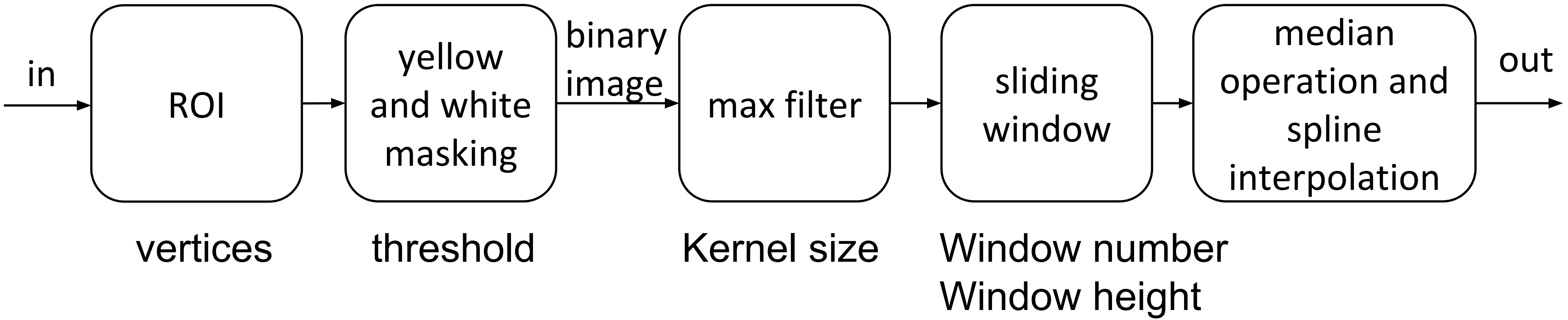
Methods	Assumption	Issue
Edge Detection + Hough Transform[1]	lane is represented by a straight line in the nearest field	can only detect straight line and nearly straight curves
B-Snake based algorithm[2]	two sides of the road boundaries are parallel. The objective is to detect the mid-line of the lane.	repeatedly quoted Hough algorithm, which increase the computation cost (0.5s/frame)
Fast Hyperbolic model based algorithm[3]	use Mid_to_Side marking two parallel lines	the use of hyperbolic fitting makes the fitted curve fluctuate largely when the road encounter more noise , success rate ~ 80%, (>1.8s/frame)

[1]J. B. McDonald. "Application of the hough transform to lane detection and following on high speed roads." Proceeding of Irish Signals and Systems Conference, pp. 340-345, 2001.
[2]YUE WANG, EAM KHWANG THOH, DINGGANG SHEN. Lane detection and tracking using B-Snake[J]. Image and Vision Computing.2004,22(4):269-280.
[3]Qiang Chen,Hong Wang. A Real-time Lane Detection Algorithm Based on a Hyperbola-Pair Model.2006,13(15): 510-515.

Assumptions, Pipeline and key parameters

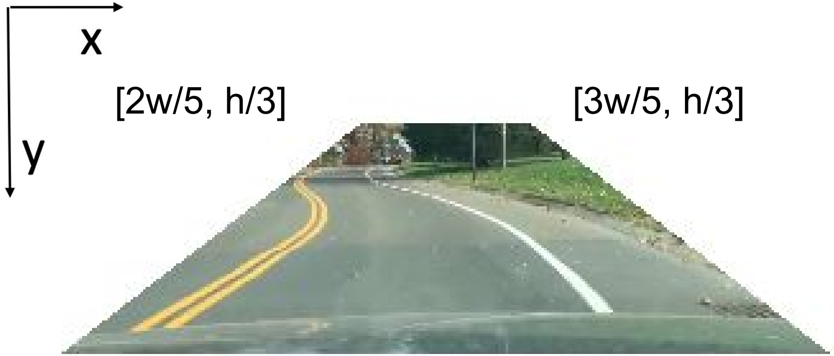
Assume:

