



# Dash-Cam Single Image Processing

## Real Time case study

- Group G4: Amin Ghandi - Bahar Oveis

# Problem Definition

---

- ◆ There are millions of dash-cams which mostly are used for critical scenes and accidents.



# Problem Definition

---

- ❖ However, these dash-cam records can be used for more applications. With the use of image processing, we are able to use these records for urban traffic analysis, urban city mapping, obstacle detection, traffic sign recognition, driving warning etc.



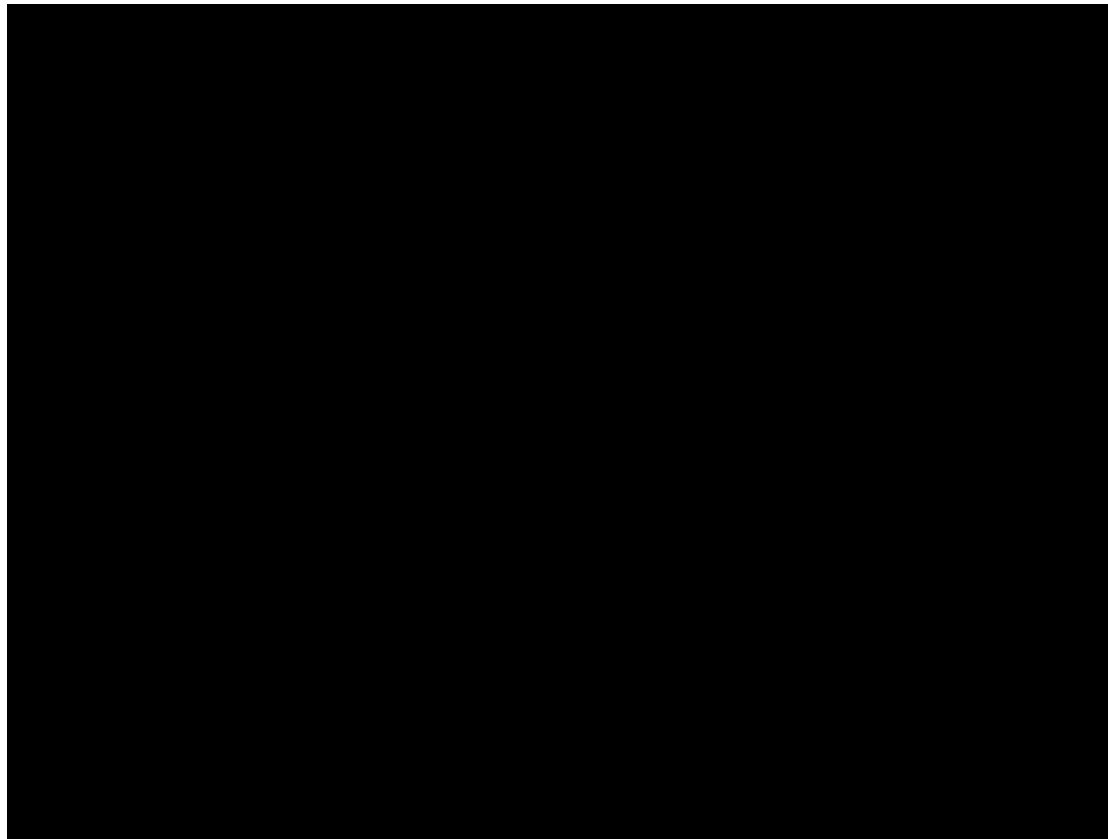
## Problem Definition

---

- ❖ In this project, we use Image Processing with conventional approach to extract some common information from a fairly long dash-cam record. For reaching our goal, to be able extract information in all time, we choose dash-cam record in harsh condition.

# Case Study

---



## Challenge:

---

- The case study is recorded in a foggy day.
- The case study is included different scene in different lightness since the driver is going from suburban to urban city.
- According to dash-cam record characteristics and weather condition, quality is not very high.
- We have to avoid time consuming methods and we are just allowed to use single image processing approach.
- We are looking for general methods to use. Therefore, we are not allowed to use different approaches for different scenes.

# Table contents:

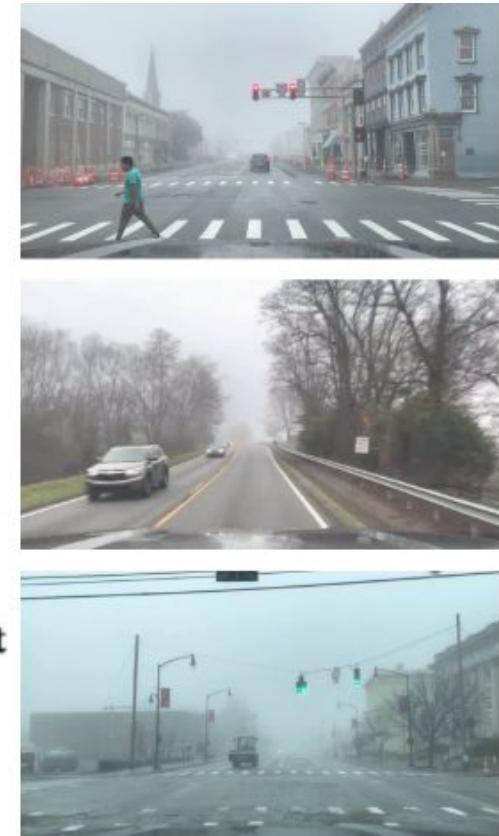
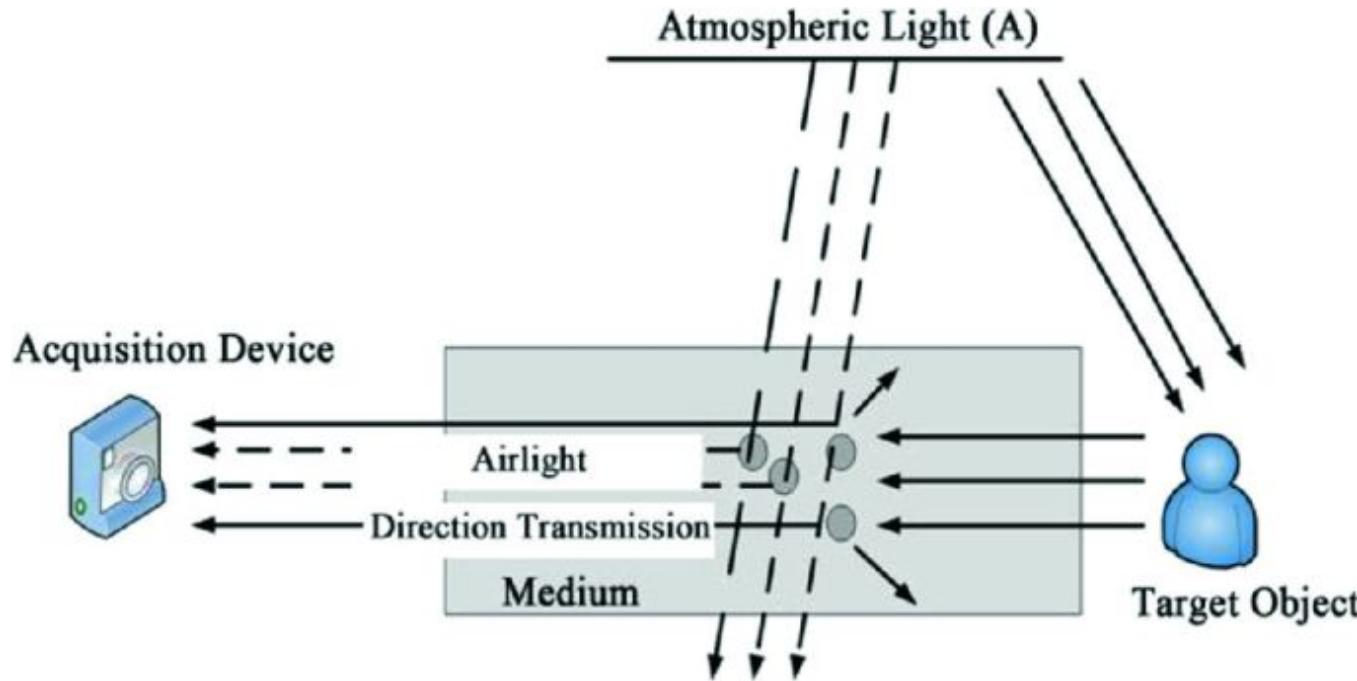
---

- **De-hazing**
  - General Theory
  - Transmission Map Estimation
  - Transmission Function Optimization
  - Outputs
- **Marks & lane detection**
  - Bird Eye View
  - Detection Methods & Comparison
  - Outputs
  - Additional Tests
- **Compression**
  - jpeg
- **Final results**
- **References**

---

# De-hazing

# De-hazing Theory



# De-hazing Theory

---

$$\mathbf{I}(x) = t(x)\mathbf{J}(x) + (1 - t(x))\mathbf{A},$$



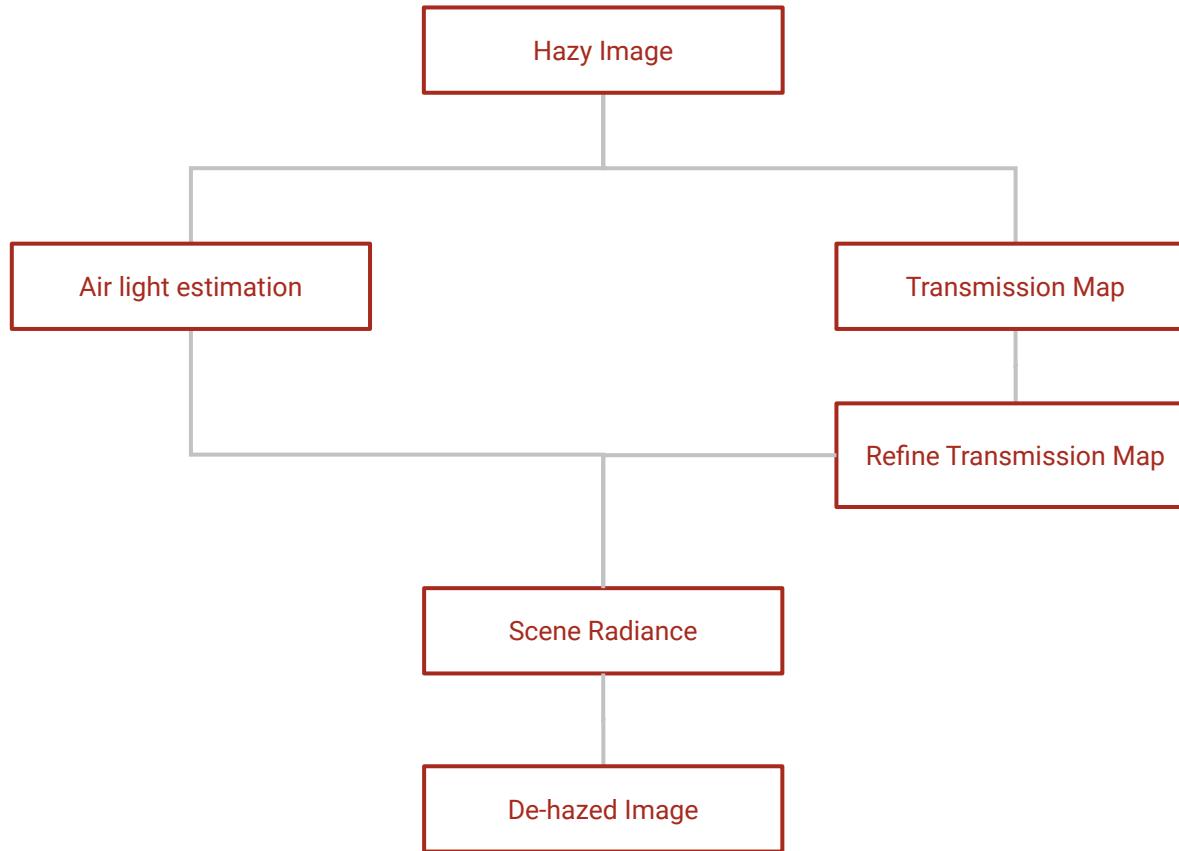
$$\mathbf{J}(x) = \frac{\mathbf{I}(x) - \mathbf{A}}{[\max(t(x), \epsilon)]^\delta} + \mathbf{A},$$

## Challenge:

- Dehazing from a single image is highly underconstrained, since the number of unknowns is much greater than the number of available equations.

