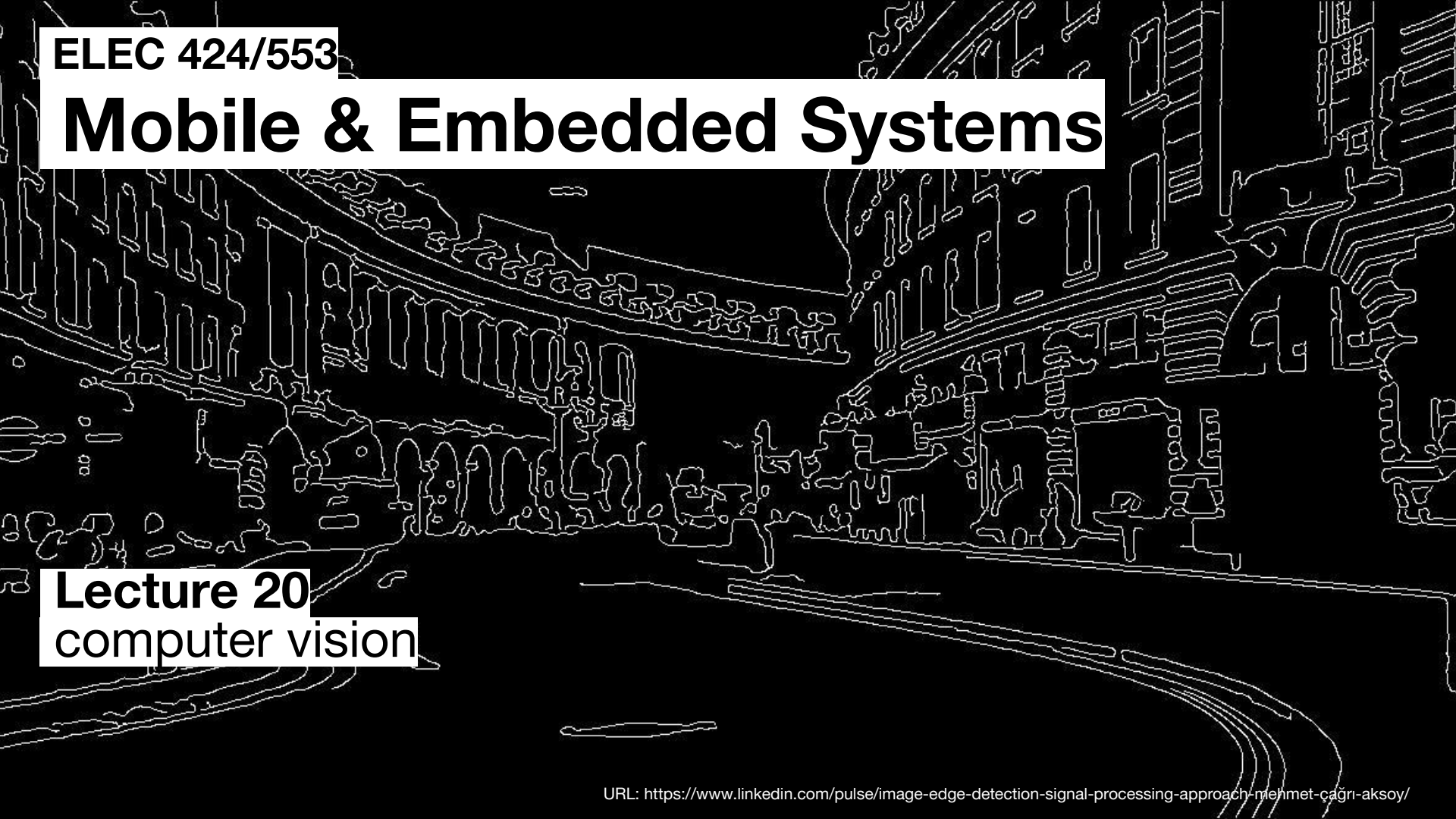
ELEC 424/553

Mobile & Embedded Systems



Lecture 20
computer vision

URL: https://www.linkedin.com/pulse/image-edge-detection-signal-processing-approach-mehmet-çağrı-aksoy/

Getting Computer Vision (CV) To Work On Your BBB

```
sudo apt update
sudo apt upgrade

git clone https://github.com/derekmolloy/boneCV
git clone https://github.com/jayrambhia/Install-OpenCV.git

cd ~/Install-OpenCV/Ubuntu/
./opencv_latest.sh

sudo apt install libv4l-dev

cd ~/boneCV
./build

sudo apt install ffmpeg

sudo apt install python3-opencv
```

Follow along at

https://docs.google.com/document/d/13MDKEBBn4ciiTB9MOuKHkvzUFOXhZ5aDKpfSqapMl9g/edit#

Housekeeping

- **Project 3 due Friday @ 4pm**
- **The one and only exam will be during the last week of classes**
  - There will be a recorded in-person review session led by me
- **Final project due on the last day of final exams**
  - Team project (3-4 people)
  - Requires Python
    - I will host a help session on Python if requested
  - Graduate students will have extra requirement(s)

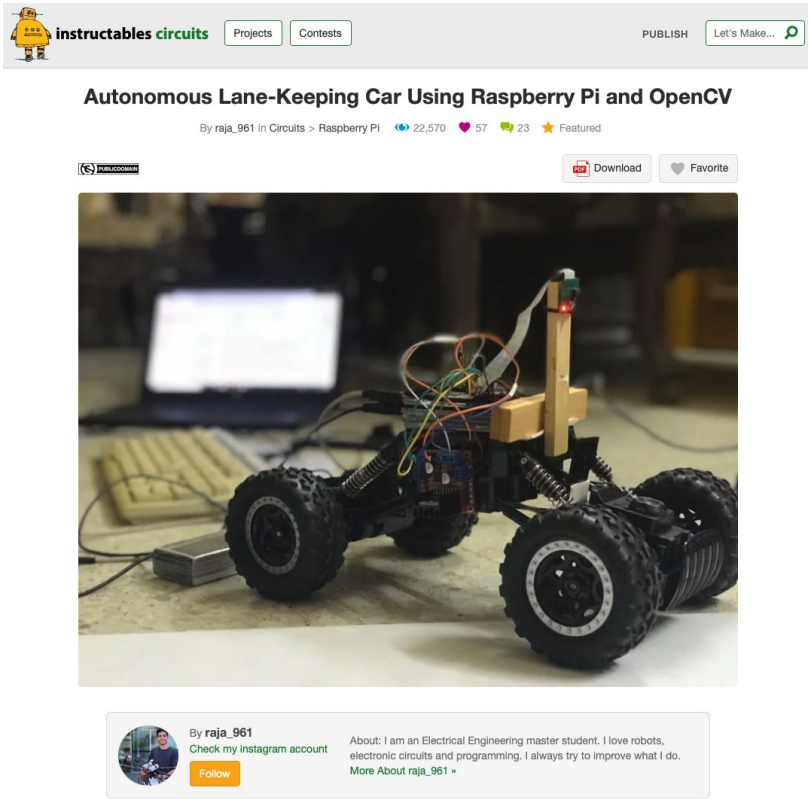
Housekeeping

- **Midterm** posted
  - Due Monday November 13th at 11:59pm
- Lecture schedule
  - **Mon Nov 13**: No lecture - I'll be on zoom for any midterm clarification questions
  - **Wed Nov 15**: No lecture - I am going to Denmark!
  - **Mon Nov 20**: Zoom lecture [exercise 14]
  - ***Wed Nov 22****: No lecture - Thanksgiving (food overflow)*
  - **Mon Nov 27**: No lecture - Office hours for final project
  - **Wed Nov 29**: No lecture - Office hours for final project

Housekeeping (continued)

- **Project 3**
  - Teams announced
  - Motor control
  - Goodies will be given
- **Final Project**
  - Autonomous lane-keeping RC car
  - Due Fri Dec 1 (last day of classes)
- **Assignment 3** (optional)
  - Rust
  - Replace assignment or project grade of your choosing (not final project though)
  - TBA on posting

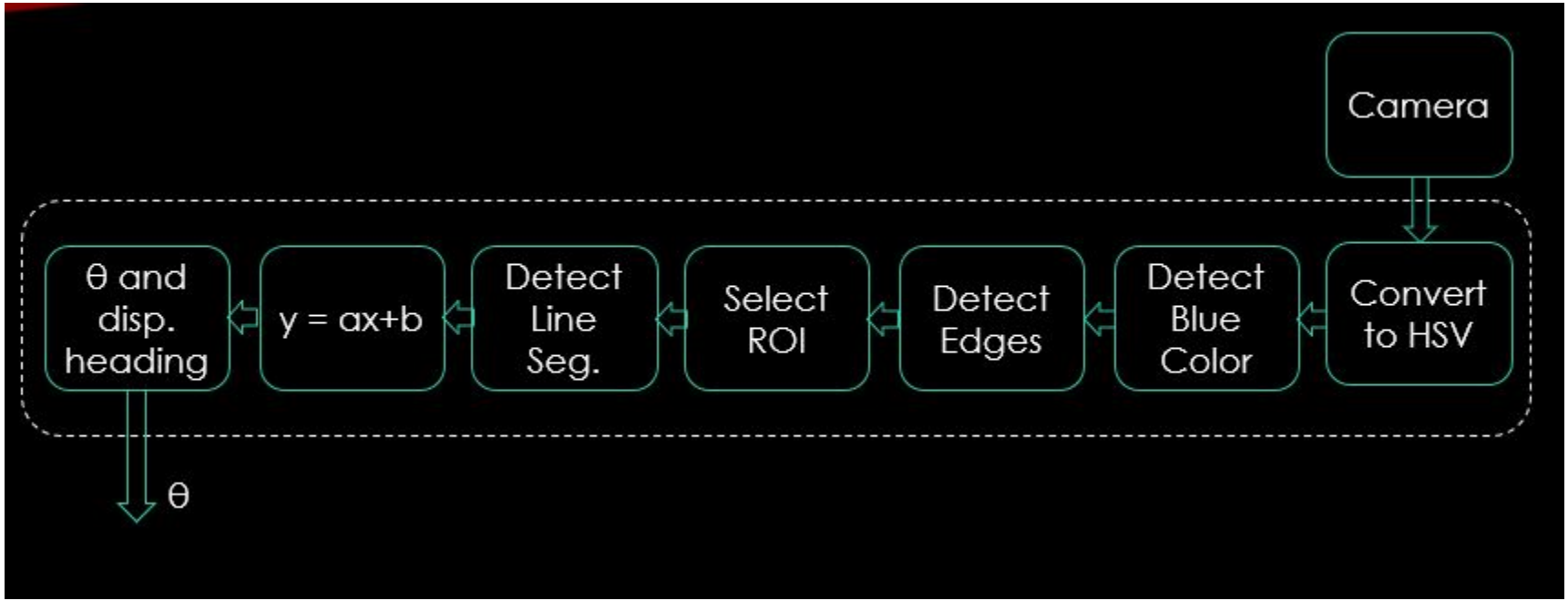
Making Our Own Computer Vision Application