- Constrained environment (sunny weather, no vehicles ahead, no cross-sections)
- Camera with fixed location and orientation
- Road is mostly flat



Region of Interest

ROI → masking → max filter → sliding window → median operation and spline

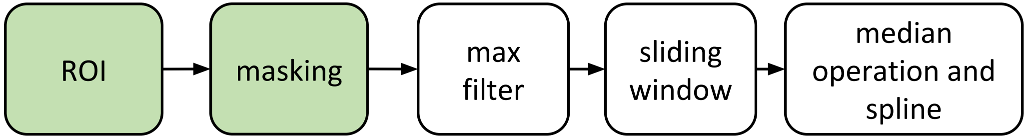


❑ Result demo of ROI

Objective: reduce noise effect and reduce computational cost

Parameters: Four vertices of ROI, determined based on training images

Masking



Objective: find yellow and white lanes

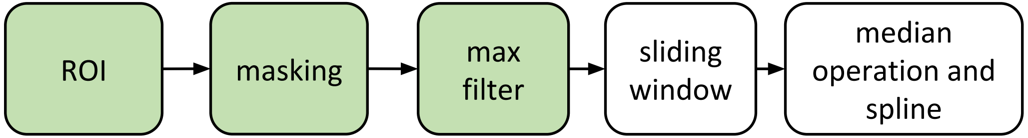
❏ Result Demo of Masking

Parameters:

- Yellow range: [0, 100, 170] ~ [255, 240, 255]
- White range: 210 ~ 255 in gray image

Output: binary image

Max filter

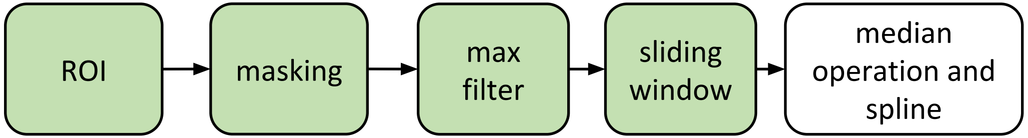


❑ Result Demo of Max Filter (before and after)