# Road Map : De-hazing

---



# Boundary Constraint from Radiance Cube

---

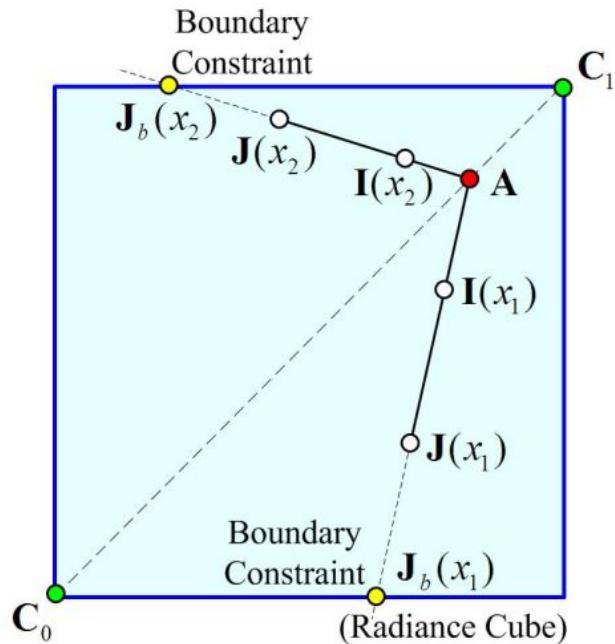
$$\mathbf{J}(x) = \frac{\mathbf{I}(x) - \mathbf{A}}{[\max(t(x), \epsilon)]^\delta} + \mathbf{A},$$



$$\frac{1}{t(x)} = \frac{||\mathbf{J}(x) - \mathbf{A}||}{||\mathbf{I}(x) - \mathbf{A}||}.$$

Conditions:

$$\mathbf{C}_0 \leq \mathbf{J}(x) \leq \mathbf{C}_1, \forall x \in \Omega,$$



# Boundary Constraint from Radiance Cube

---

$$\frac{1}{t(x)} = \frac{\|\mathbf{J}(x) - \mathbf{A}\|}{\|\mathbf{I}(x) - \mathbf{A}\|}.$$

- With the assumed condition and a supposed A , we can reach to J<sub>b</sub> (b ~ Boundary)

Conditions:

$$\mathbf{C}_0 \leq \mathbf{J}(x) \leq \mathbf{C}_1, \forall x \in \Omega,$$

0 ≤ t<sub>b</sub>(x) ≤ t(x) ≤ 1,

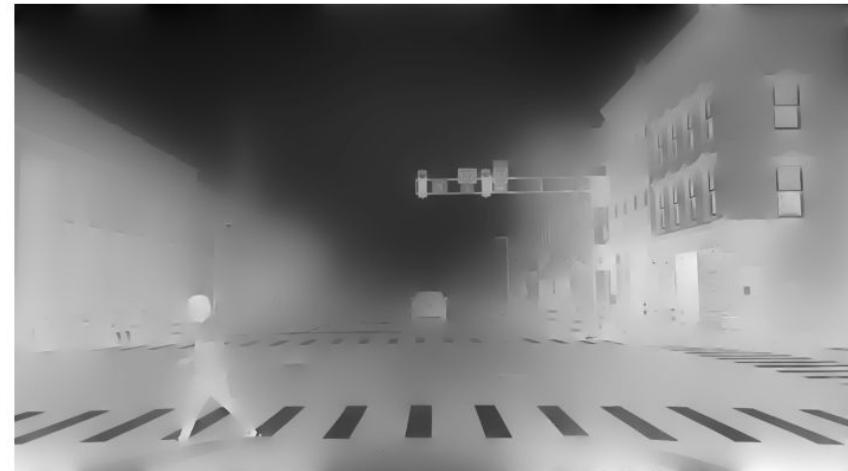
$$t_b(x) = \min \left\{ \max_{c \in \{r, g, b\}} \left( \frac{A^c - I^c(x)}{A^c - C_0^c}, \frac{A^c - I^c(x)}{A^c - C_1^c} \right), 1 \right\} \rightarrow \tilde{t}(x) = \max_{y \in \omega_x} t_b(y)$$

# Scene Transmission Estimation

---

$$\hat{t}(x) = \min_{y \in \omega_x} \max_{z \in \omega_y} t_b(z).$$

$$\frac{\lambda}{2} \|t - \hat{t}\|_2^2 + \sum_{j \in \omega} \|W_j \circ u_j\|_1 + \frac{\beta}{2} \left( \sum_{j \in \omega} \|u_j - D_j \otimes t\|_2^2 \right) \rightarrow \boxed{\text{Optimization}}$$