https://www.instructables.com/Autonomous-Lane-Keeping-Car-Using-Raspberry-Pi-and/

User raja_961, “Autonomous Lane-Keeping Car Using Raspberry Pi and OpenCV”. https://www.instructables.com/Autonomous-Lane-Keeping-Car-Using-Raspberry-Pi-and/

Pipeline of Image Processing From Instructable

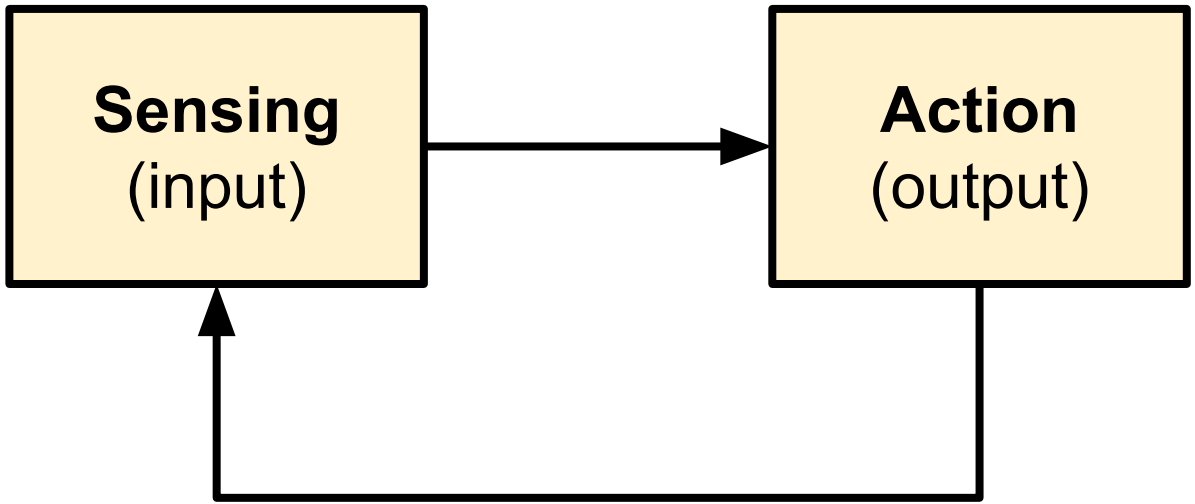


User raja_961, “Autonomous Lane-Keeping Car Using Raspberry Pi and OpenCV”. https://www.instructables.com/Autonomous-Lane-Keeping-Car-Using-Raspberry-Pi-and/

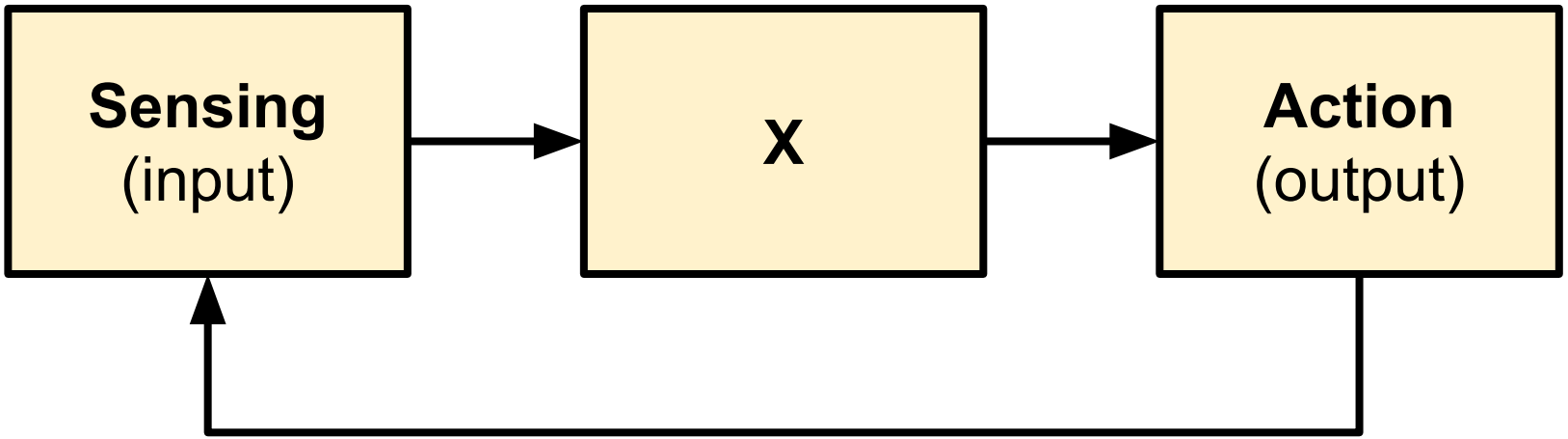
What Do We Call This Approach

- Pipeline approach
- Guess who uses pipeline approaches?
- Tesla
- Likely ever other major player apart from Comma AI

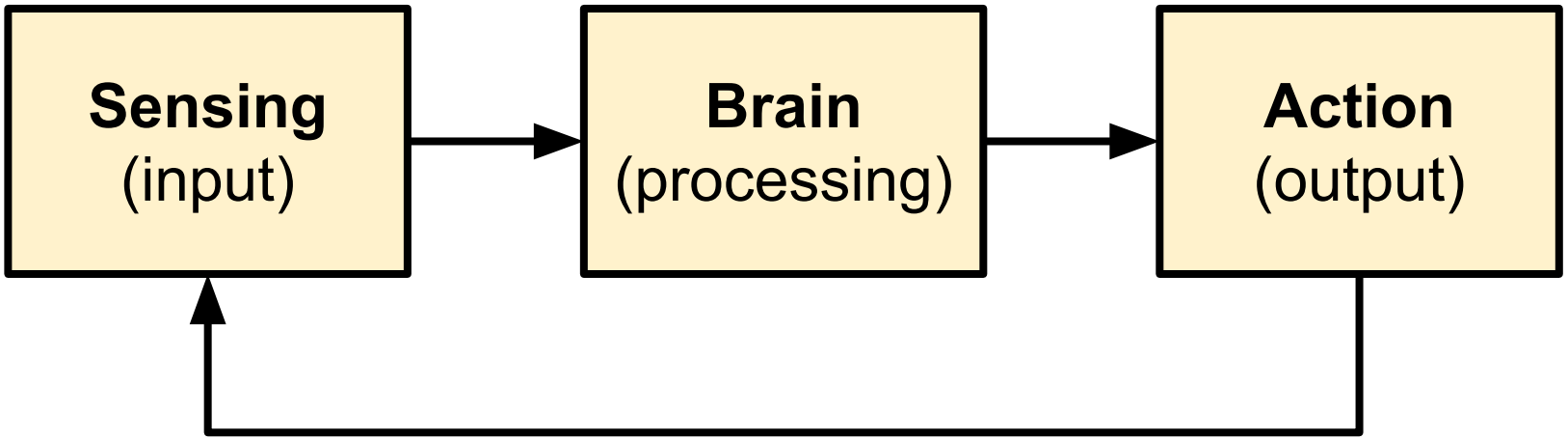
Life - What's It All About?