Objective: Connect discrete regions to connected lines

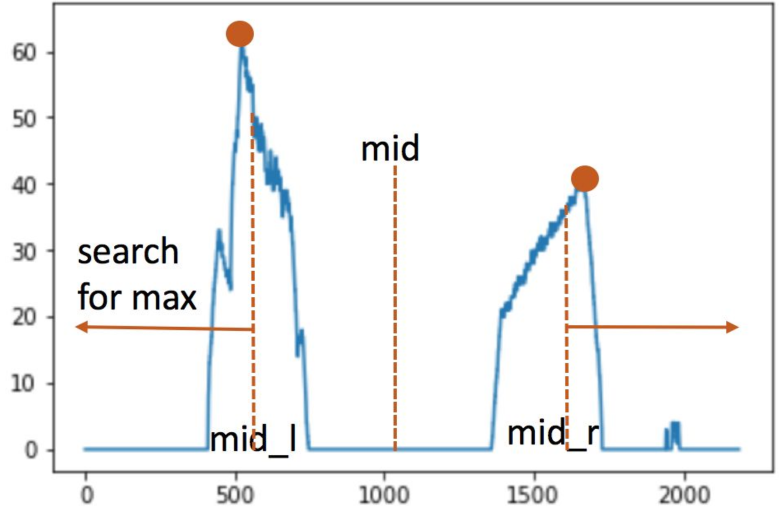
Parameters: Kernel size 9*9

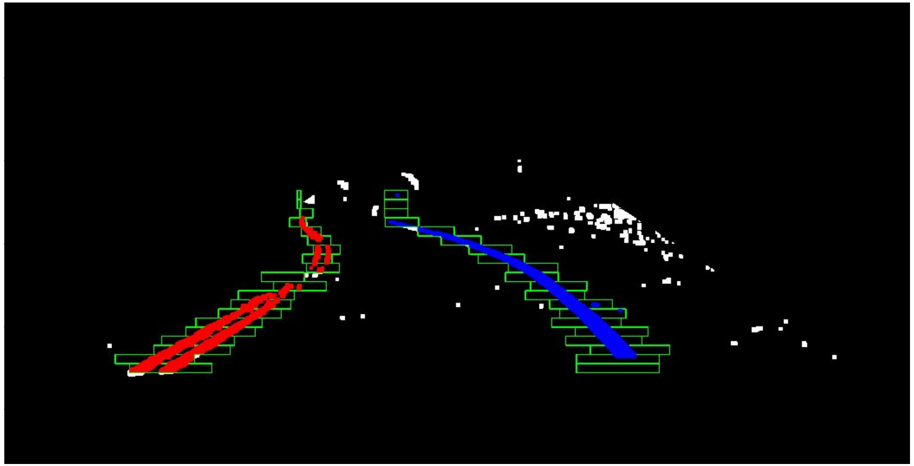
Sliding window



Objective: detect lane pixels

Design parameter: window number, window height

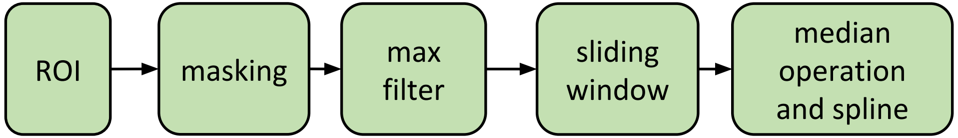


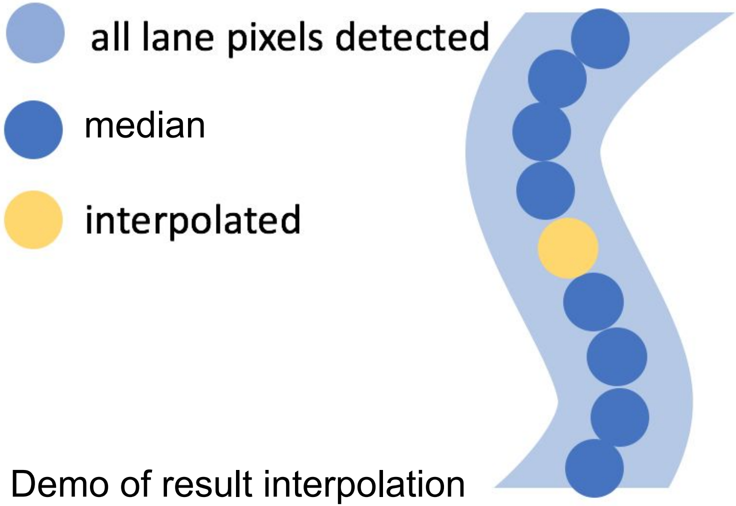


- ❑ Finding the start points of windows by histogram
- ❑ Result demo of sliding window method

Pixel extraction and spline interpolation

- Objective: for every y, find x position using spline interpolation
- Result is a pixel line
- Gaps filled by spline interpolation