# Remove sky

orginal image



haze\_map image



defoged image



orginal image



haze\_map image



defoged image



orginal image



haze\_map image



defoged image



# Final De-hazed Image

---

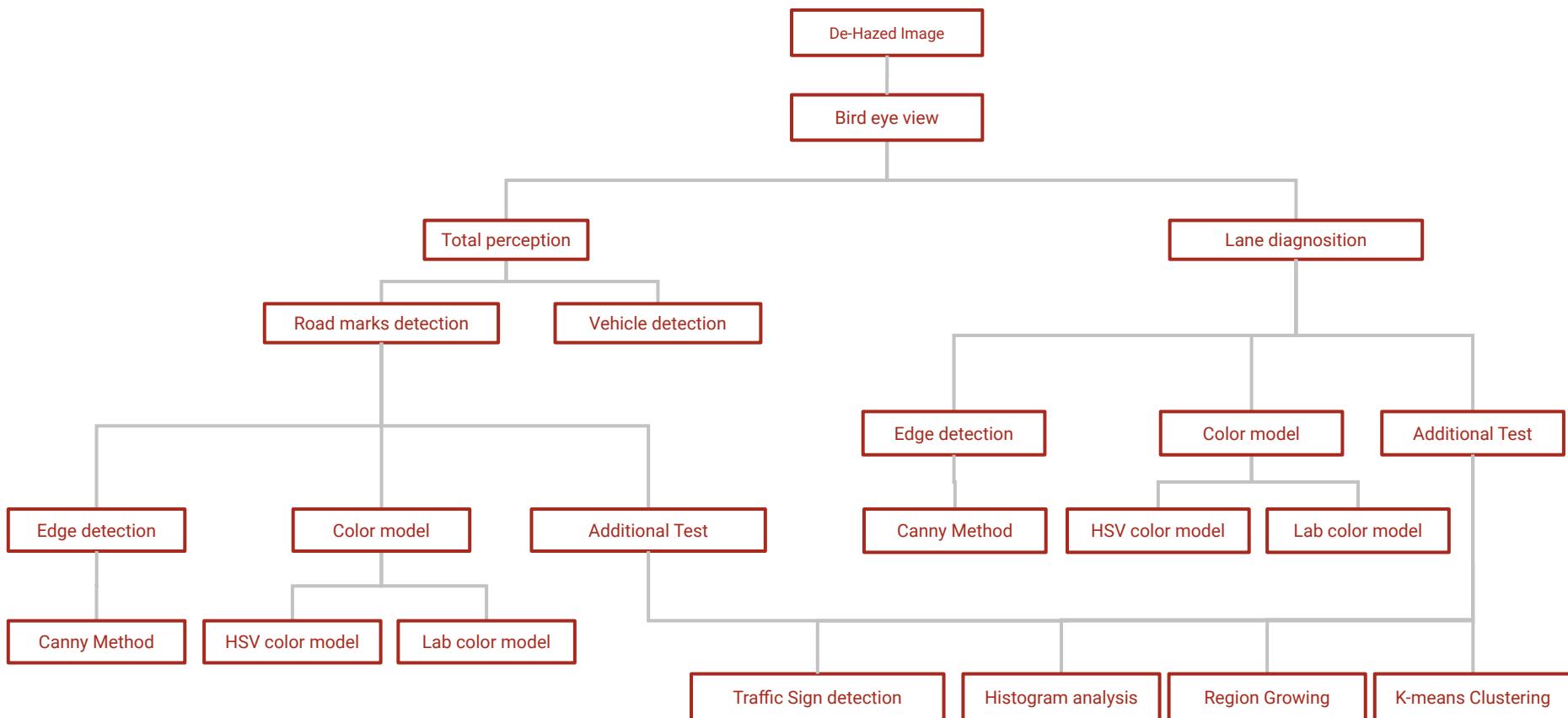


---

# Mark & Lane Detection

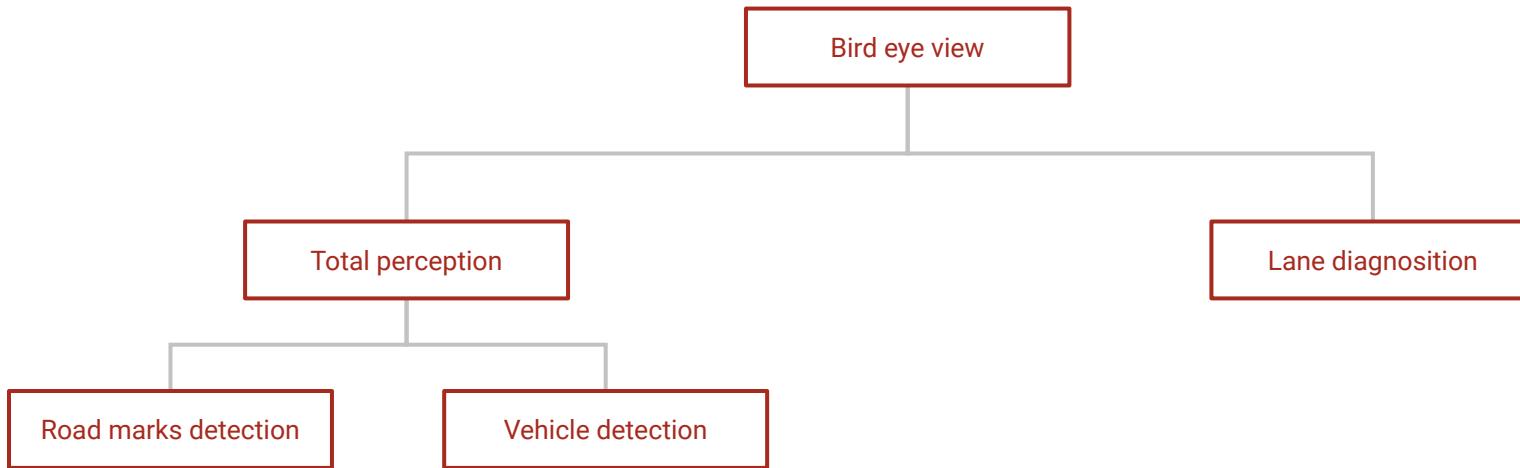
# Road Map : Mark & Lane Detection

---

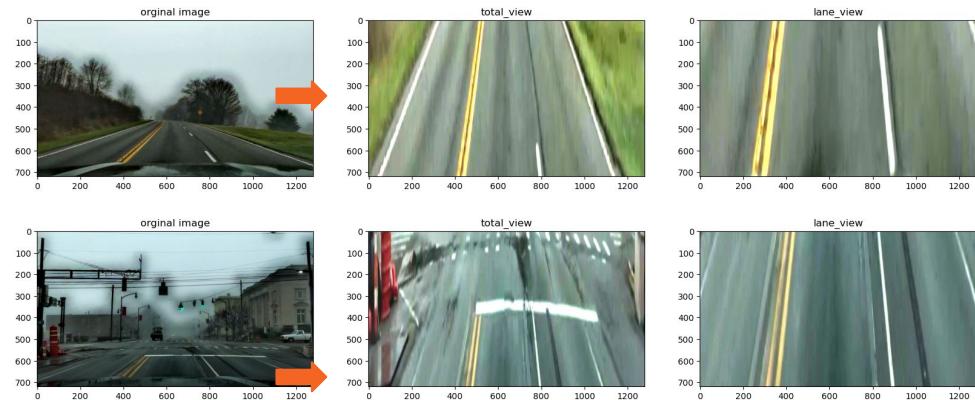
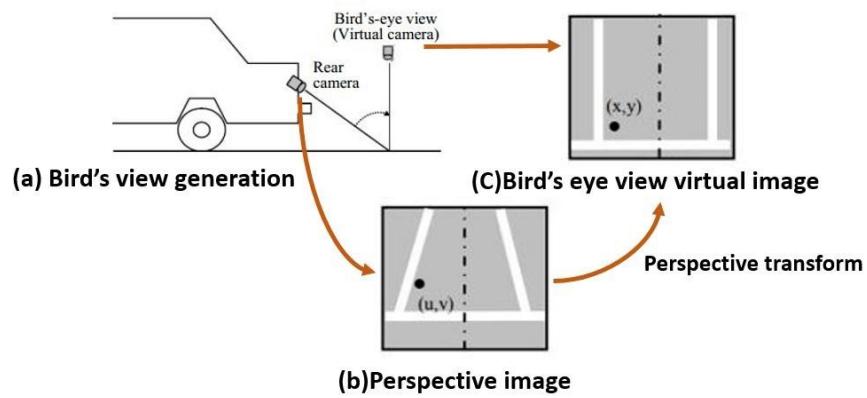


# Bird Eye View

---

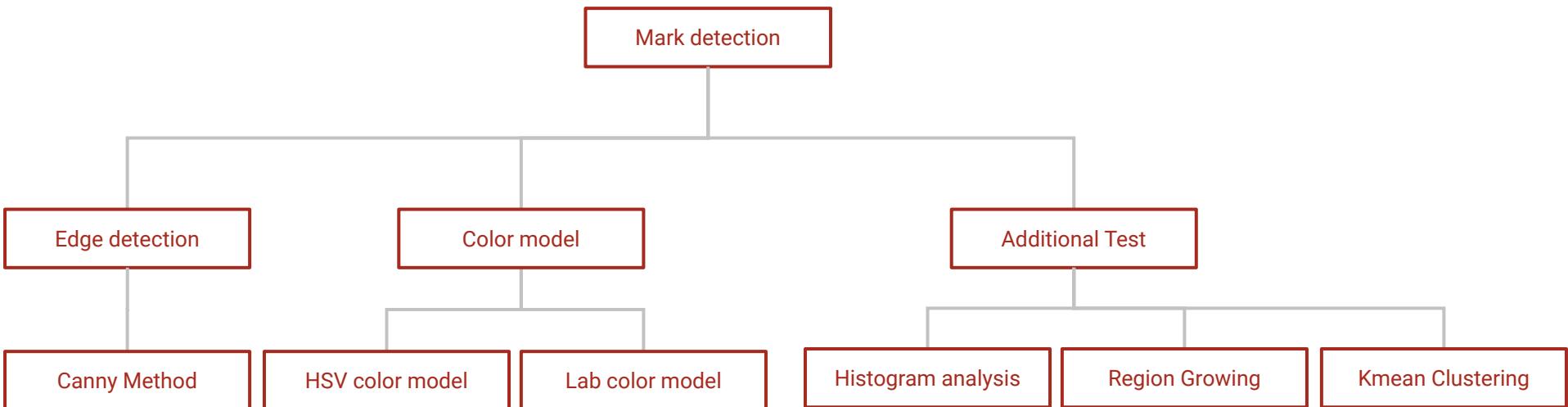


# Bird Eye View

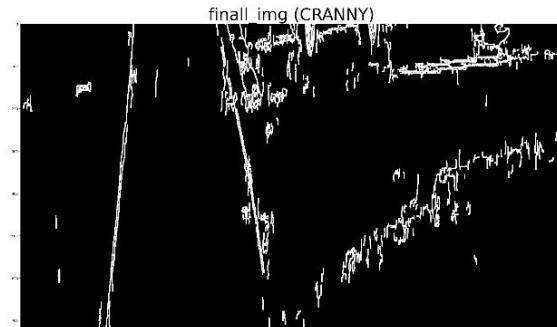
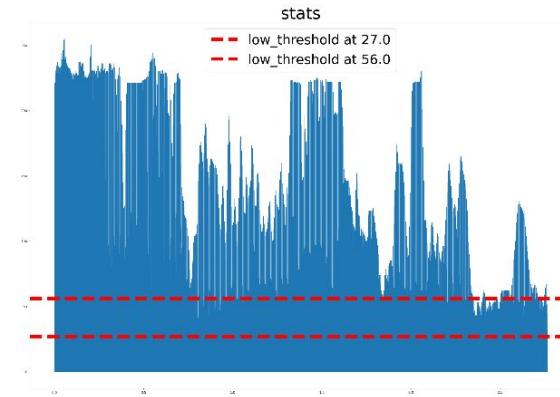
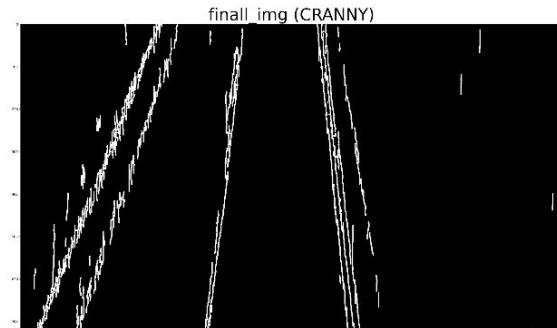
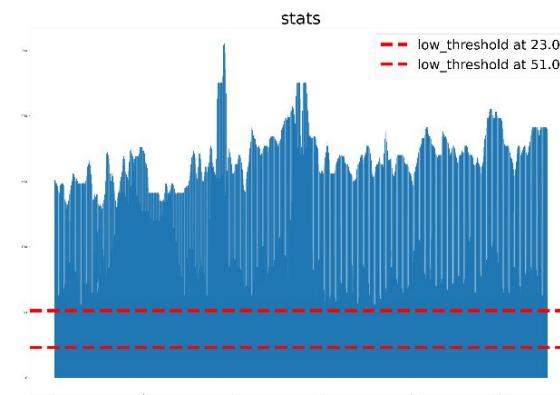
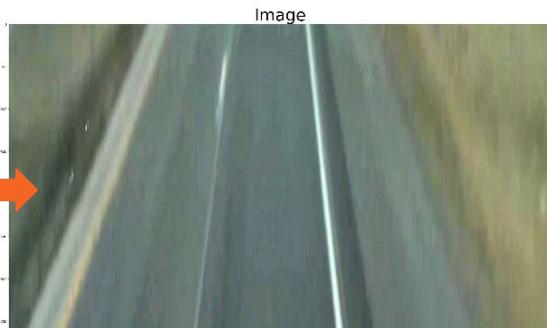


# Marks detection

---

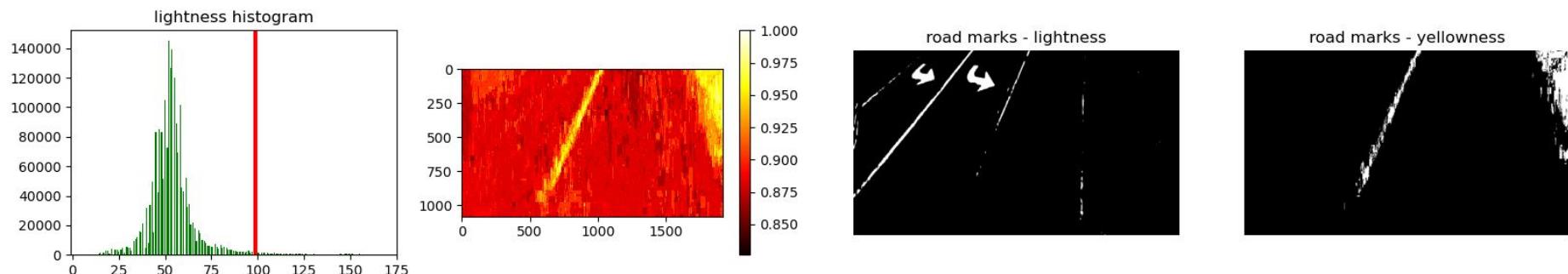
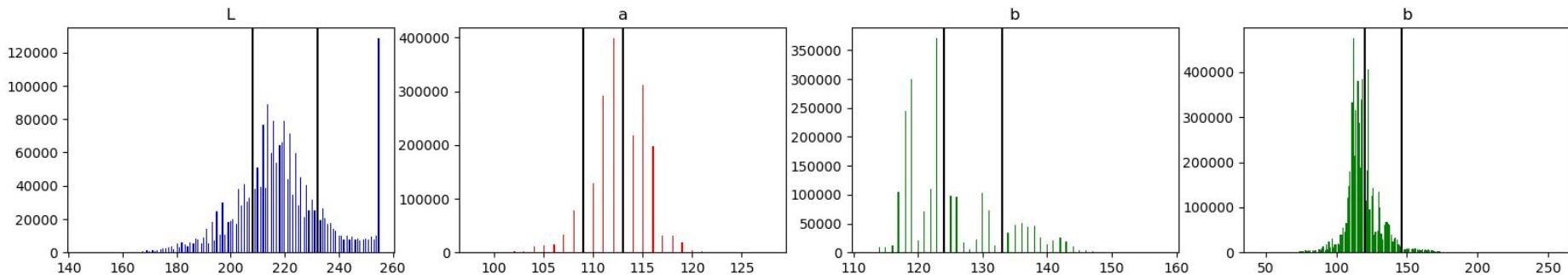