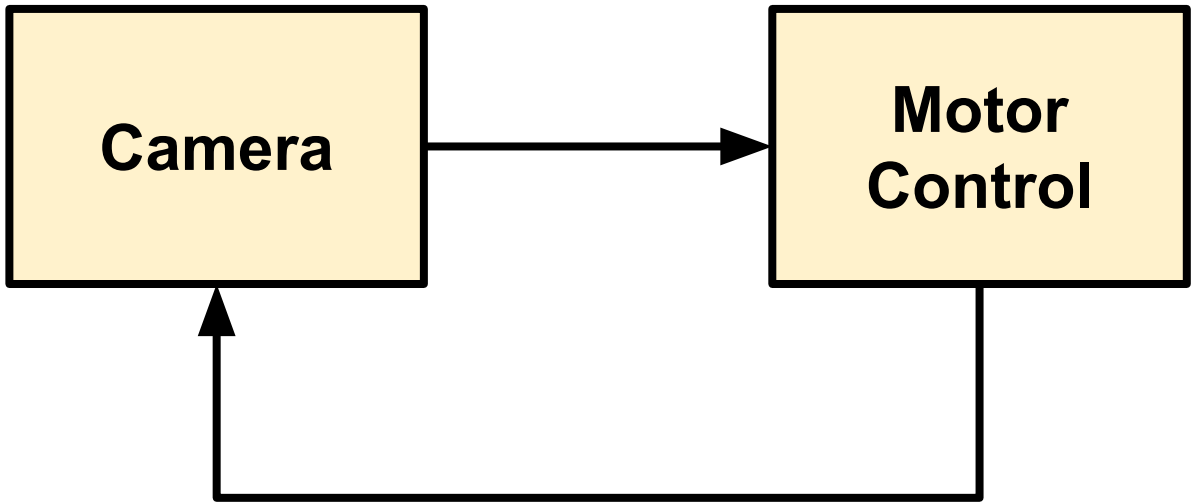
Humans (Sometimes) Use A Little More Than This



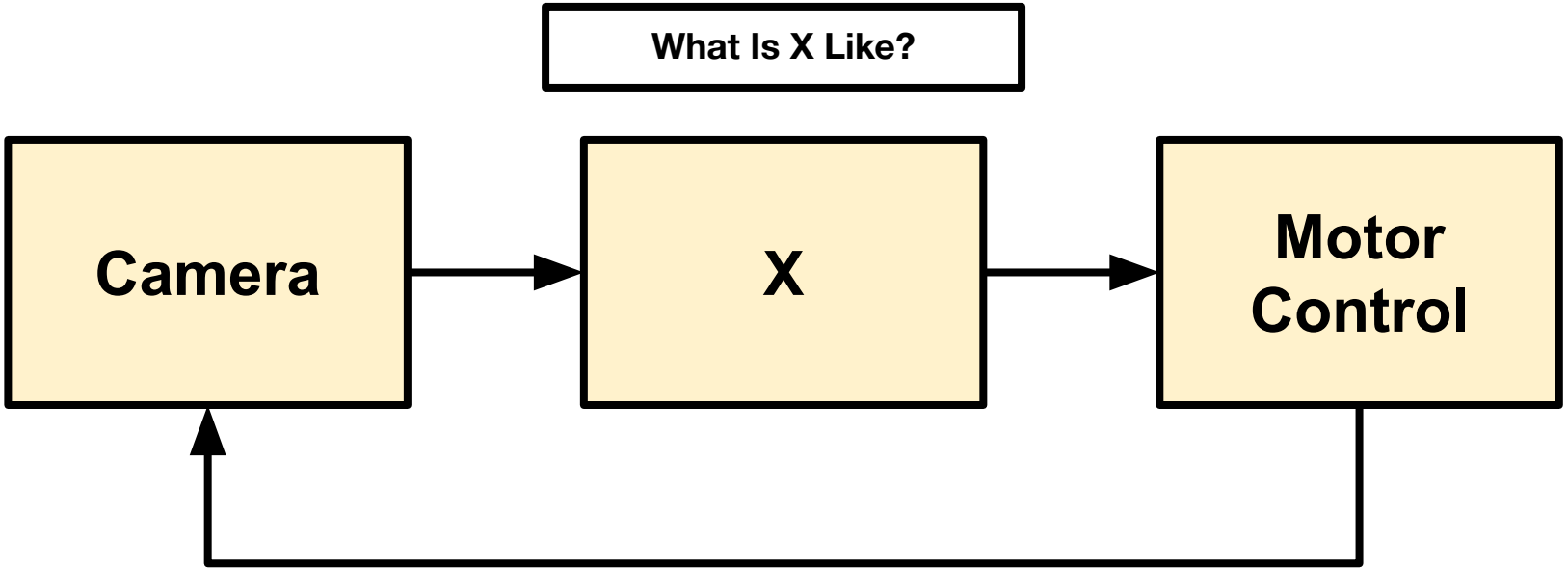
Humans (Sometimes) Use A Little More Than This



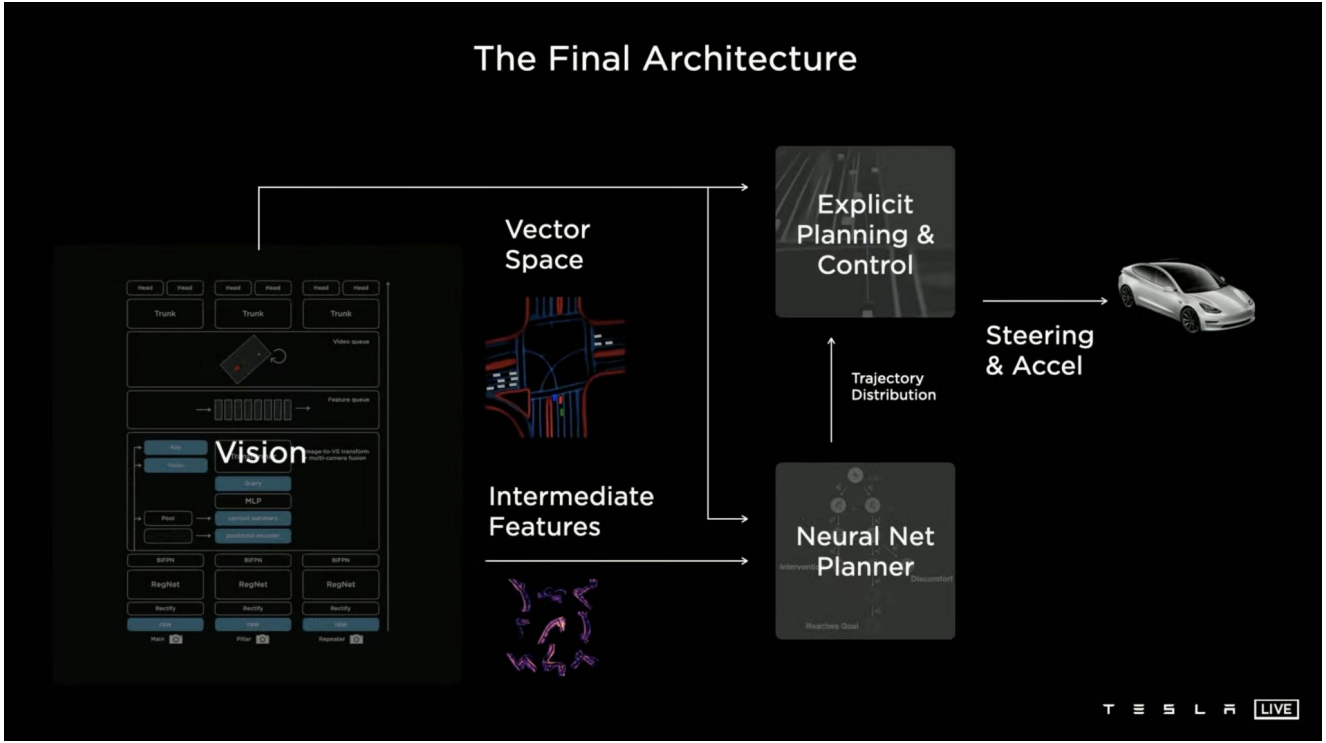
Autonomous Vehicle: What's The Fundamental Goal?