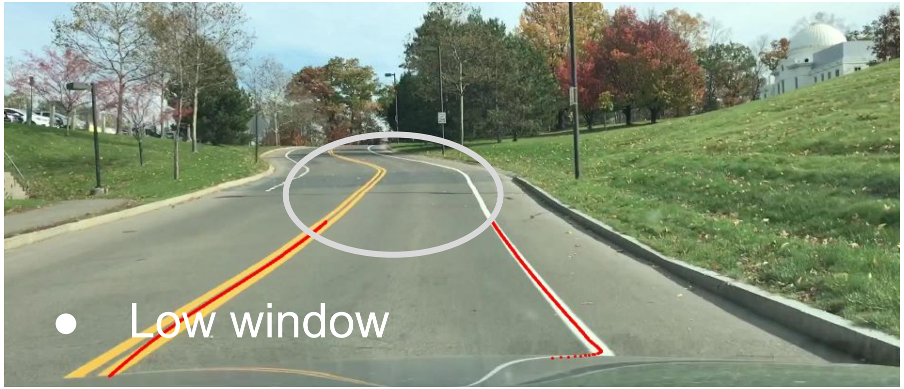


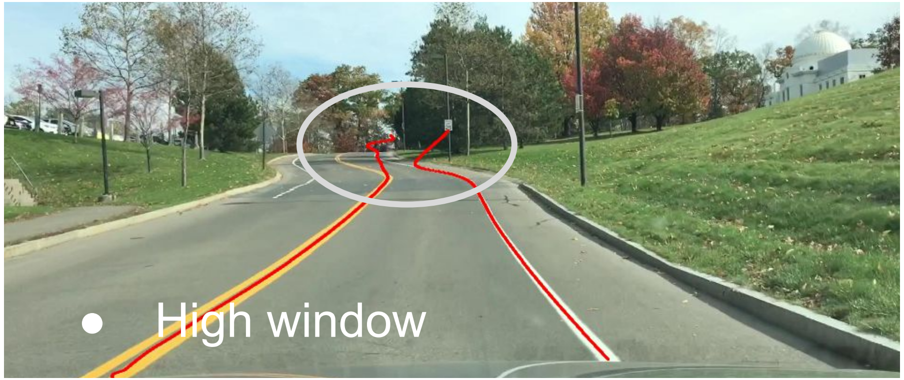


- ❑ Demo of result interpolation

- ❑ Output result

Optimization and Parameter tuning





Data Source

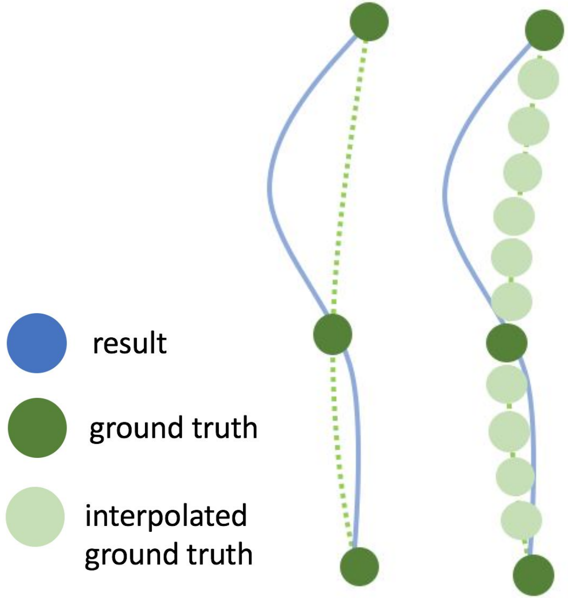
- Filmed by our team in Ithaca
- Uniformly sample images from video clips
- Algorithm development (Training images): 10 images
- Evaluation (Testing images): 70 images

- Examples of data images

Ground truth labeling and linear interpolation

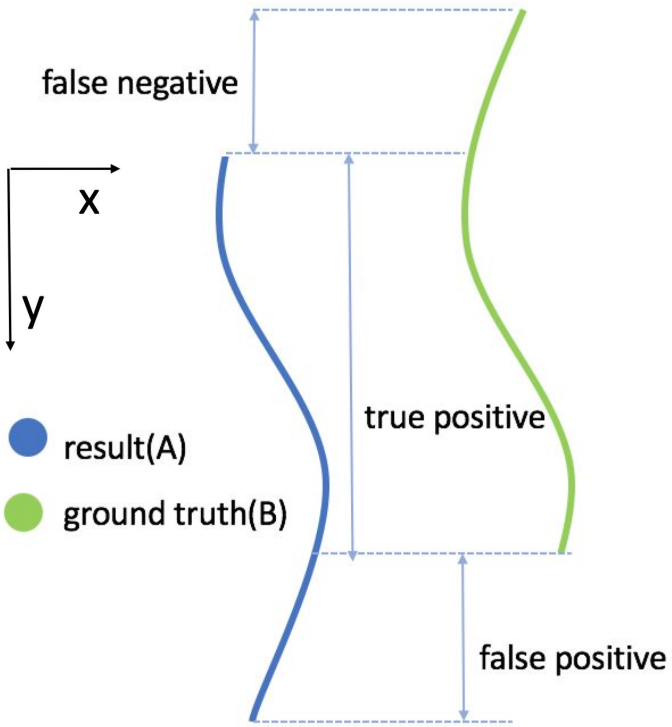
- 3 people manually labeled ground truth
- 3 - 8 data points for each side
- Linear interpolation