# Marks detection : Canny edge detection



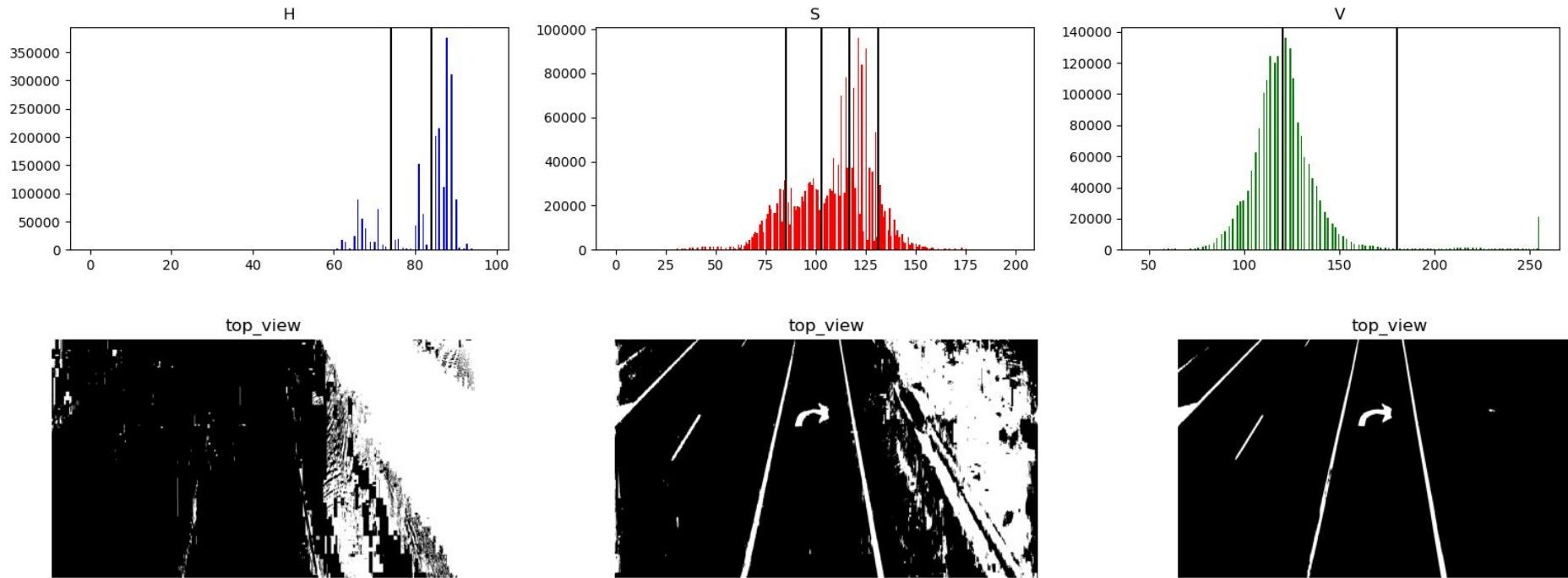
# Marks detection : Lab color model

---



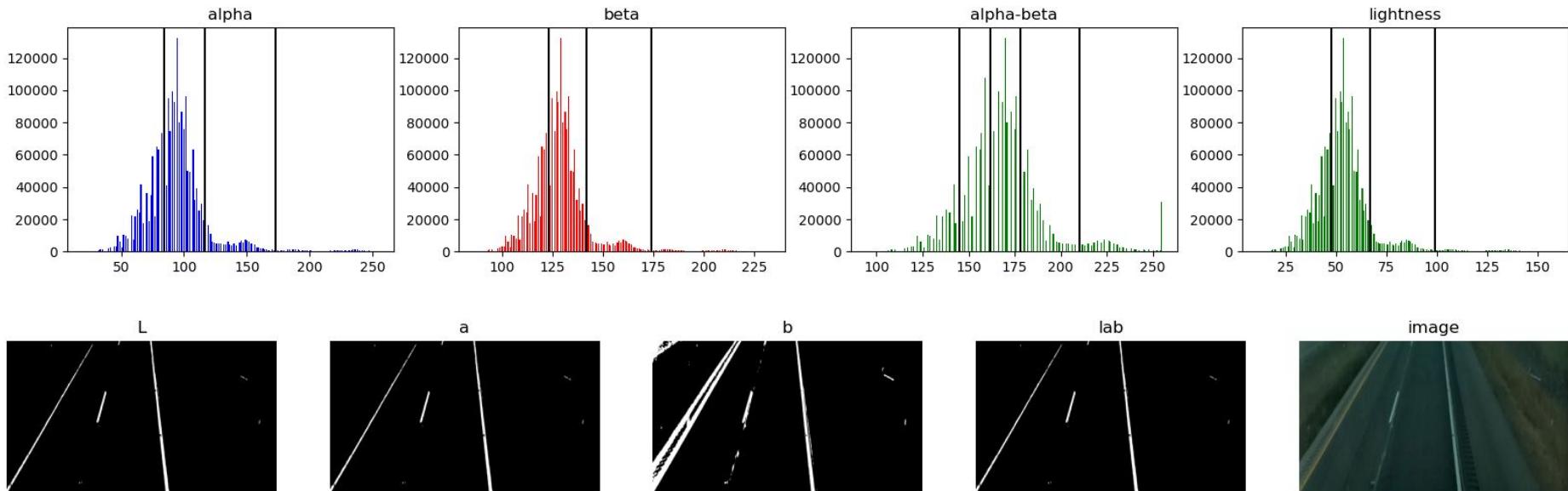
# Marks detection : HSV color model

---



# Marks detection : Contrast & Brightness

---

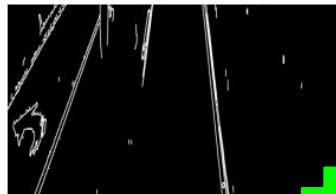


# Marks detection : Comparison

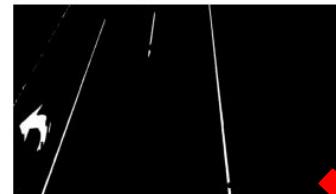
original image



canny method



histogram method



Lightness - Lab



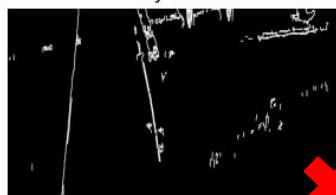
Value - HSV



original image



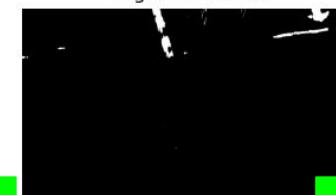
canny method



histogram method



Lightness - Lab



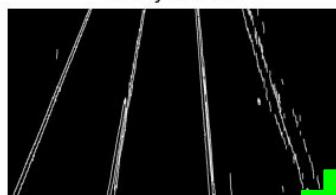
Value - HSV



original image



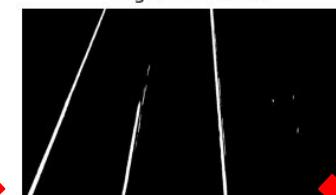
canny method



histogram method



Lightness - Lab



Value - HSV

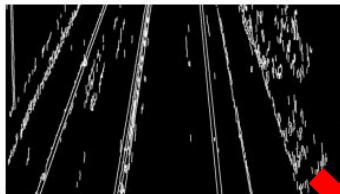


# Marks detection : Comparison

original image



canny method



histogram method



Lightness - Lab



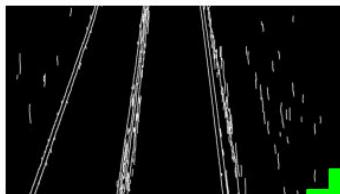
Value - HSV



original image



canny method



histogram method



Lightness - Lab



Value - HSV



original image



canny method



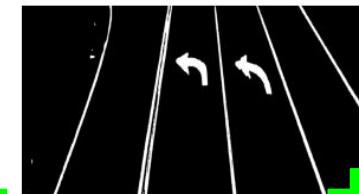
histogram method



Lightness - Lab



Value - HSV



# Marks detection : Comparison

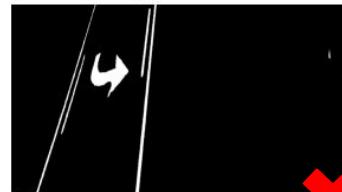
original image



canny method



histogram method



Lightness - Lab



Value - HSV



original image



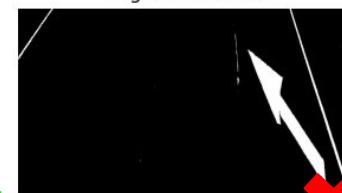
canny method



histogram method



Lightness - Lab



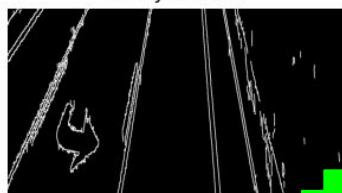
Value - HSV



original image



canny method



histogram method