We Need Something More

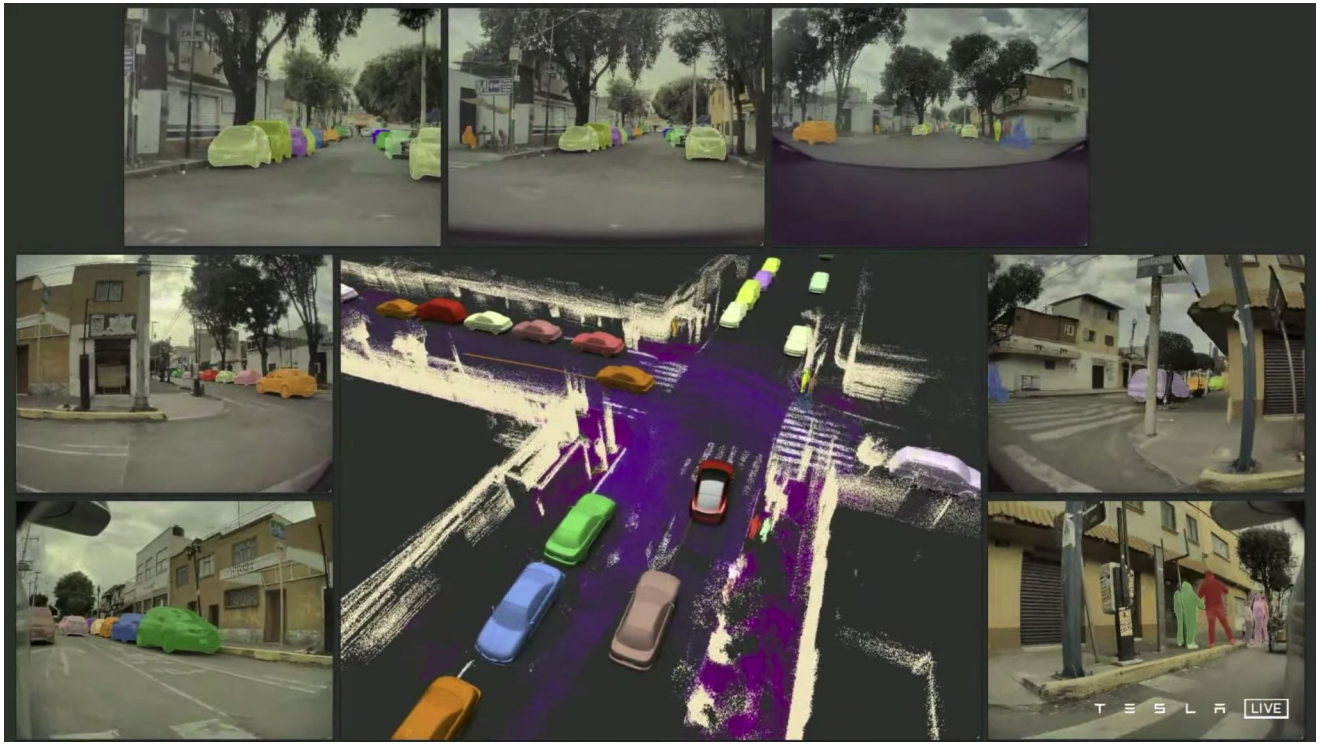


Tesla's X



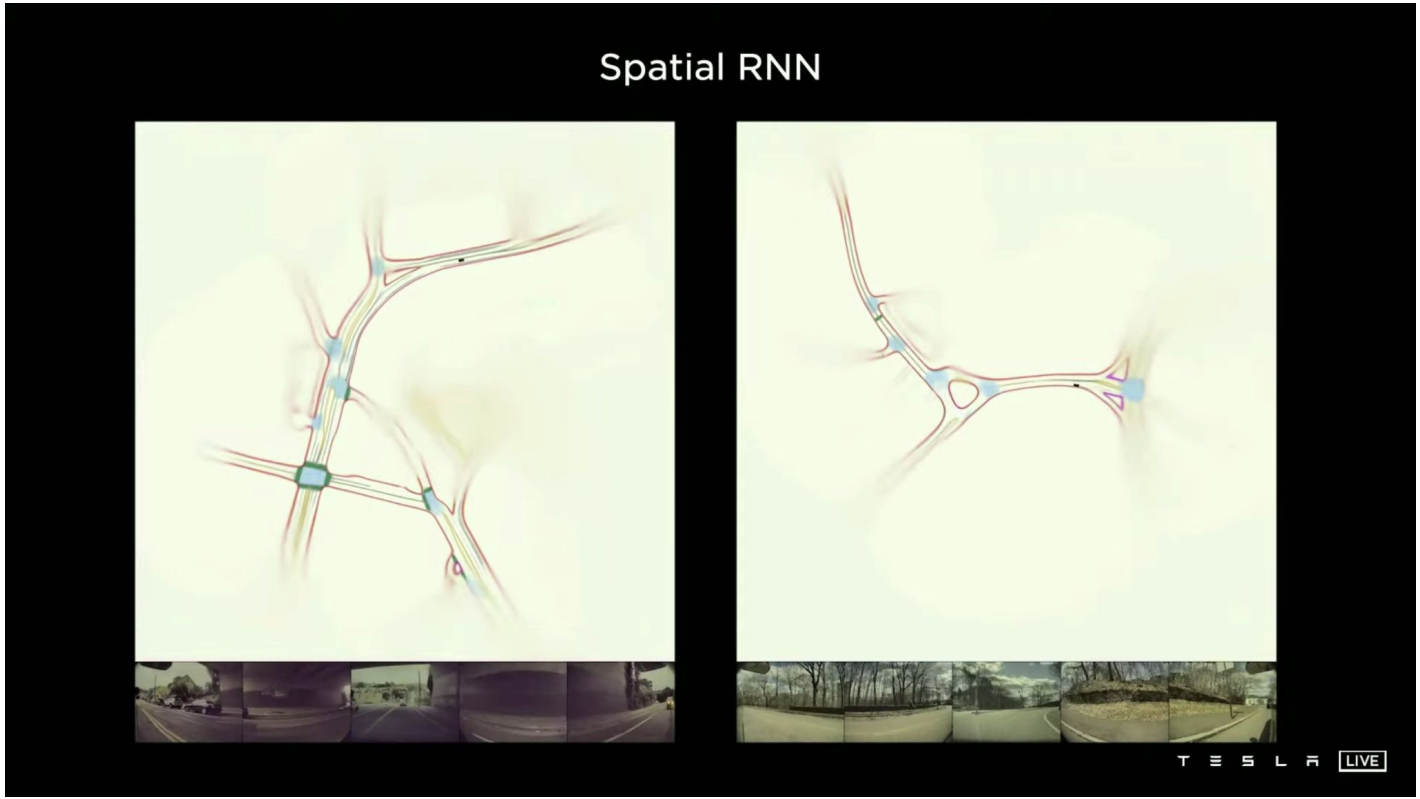
From Tesla's AI Day event: "LIVE Blog: Tesla AI Day". TESLARATI. URL: https://www.teslarati.com/tesla-ai-day-live-blog/

Tesla's X



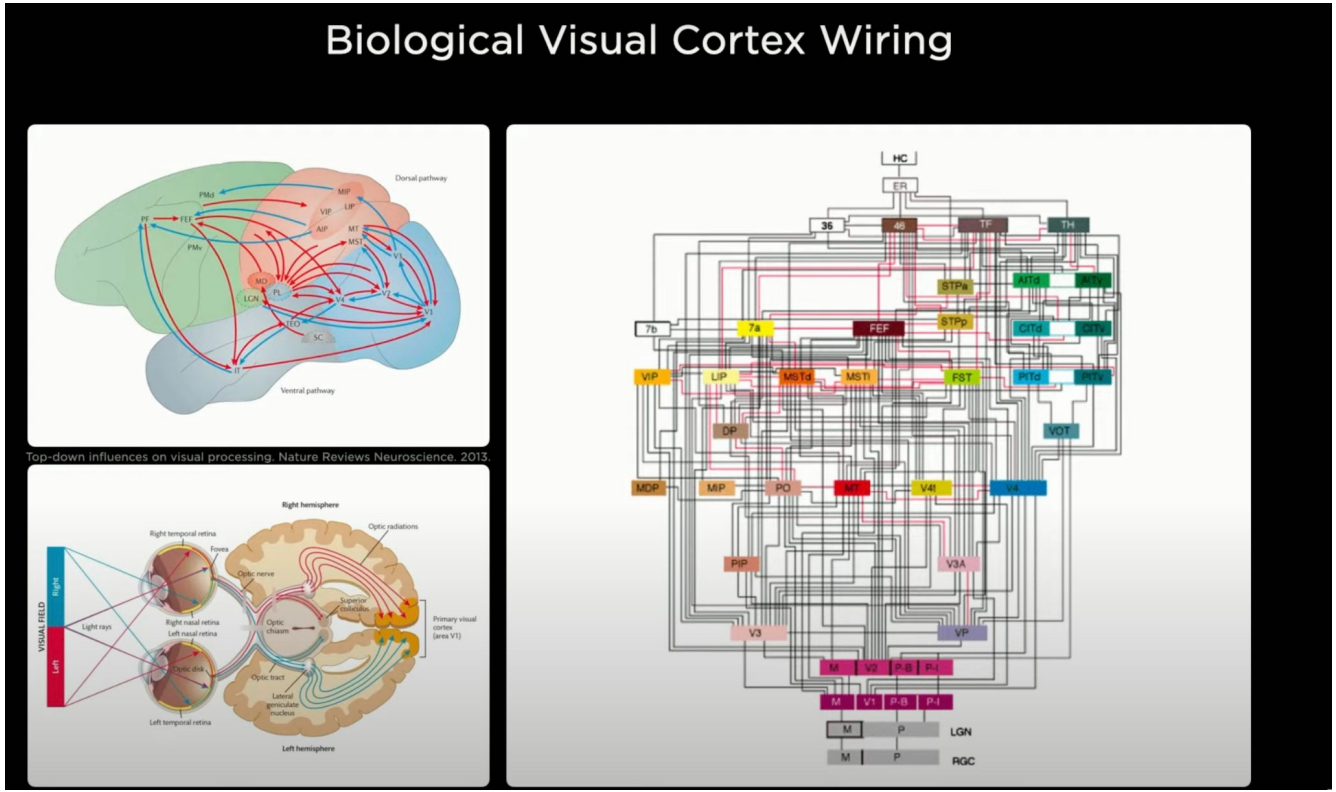
From Tesla's AI Day event: Teslascope thread on Twitter. URL:
https://twitter.com/teslascope/status/1428524117741871113?ref_src=twsrc%5Etfw%7Ctwcamp%5Etweetembed%7Ctwterm%5E1428524117741871113%7Ctwgr%5E%7Ctwcon%5Es1_&ref_url=https%3A%2F%2Fwww.teslarati.com%2Ftesla-ai-day-live-blog%2F

Tesla's X



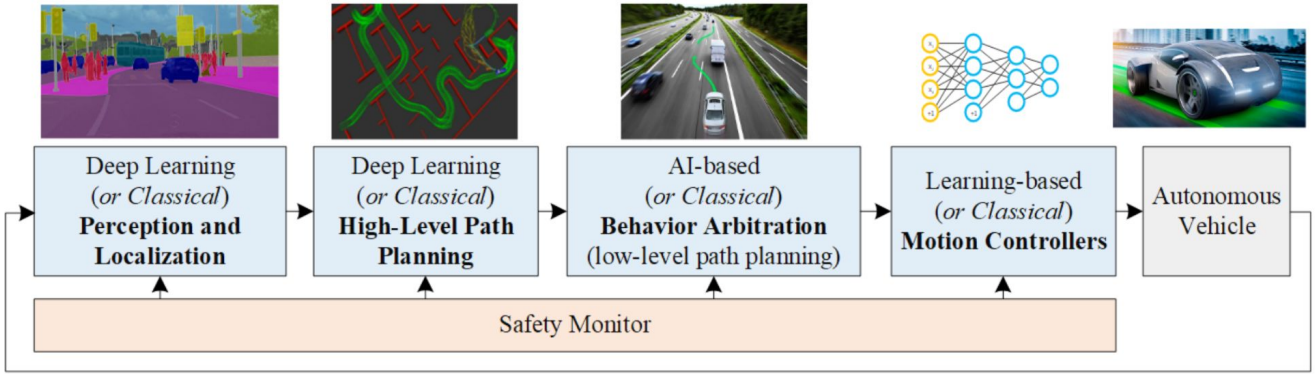
From Tesla's AI Day event: Teslascope thread on Twitter. URL:
https://twitter.com/teslascope/status/1428524117741871113?ref_src=twsrc%5Etfw%7Ctwcamp%5Etweetembed%7Ctwterm%5E1428524117741871113%7Ctwgr%5E%7Ctwcon%5Es1_&ref_url=https%3A%2F%2Fwww.teslarati.com%2Ftesla-ai-day-live-blog%2F