❏ Example of ground truth labeling



❏ Demo of ground truth interpolation

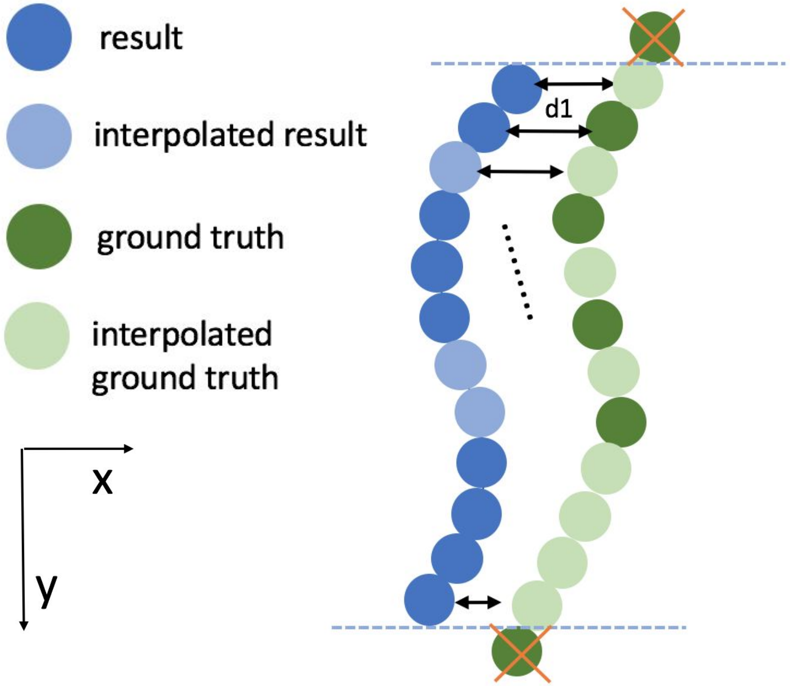
Evaluation function: success detection



$$FPC = \frac{|A| - (|A \cap \bar{B}| + |\bar{A} \cap B|)}{|A|}$$

- The detection is successful if FPC > 0.7 for both sides
- 0.7 determined by considering human marking error

Evaluation function: precision of detection



Result offset:

$\max(d_i)\,, i = 1, \ldots \mathit{length\ of\ result}$

Result offset:

$\max(d_i)\,, i = 1, \ldots \mathit{length\ of\ result}$

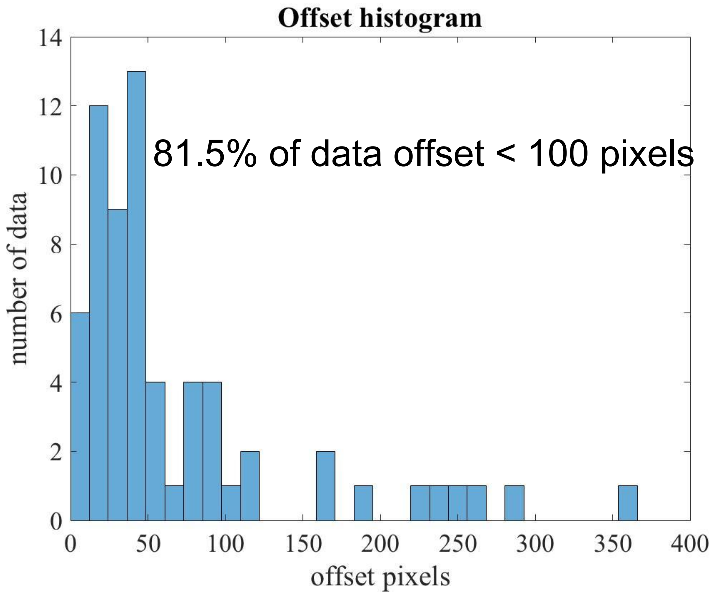
Hypothesis

- Achieve success detection rate of 80%, since previous work by Chen, Wang, etc. reaches 80%[1]
- For success detection, achieve average offset less than 100 pixels (approximated by ⅓ lane width)

[3]Qiang Chen,Hong Wang. A Real-time Lane Detection Algorithm Based on a Hyperbola-Pair Model.2006,13(15): 510-515.

Evaluation result

Success Detection with high precision:

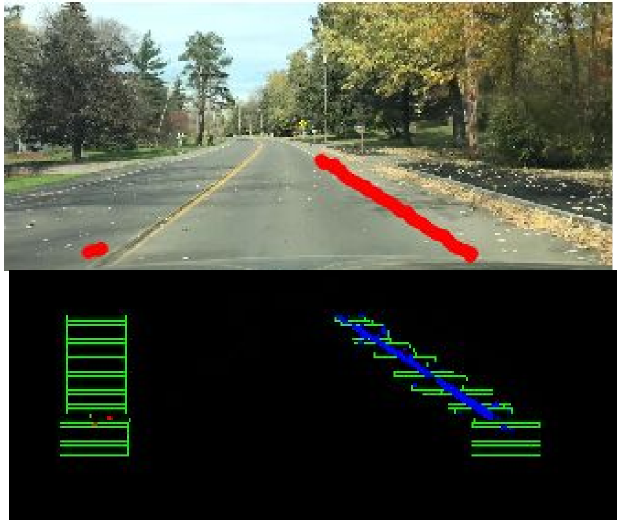


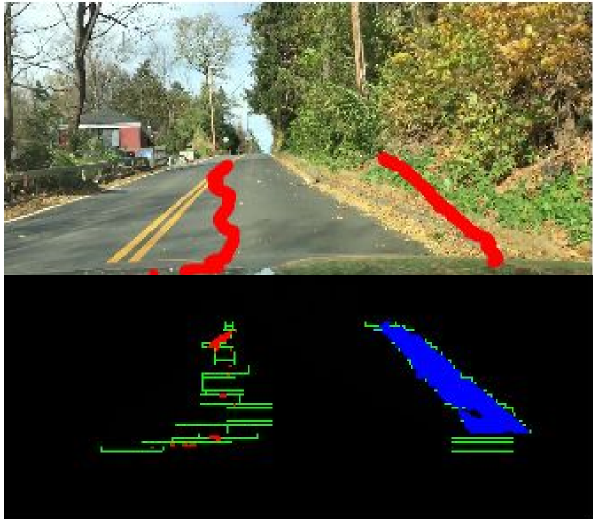
- The algorithm has achieved success rate of 80%
- Average FPC is 85% and offset is 70.9 pixels (image width: 1920)

Discussion

Causes of error:

- Starting positions of sliding window
- Lane marks not continuous
- Lane marks covered by obstacles(car, leaves, people)





- ❏ Example of failed/ low precision detection

Conclusion

- The algorithm has achieved success rate of 80%. 81.5% of the images achieved high precision (offset <100 pixels)
- Average offset is 6% of image size, correspond to less than ⅓ of lane width
- Overall, the algorithm achieved a promising result with an acceptable computing complexity compared to previous published works
- Algorithm is sensitive to parameters such as ROI, and yellow and white threshold

Thank you!
Questions