Lightness - Lab

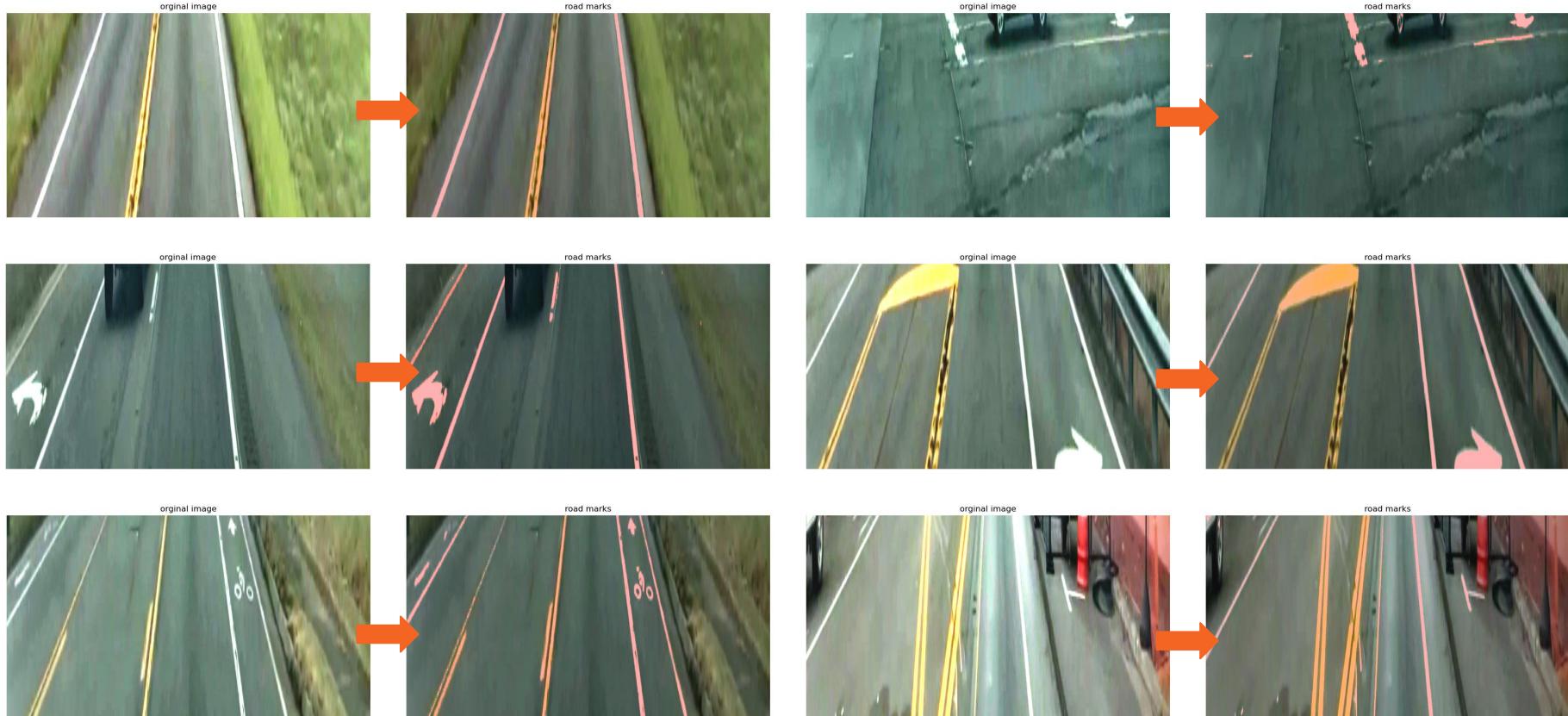


Value - HSV



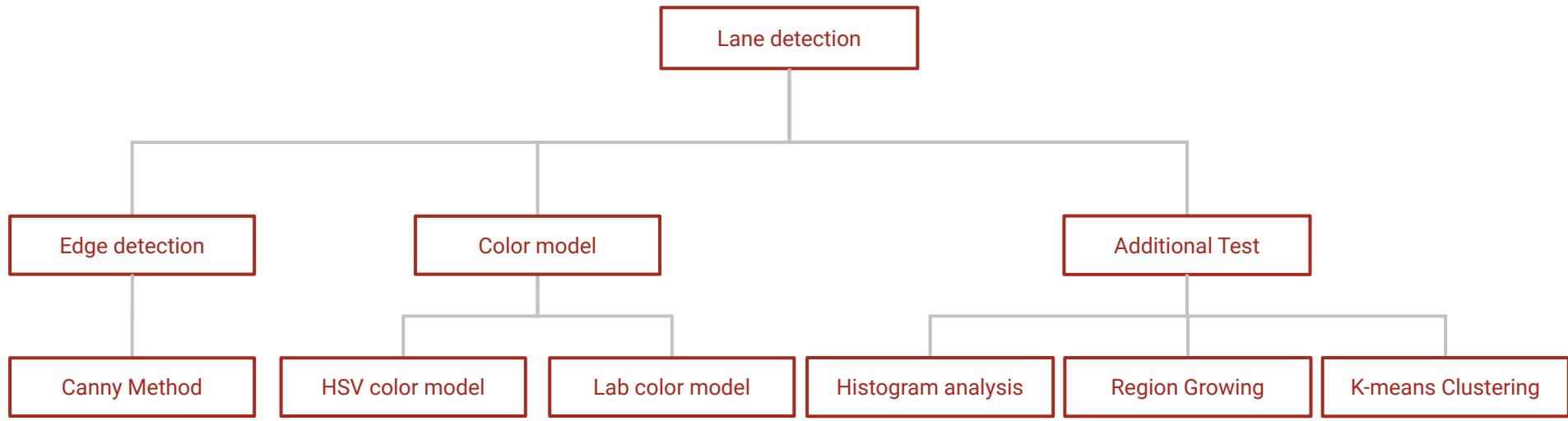
# Marks detection : Final Outputs

---



# Lane detection

---

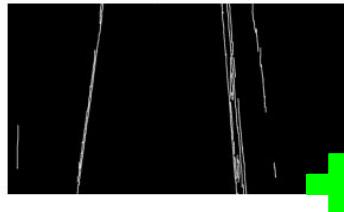


# Lane detection : Comparison

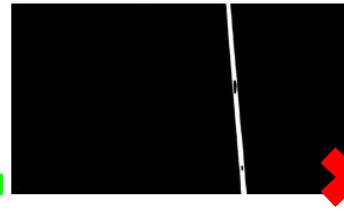
original image



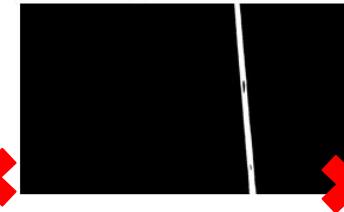
canny method



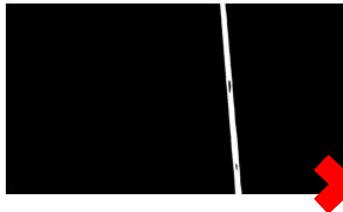
histogram method



Lightness - Lab



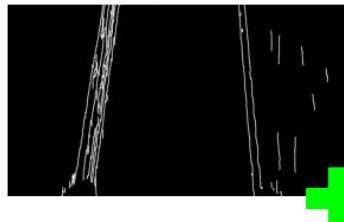
Value - HSV



original image



canny method



histogram method



Lightness - Lab



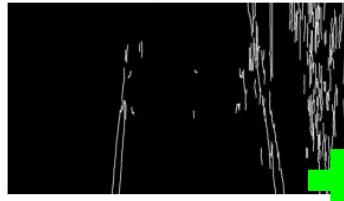
Value - HSV



original image



canny method



histogram method



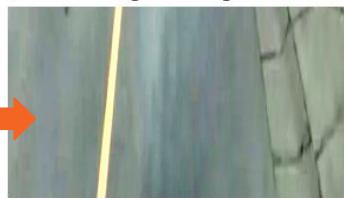
Lightness - Lab



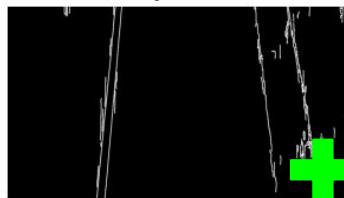
Value - HSV



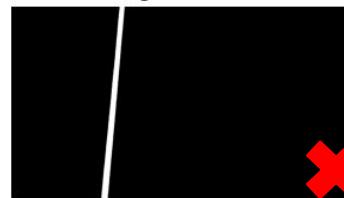
original image



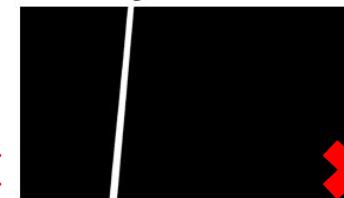
canny method



histogram method



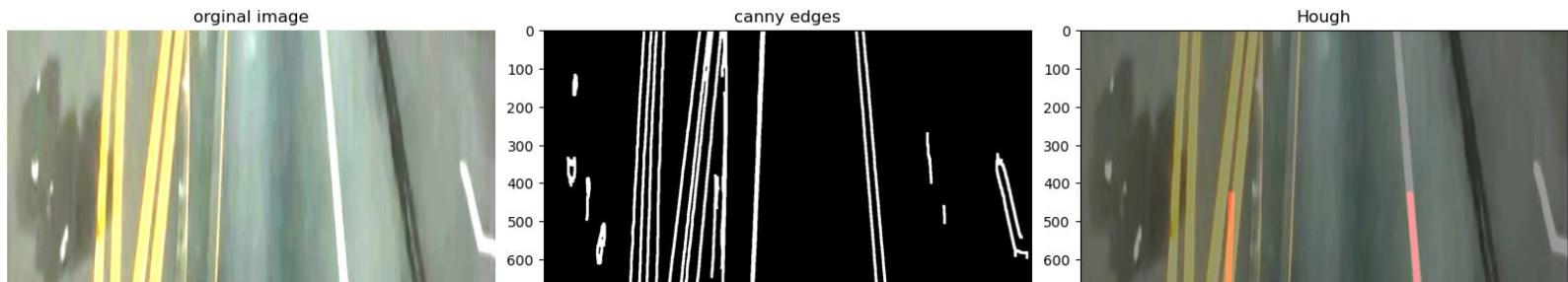
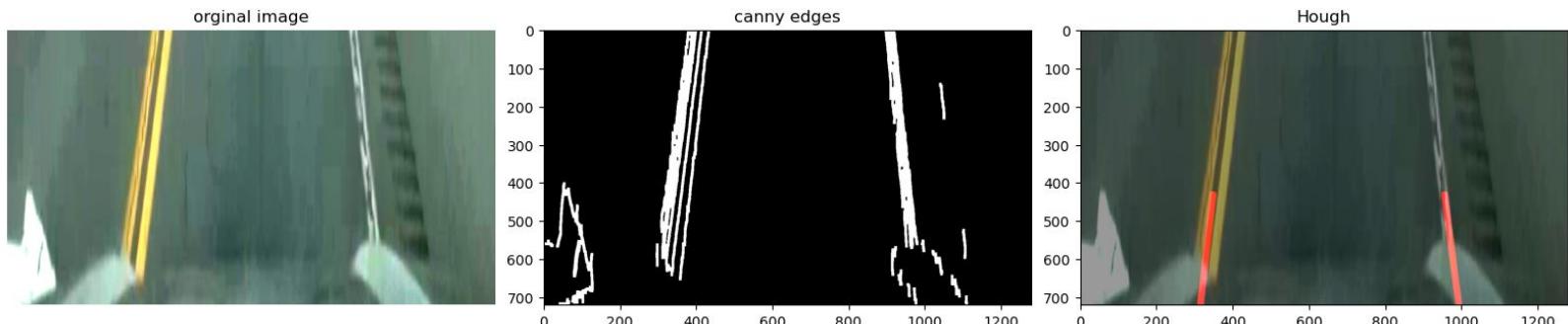
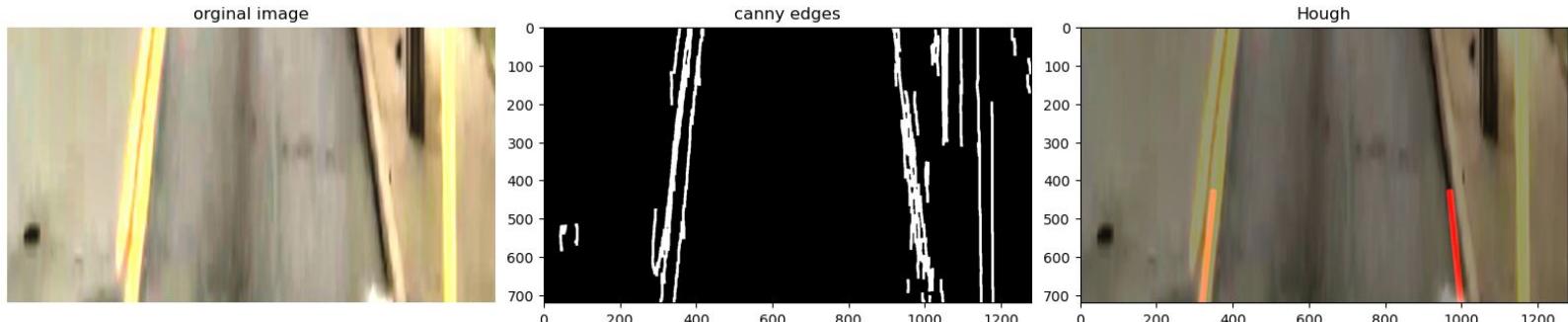
Lightness - Lab



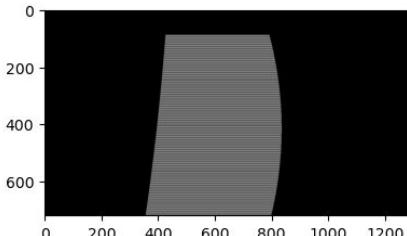
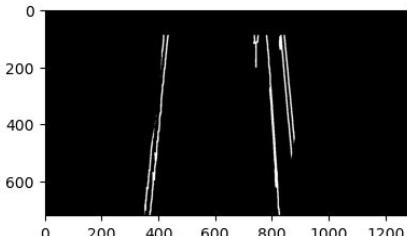
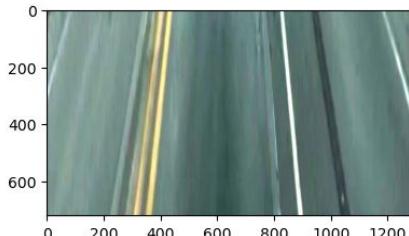
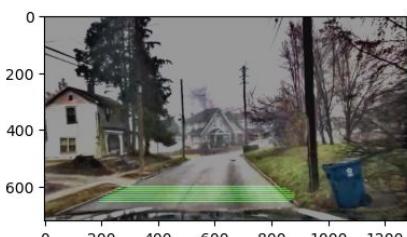
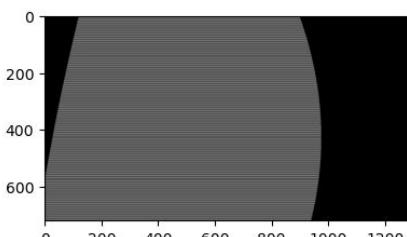
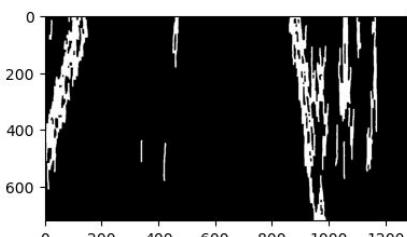
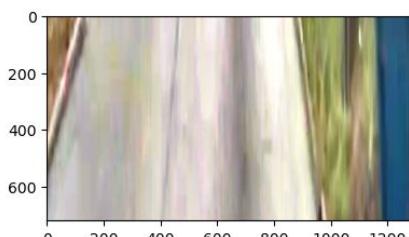
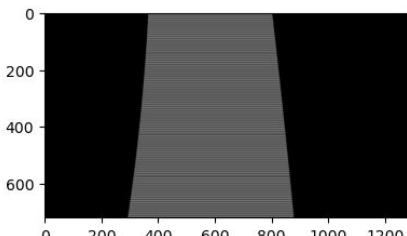
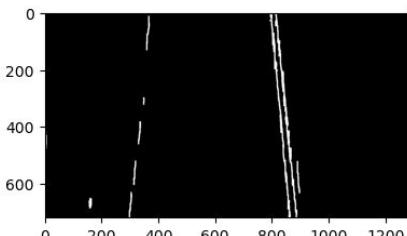
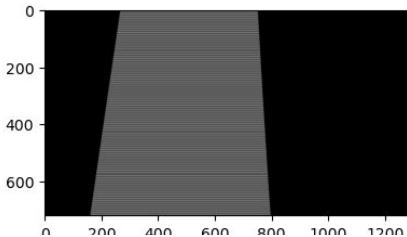
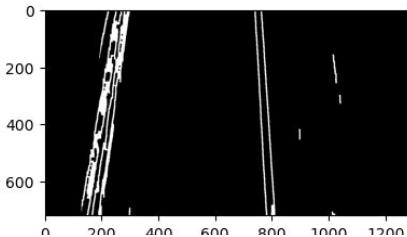
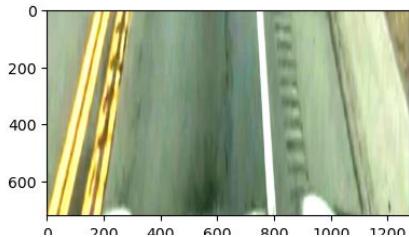
Value - HSV