Tesla's X



From Tesla's AI Day event: "LIVE Blog: Tesla AI Day". TESLARATI. URL: https://www.teslarati.com/tesla-ai-day-live-blog/

Tesla's X



Grigorescu et al. "A Survey of Deep Learning Techniques for Autonomous Driving". arXiv. (2020)

Comma AI's X

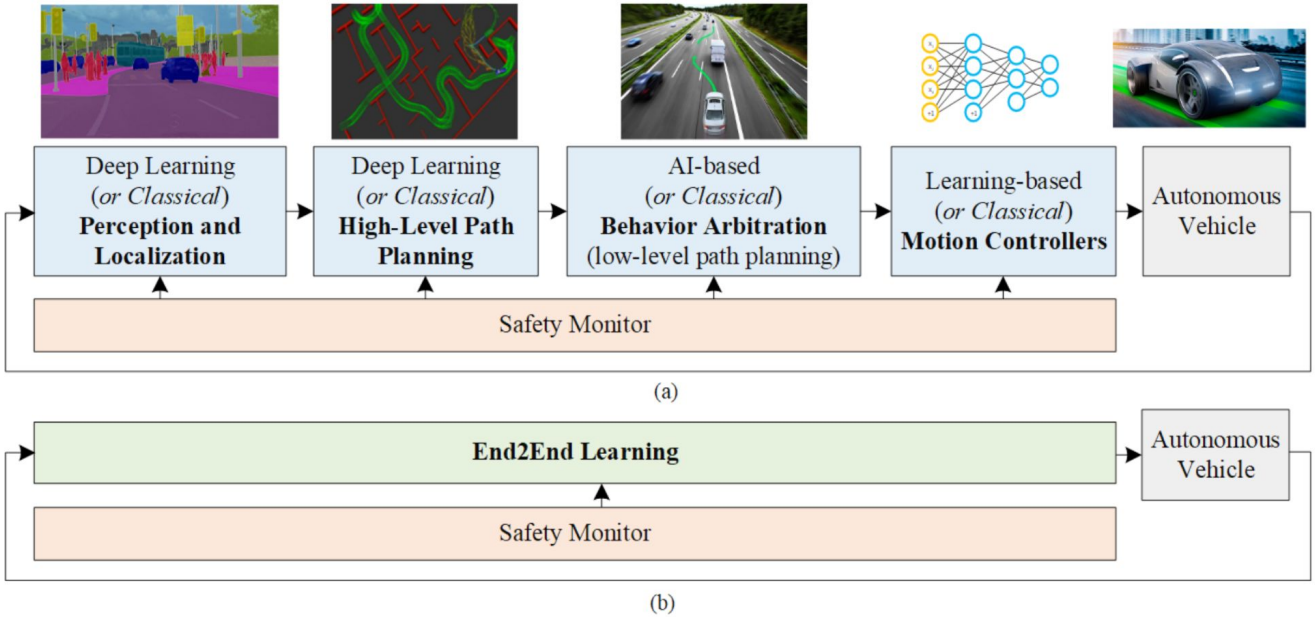
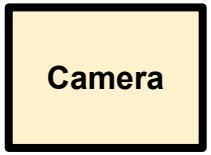
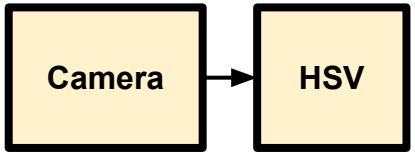


Figure 1: **Deep Learning based self-driving car**. The architecture can be implemented either as a sequential perception-planing-action pipeline (a), or as an End2End system (b). In the sequential pipeline case, the components can be designed either using AI and deep learning methodologies, or based on classical non-learning approaches. End2End learning systems are mainly based on deep learning methods. A safety monitor is usually designed to ensure the safety of each module.

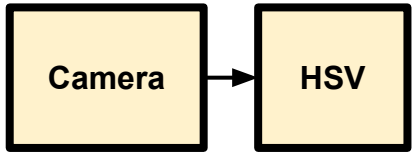
Grigorescu et al. "A Survey of Deep Learning Techniques for Autonomous Driving". arXiv. (2020)