# Lane detection : Final Outputs Hough



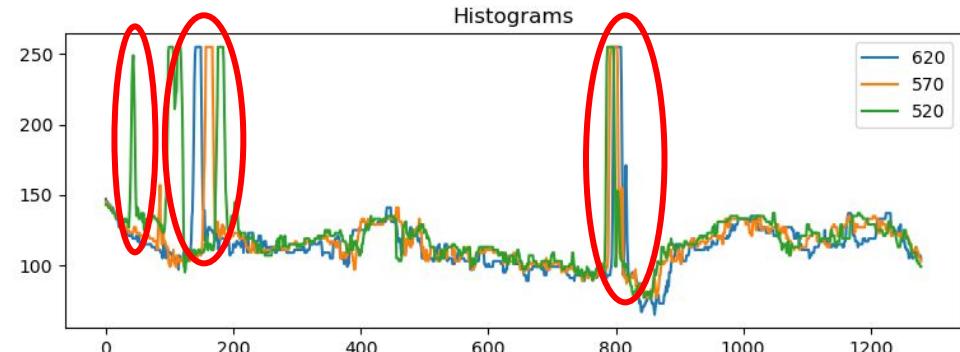
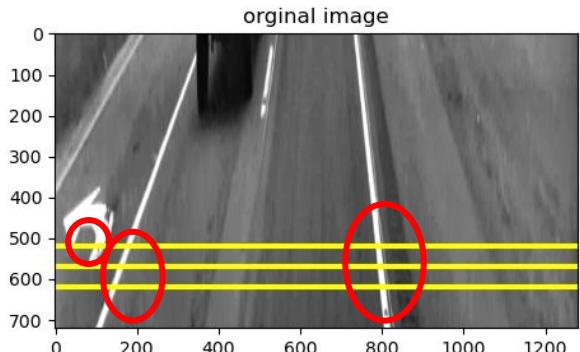
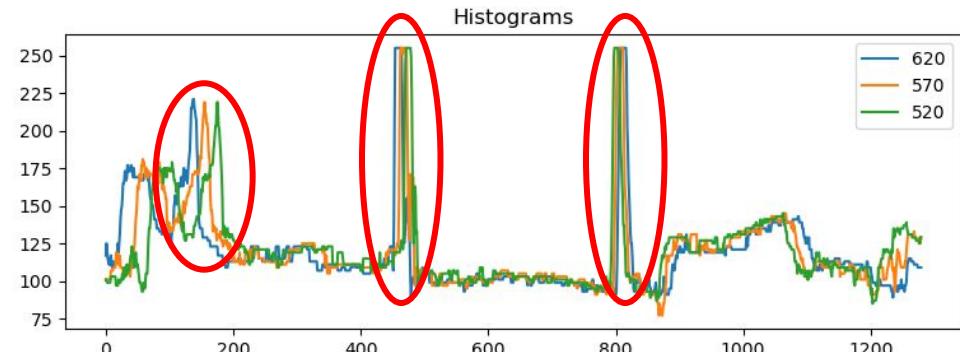
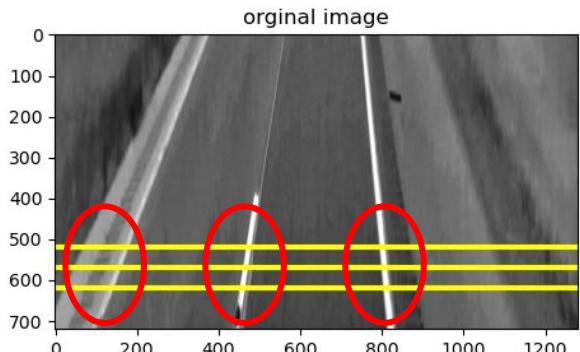
# Lane detection : Final Outputs - window sliding