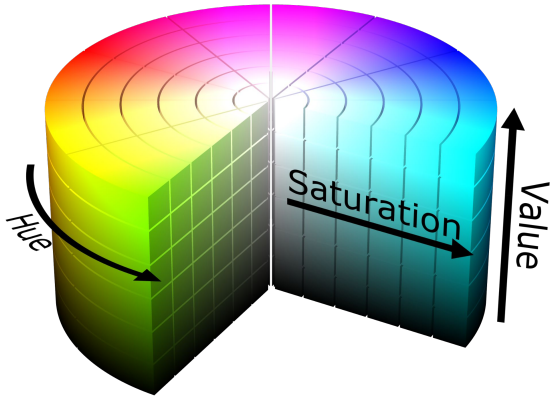
Our X

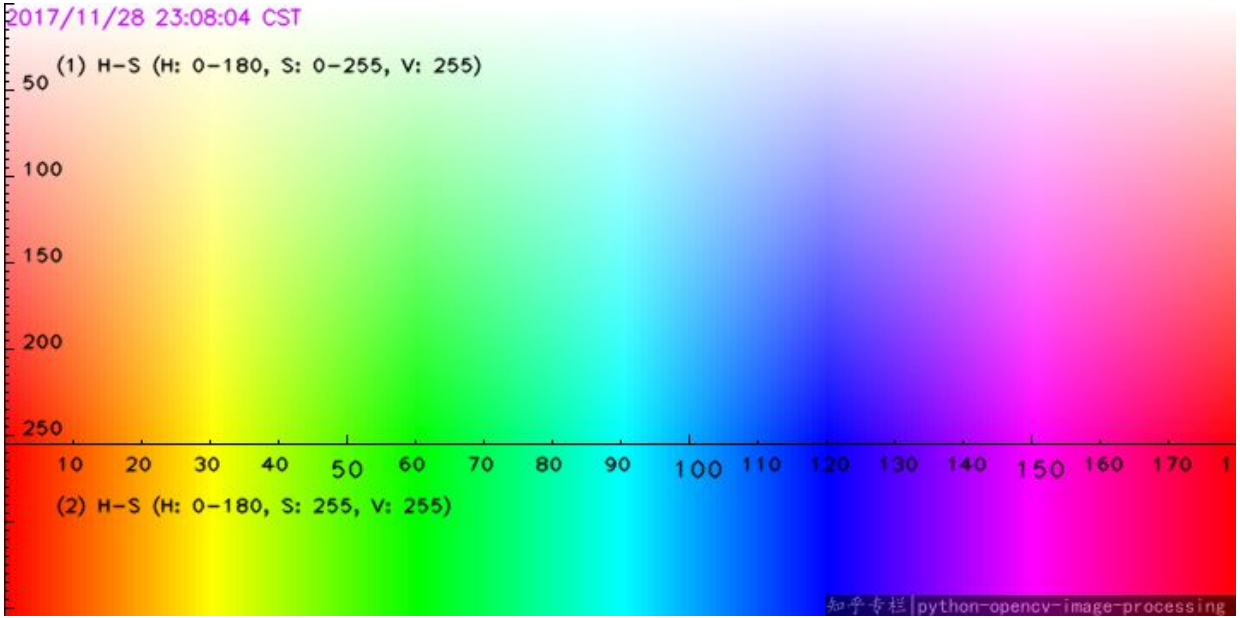

Camera


Camera
HSV

HSV

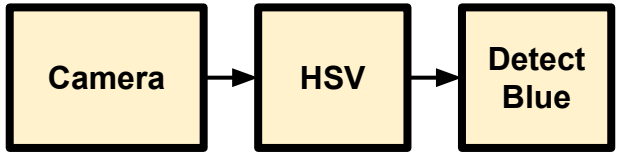


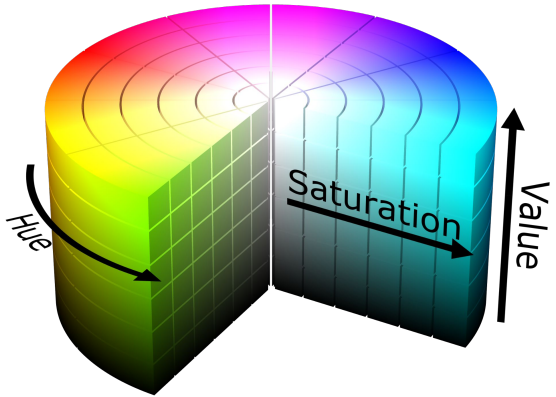


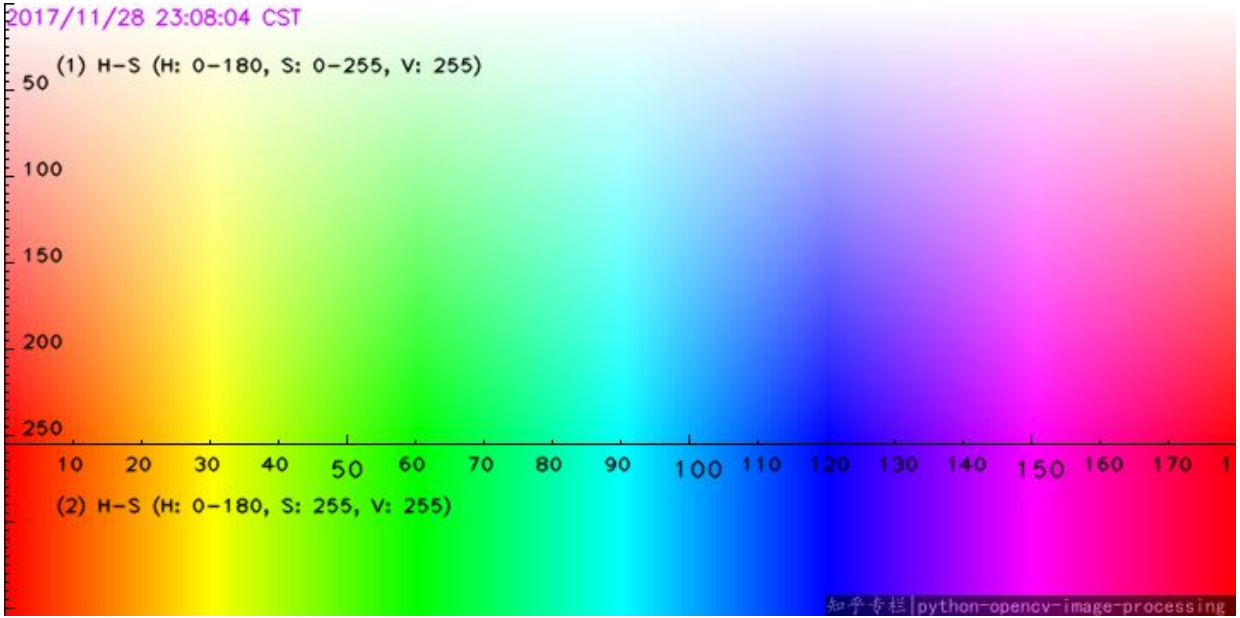


https://stackoverflow.com/questions/47483951/how-to-define-a-threshold-value-to-detect-only-green-colour-objects-in-an-image/47483966#47483966

Blue



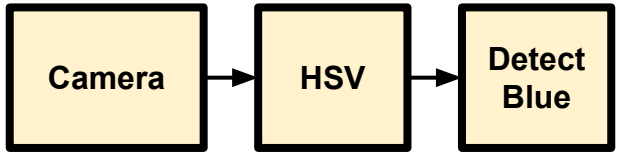




https://stackoverflow.com/questions/47483951/how-to-define-a-threshold-value-to-detect-only-green-colour-objects-in-an-image/47483966#47483966

User raja_961, “Autonomous Lane-Keeping Car Using Raspberry Pi and OpenCV”. https://www.instructables.com/Autonomous-Lane-Keeping-Car-Using-Raspberry-Pi-and/

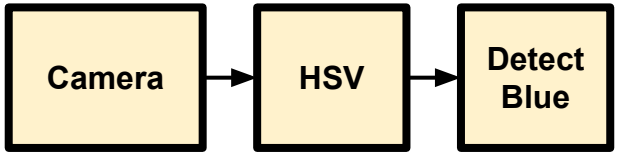
Blue



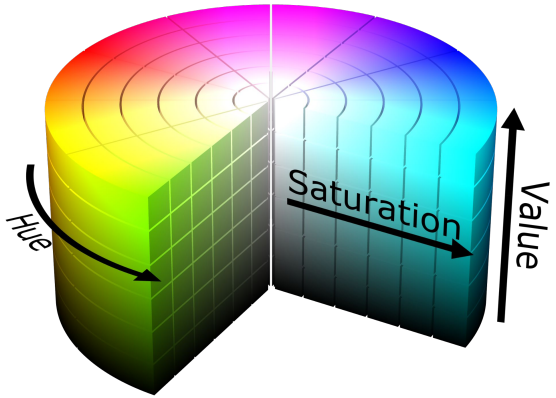

```
46  def detect_edges(frame):
47      # filter for blue lane lines
48      hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
49      #cv2.imshow("HSV",hsv)
50      lower_blue = np.array([90, 120, 0], dtype = "uint8")
51      upper_blue = np.array([150, 255, 255], dtype="uint8")
52      mask = cv2.inRange(hsv,lower_blue,upper_blue)
```

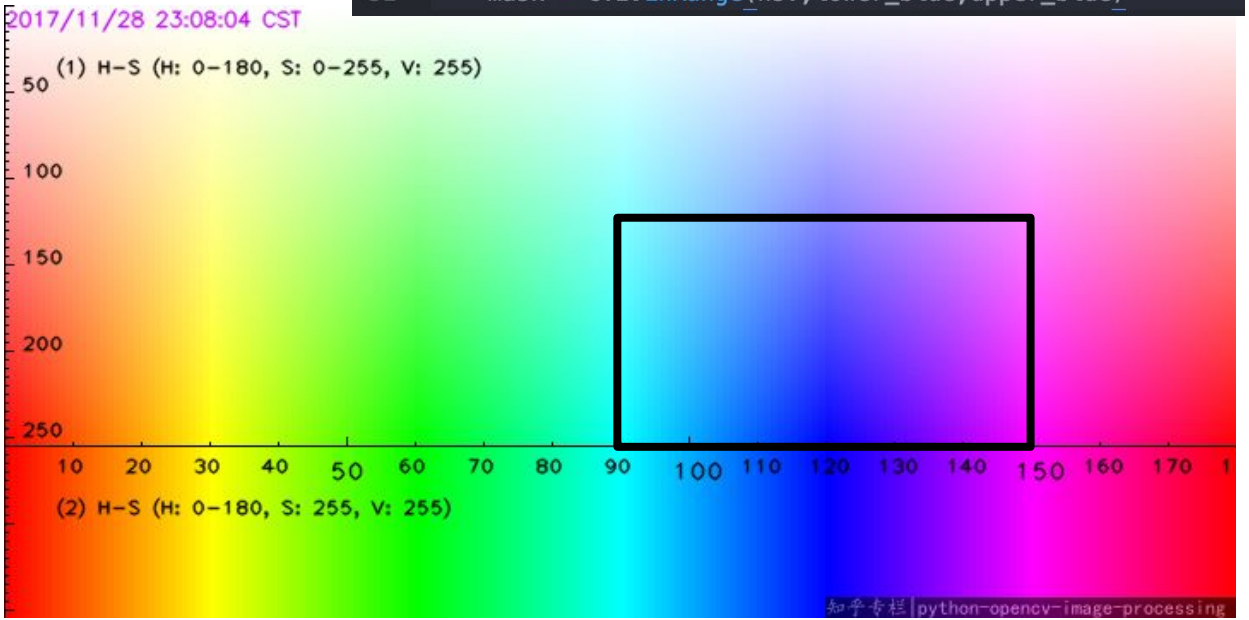

User raja_961, “Autonomous Lane-Keeping Car Using Raspberry Pi and OpenCV”. https://www.instructables.com/Autonomous-Lane-Keeping-Car-Using-Raspberry-Pi-and/

Blue




```python
def detect_edges(frame):
    # filter for blue lane lines
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    #cv2.imshow("HSV",hsv)
    lower_blue = np.array([90, 120, 0], dtype = "uint8")
    upper_blue = np.array([150, 255, 255], dtype="uint8")
    mask = cv2.inRange(hsv,lower_blue,upper_blue)
```

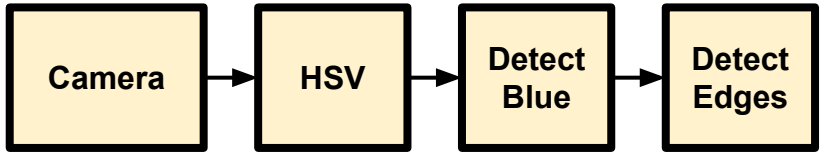




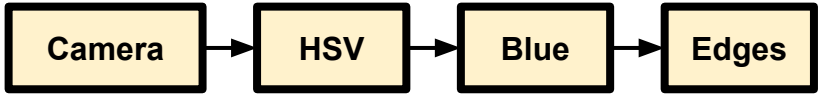
https://stackoverflow.com/questions/47483951/how-to-define-a-threshold-value-to-detect-only-green-colour-objects-in-an-image/47483966#47483966

User raja_961, “Autonomous Lane-Keeping Car Using Raspberry Pi and OpenCV”. https://www.instructables.com/Autonomous-Lane-Keeping-Car-Using-Raspberry-Pi-and/

Edges

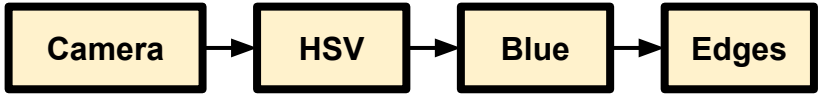


User raja_961, “Autonomous Lane-Keeping Car Using Raspberry Pi and OpenCV”. https://www.instructables.com/Autonomous-Lane-Keeping-Car-Using-Raspberry-Pi-and/