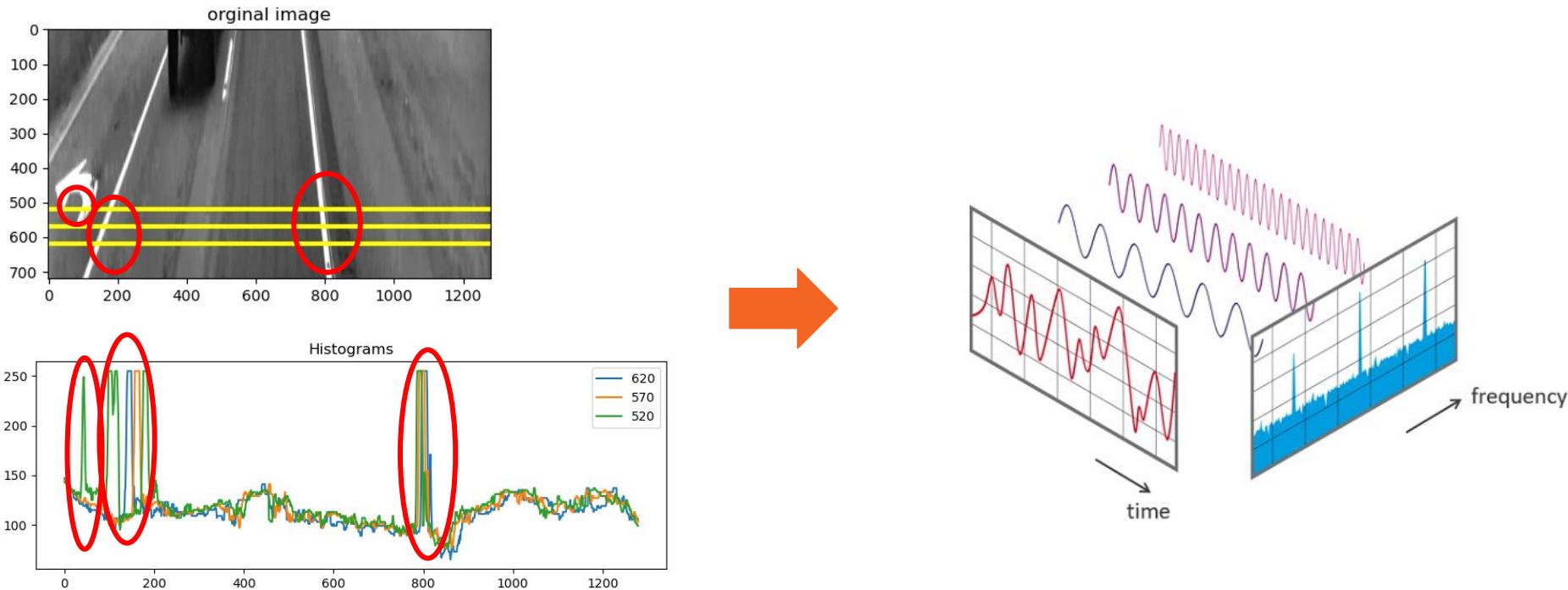
---

# Additional Test

# Additional Test : Histogram Analysis

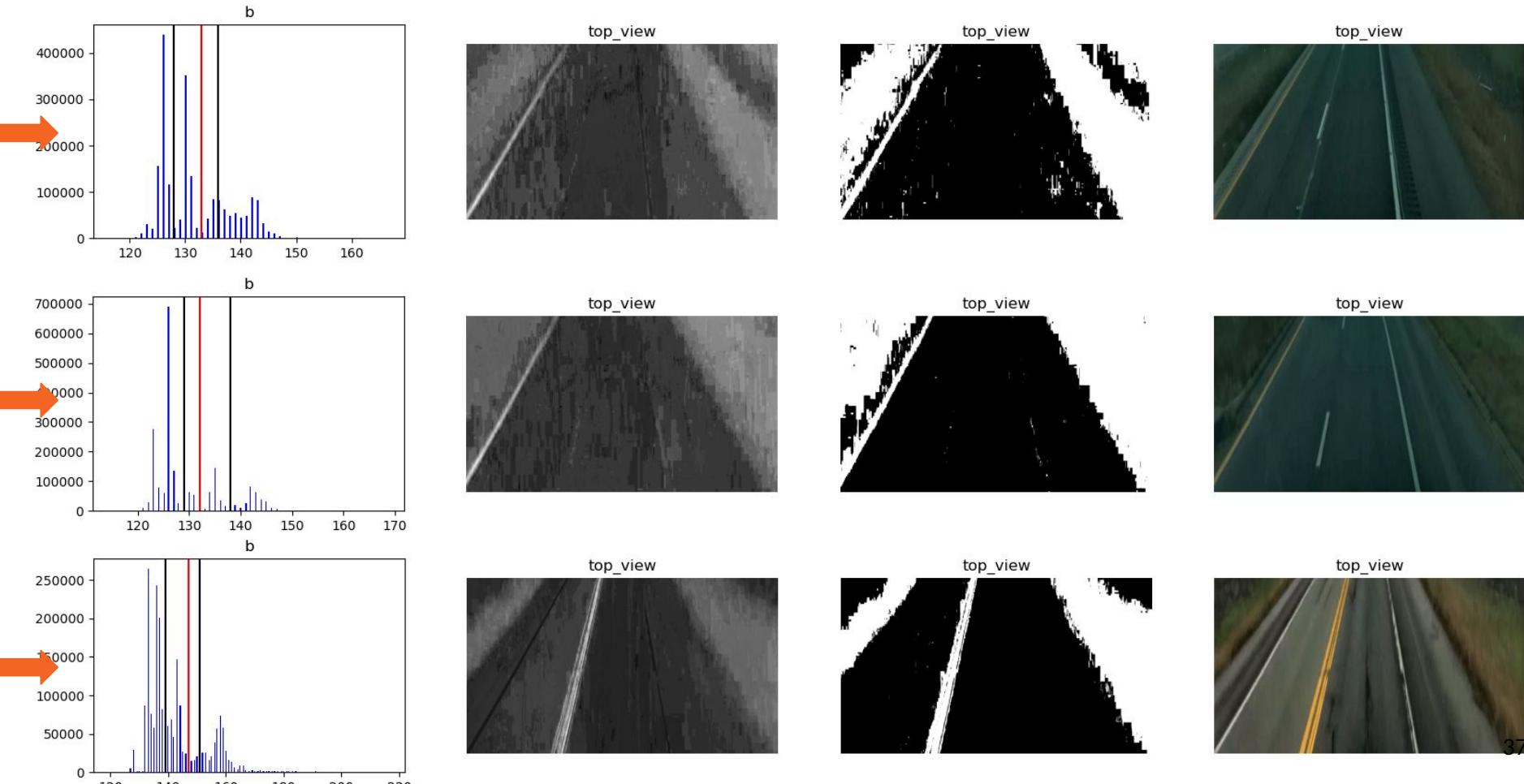


# Additional Test : Histogram Analysis

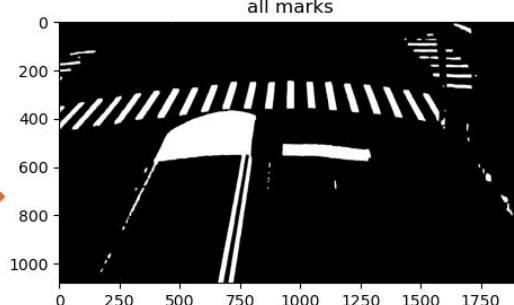
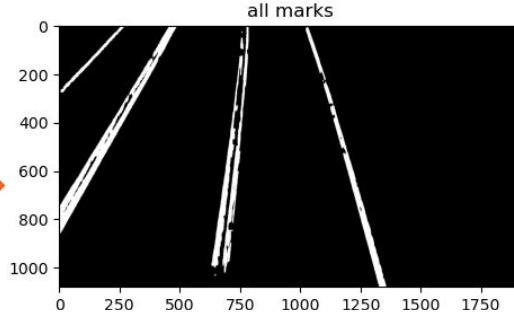
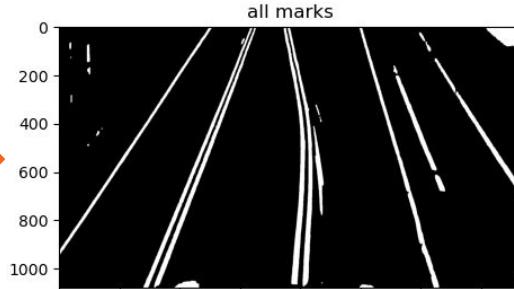


It seems interesting to model the problem as signal processing!

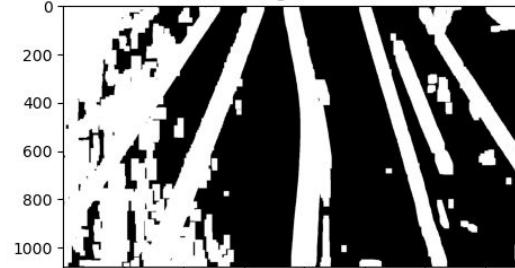
# Additional Test : Road Segmentation - Color models



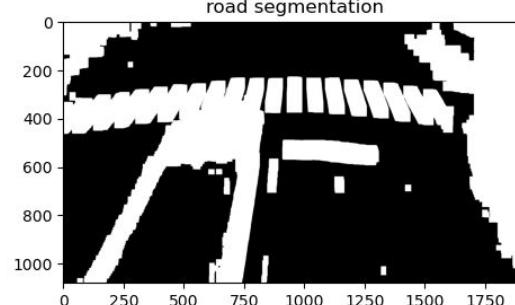
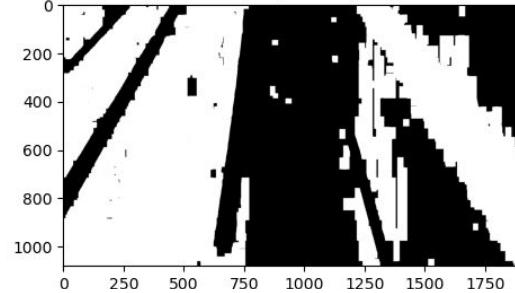
# Additional Test : Road Segmentation - Color models



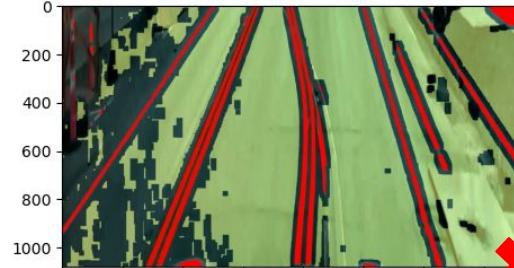
road segmentation



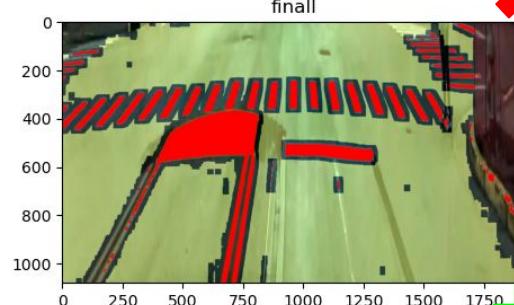
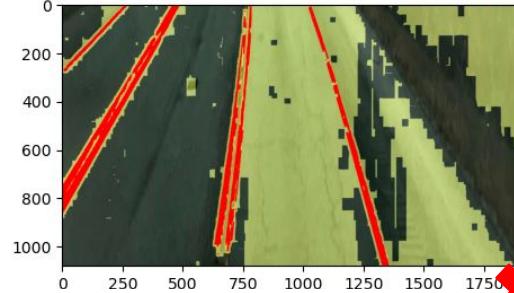
road segmentation



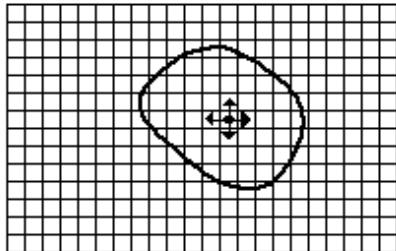
final



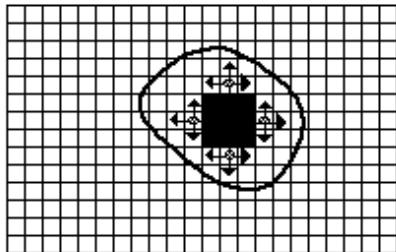
final



# Additional Test : Road Segmentation - Region\_Growing

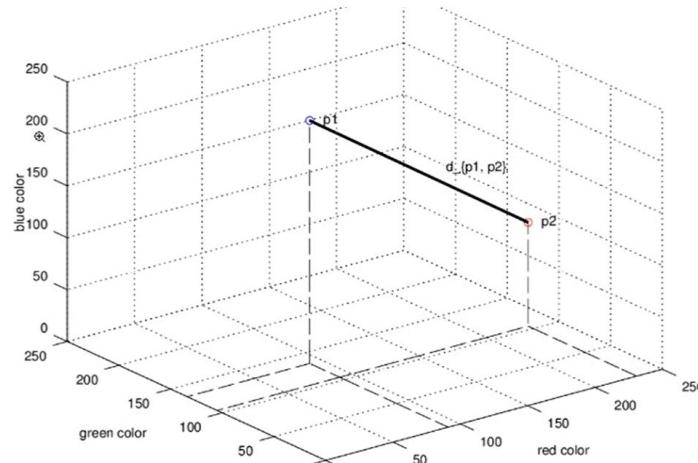


- Seed Pixel
- ↑ Direction of Growth



- Grown Pixels
- ⦿ Pixels Being Considered

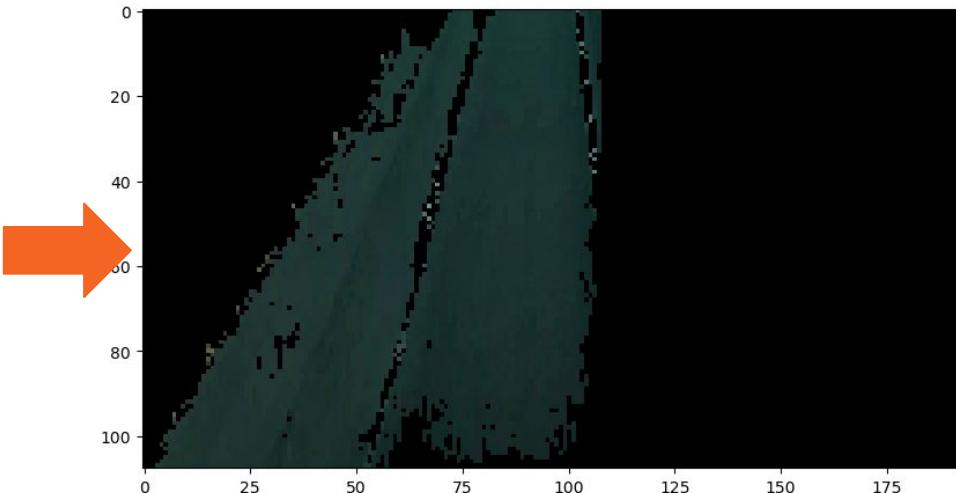
pixel p1
R: 91
G: 134
B: 234
pixel p2
R: 231
G: 105
B: 100



(b) Growing Process After a Few Iterations

# Additional Test : Road Segmentation - Region\_Growing

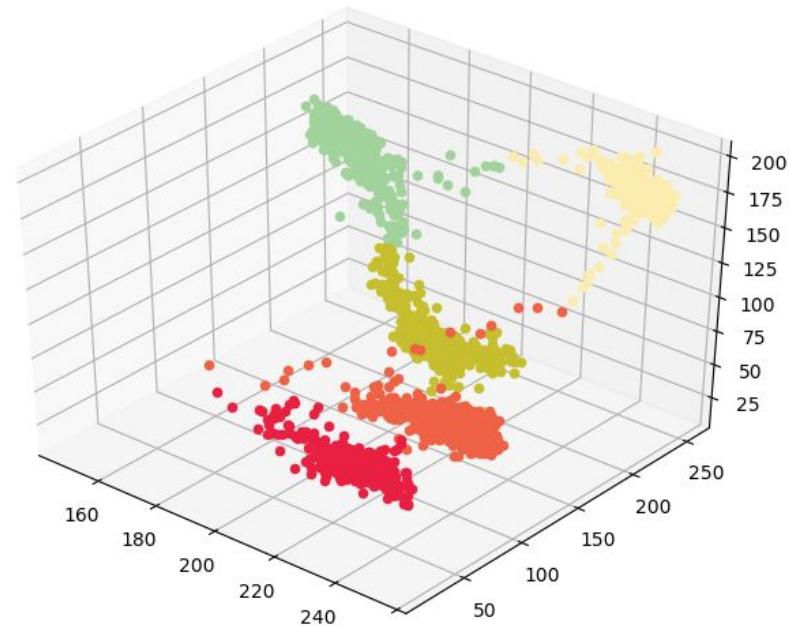
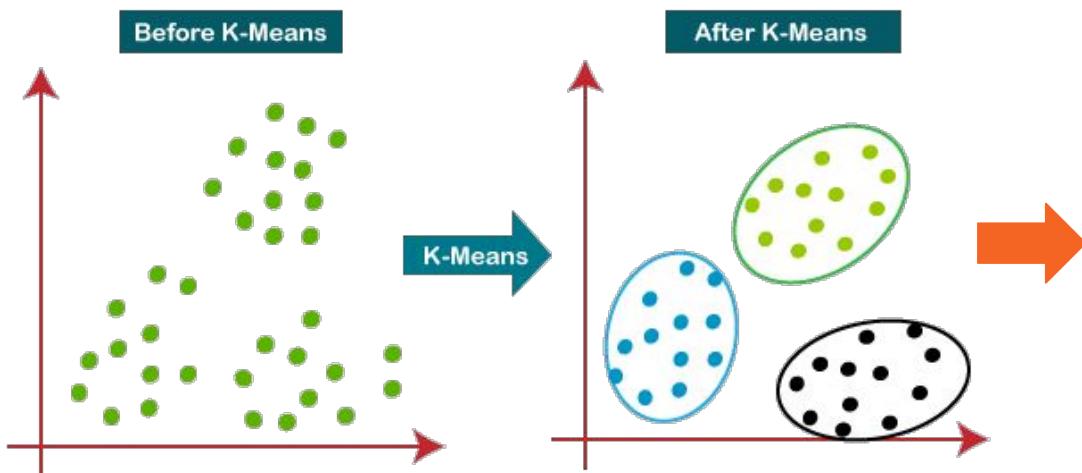
---