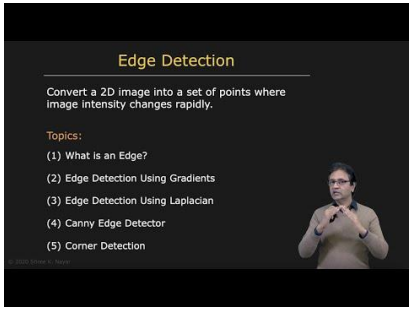


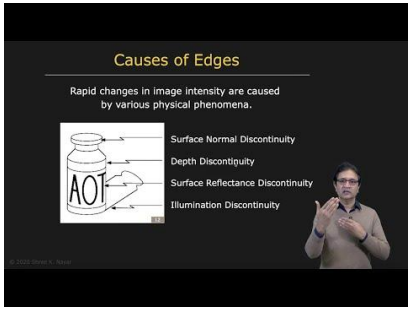
Reposition Figure…

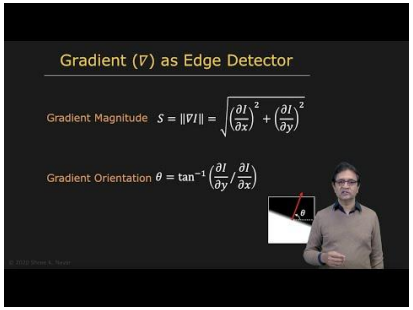
User raja_961, “Autonomous Lane-Keeping Car Using Raspberry Pi and OpenCV”. https://www.instructables.com/Autonomous-Lane-Keeping-Car-Using-Raspberry-Pi-and/

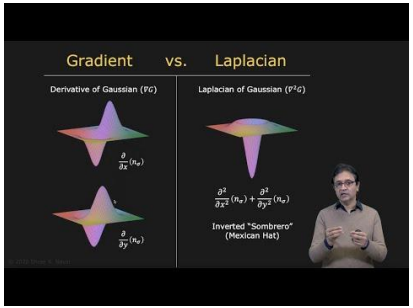


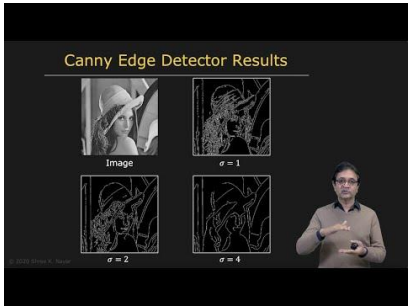
The Following Slides On CV Are Inspired By These Lectures



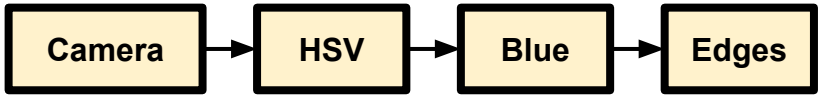






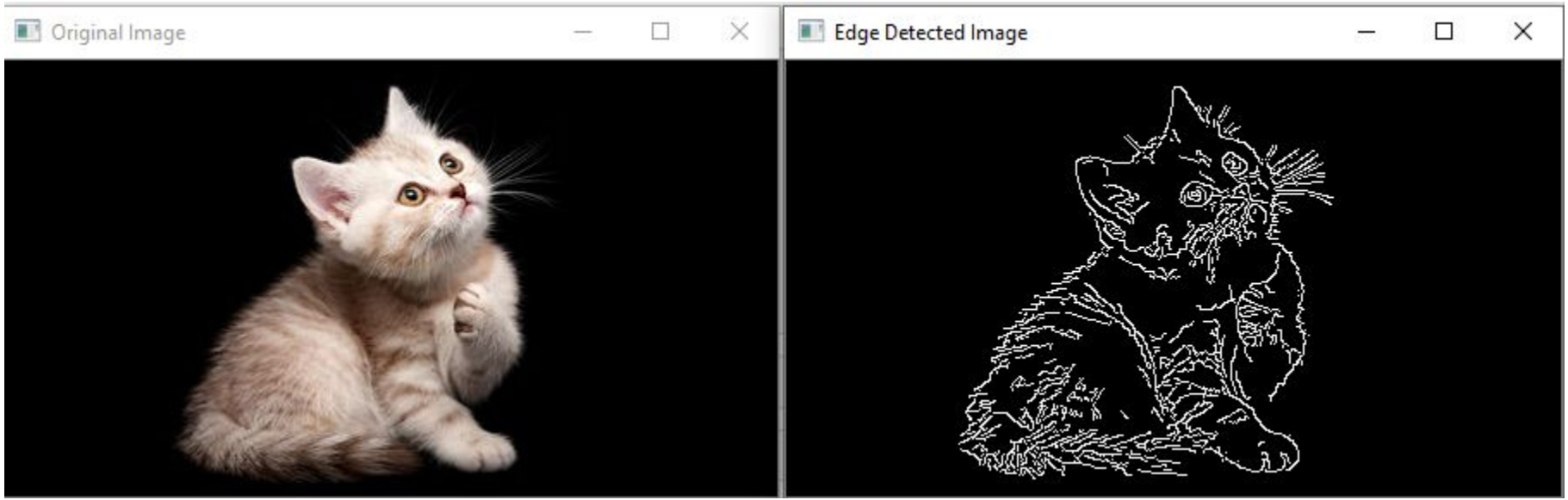


YouTube Channel: [First Principles of Computer Vision]

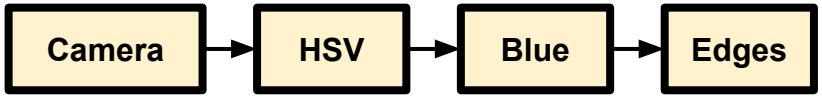


Goal of Edge Detection

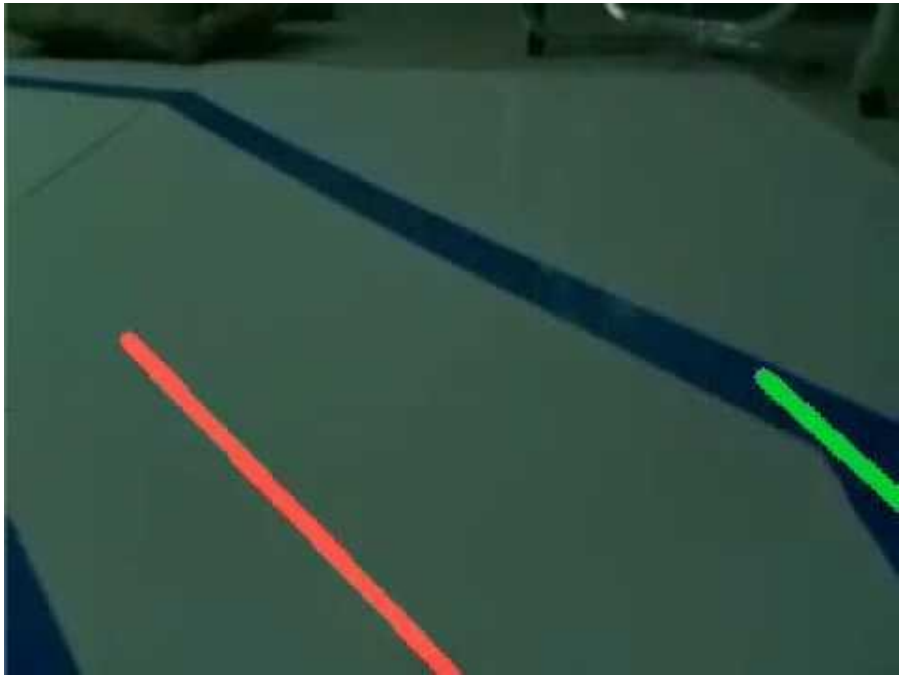
What is an edge?



"OpenCV Canny Edge Detection". URL: https://www.javatpoint.com/opencv-canny-edge-detection



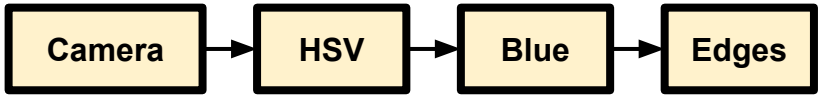
Goal of Edge Detection (2)



https://www.instructables.com/Autonomous-Lane-Keeping-Car-Using-Raspberry-Pi-and/

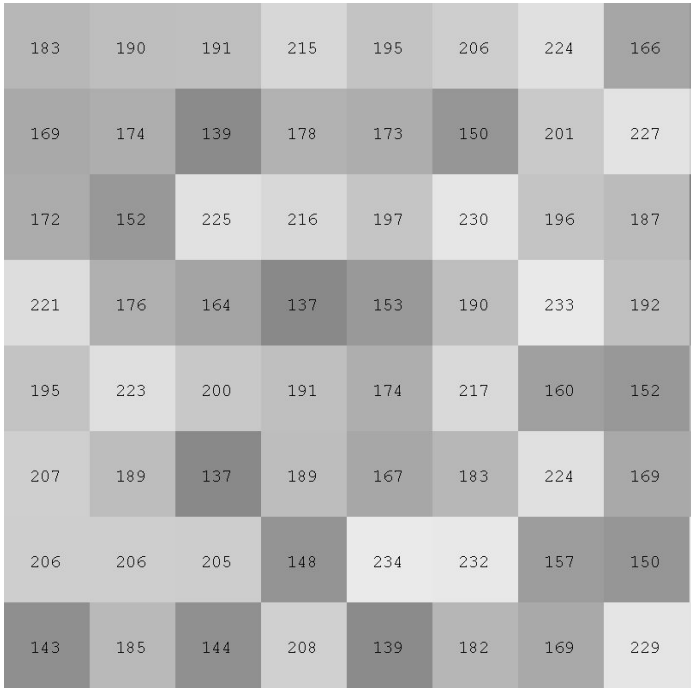
Note: OpenCV & waitKey

- You often (always?) have to add cv2.waitKey(1) in your loop if you are continually showing images using openCV
- Otherwise you may not get any image window to appear
    - I had this issue on my Mac
- More info - MLK, URL: https://machinelearningknowledge.ai/opencv-cv2-waitkey-tutorial-with-examples/