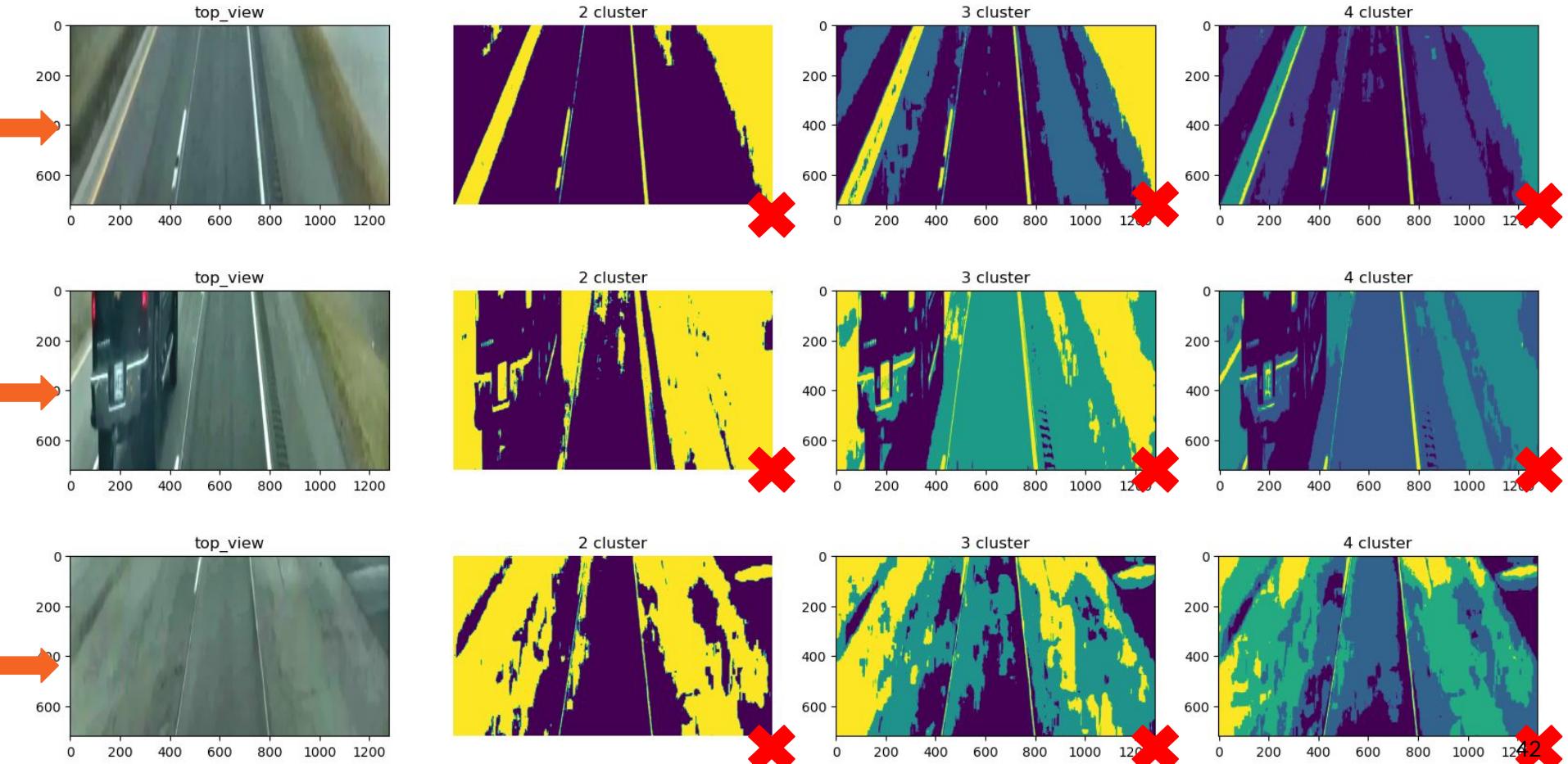
Adv : Good Results in Road Segmentation!

Dis\_Adv : High Running Time!

# Additional Test : Road Segmentation - K-means Clustering

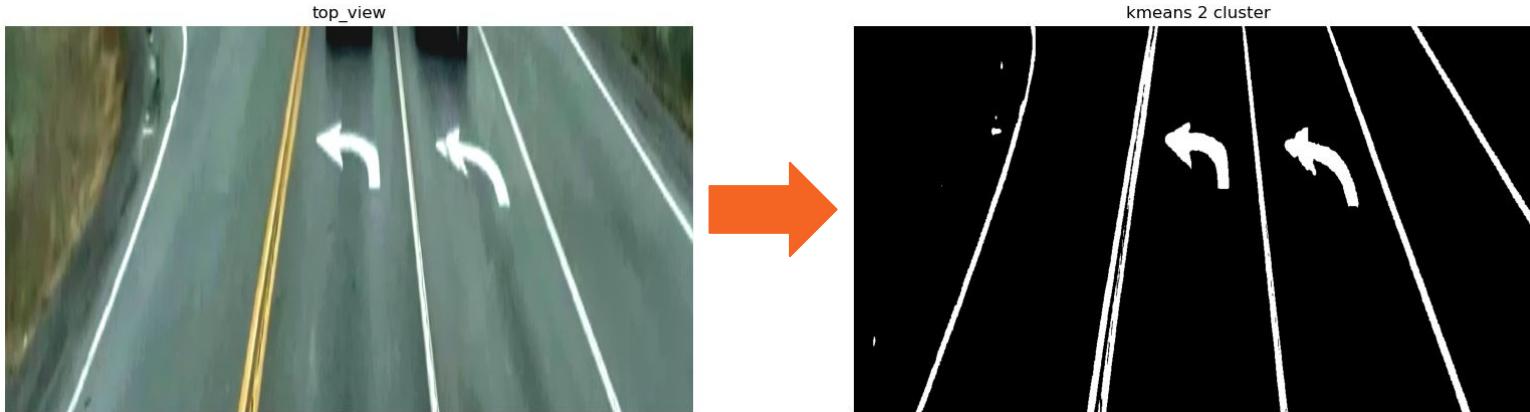


# Additional Test : Road Segmentation - K-means Clustering



# Additional Test : Road Segmentation - K-means Clustering

---



Adv : Good Results in Marks Detection!

Dis\_Adv : High Running Time!

# Additional Test : Traffic Sign Detection

---

There are several traditional image processing techniques :

- Color-based Segmentation —> e.g. Red & yellow Signs
- Shape-based Detection —> Contours and Hough methods

These techniques are good for images with good quality and low noisiness!

# Additional Test : Traffic Sign Detection

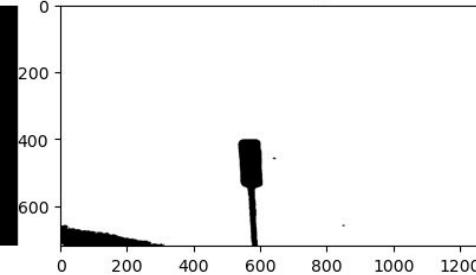
original image



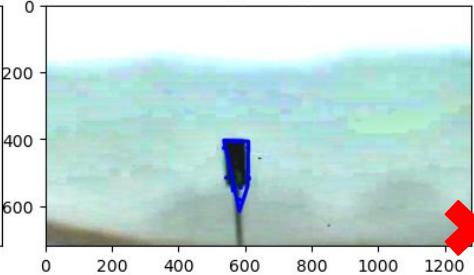
canny method



thresh image



hough image



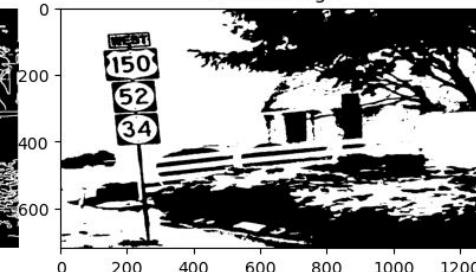
original image



canny method



thresh image



hough image



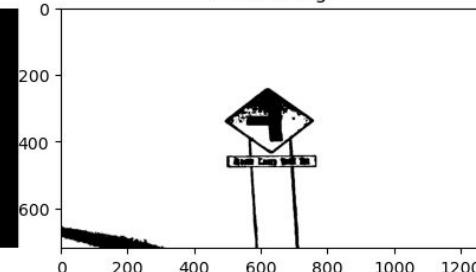
original image



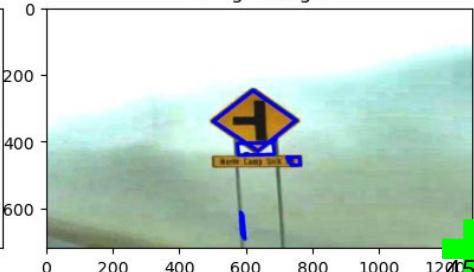
canny method



thresh image



hough image



---

# Compression

# JPEG Compression (80%)

---



PNG FILE

572 KB

JPG FILE

74 KB

---

# Final Results

# Final Result

---



# References

---

- ❖ Meng, G., Wang, Y., Duan, J., Xiang, S., & Pan, C. (2013). Efficient image dehazing with boundary constraint and contextual regularization. *Proceedings of the IEEE international conference on computer vision*, 617-624.
- ❖ Gautam, S., Gandhi, T. K., & Panigrahi, B. K. (2020). An Improved Air-Light Estimation Scheme for Single Haze Images Using Color Constancy Prior. *IEEE Signal Processing Letters*, 27, 1695-1699. doi: 10.1109/LSP.2020.3025462.
- ❖ Raikwar, S. C., & Tapaswi, S. (2020). Lower Bound on Transmission Using Non-Linear Bounding Function in Single Image Dehazing. *IEEE Transactions on Image Processing*, 29, 4832-4847. doi: 10.1109/TIP.2020.2975909.
- ❖ Towards Data Science. (n.d.). Image segmentation with classical computer vision-based approaches. Retrieved from <https://towardsdatascience.com/image-segmentation-with-classical-computer-vision-based-approaches-80c75d6d995f>
- ❖ Korting, T. (n.d.). Region-growing-segmentation. Retrieved from <https://github.com/tkorting/youtube/blob/master/region-growing-segmentation/main.m>
- ❖ Towards Data Science. (n.d.). Image segmentation with clustering. Retrieved from <https://towardsdatascience.com/image-segmentation-with-clustering-b4bbc98f2ee6>
- ❖ Ayca, Y. (n.d.). Image-Segmentation. Retrieved from <https://github.com/YCAyca/Image-Segmentation/tree/main/Clustering>
- ❖ Cao, J., Song, C., Song, S., Xiao, F., & Peng, S. (2019). Lane Detection Algorithm for Intelligent Vehicles in Complex Road Conditions and Dynamic Environments. *Sensors*, 19(14), 3166. doi: 10.3390/s19143166
- ❖ YouTube. (n.d.). Lane Detection for Autonomous Driving: Conventional and CNN approaches. Retrieved from <https://www.youtube.com/watch?v=xIRT3rgWrFQ>