Edge Detection Specifics

- Often done with black and white (BW) images
- Looking for:
  - Sharp contrast
  - Sharp changes in pixel intensity
  - High frequency activity
- Consider that a BW image is just a matrix of pixel intensities/brightness values

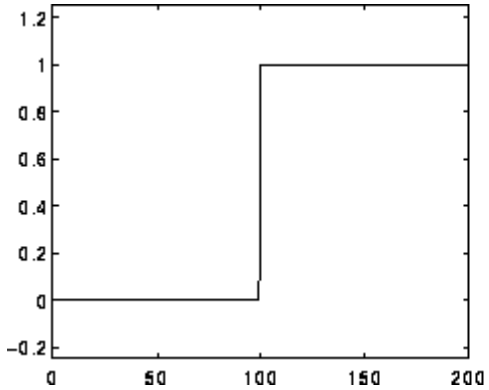


8-bit image: Range of 0 to 255

https://medium.com/@damiandn/an-introduction-to-biological-image-processing-in-imagej-part-1-what-is-an-image-54fc31f3d02d

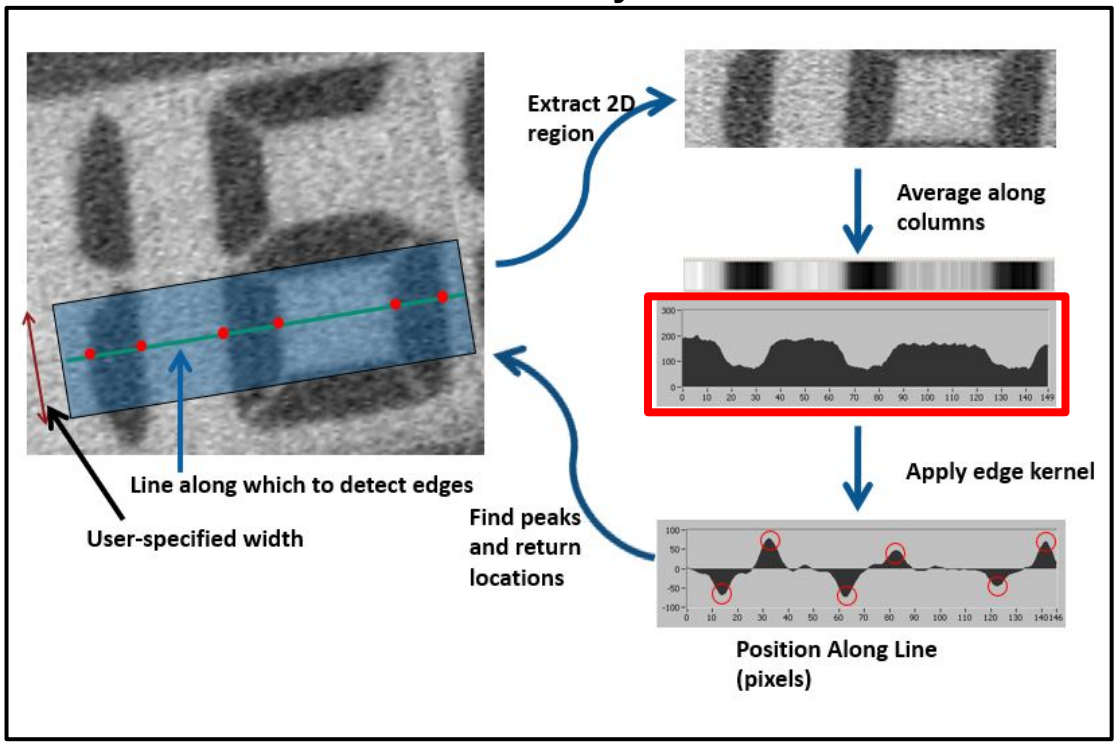
What Do Actual Edges Look Like?

Ideal



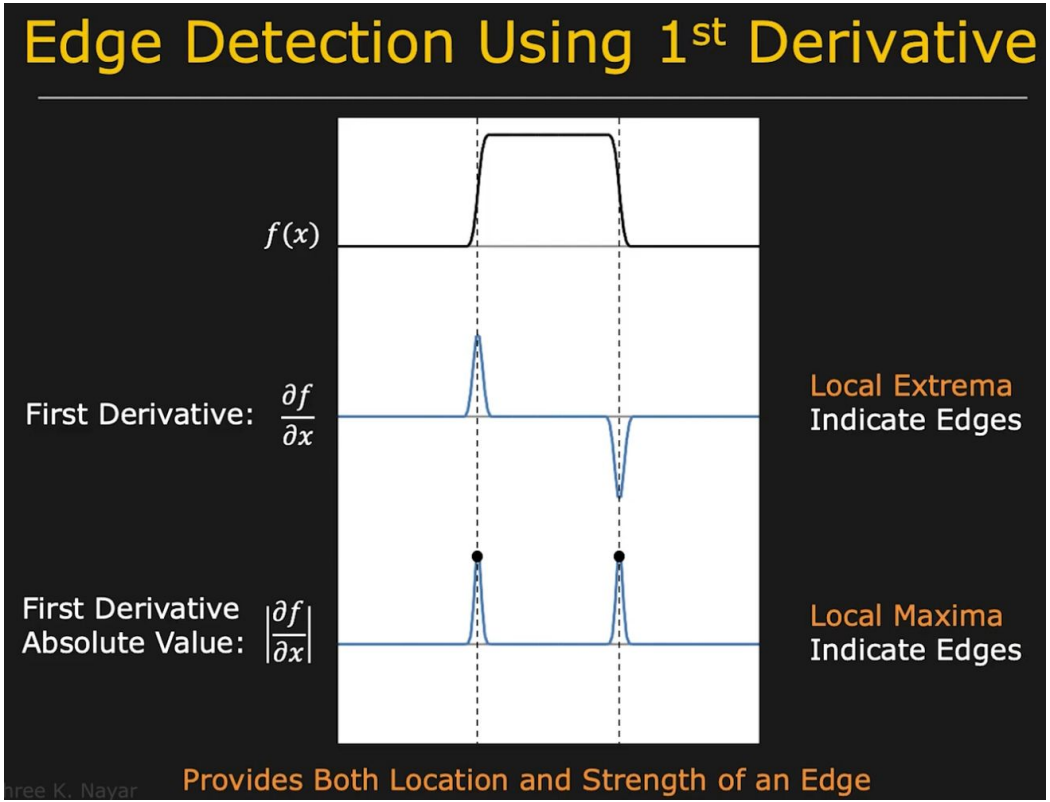
"Zero Crossing Detector". URL: https://homepages.inf.ed.ac.uk/rbf/HIPR2/zeros.htm

Reality



"Which Kernel Does the IMAQ Edge Tool 3 Use?". URL: https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000015CV0SAM&l=en-US

Gradients & Edge Detection



Slide from “Edge Detection Using Gradients | Edge Detection”. First Principles of Computer Vision. Shree Nayar. URL: https://www.youtube.com/watch?v=lOEBsQodtEQ

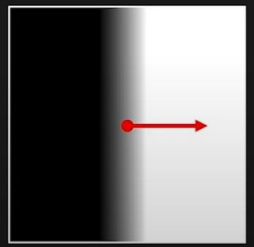
Gradients & Edge Detection (2)

Gradient (∇)

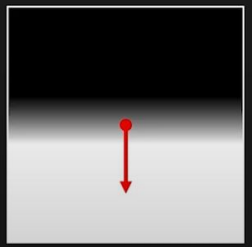
Gradient (Partial Derivatives) represents the direction of most rapid change in intensity

$$\nabla I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right]$$

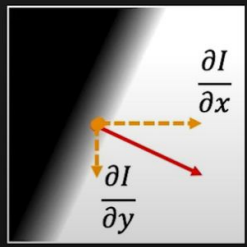
Pronounced as "Del I"



$$\nabla I = \left[\frac{\partial I}{\partial x}, 0\right]$$



$$\nabla I = \left[0, \frac{\partial I}{\partial y}\right]$$



$$\nabla I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right]$$

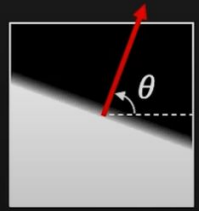
Slide from "Edge Detection Using Gradients | Edge Detection". First Principles of Computer Vision. Shree Nayar. URL: https://www.youtube.com/watch?v=lOEBsQodtEQ

Gradients & Edge Detection (3)

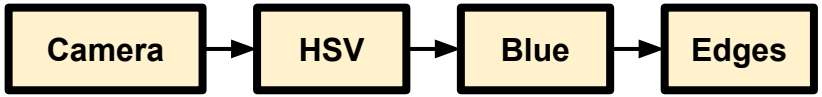
Gradient (∇) as Edge Detector

Gradient Magnitude $S = \|\nabla I\| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$

Gradient Orientation $\theta = \tan^{-1}\left(\frac{\partial I}{\partial y} / \frac{\partial I}{\partial x}\right)$



Slide from “Edge Detection Using Gradients | Edge Detection”. First Principles of Computer Vision. Shree Nayar. URL: https://www.youtube.com/watch?v=lOEBsQodtEQ



Gradients & Edge Detection (4)

Discrete Gradient (∇) Operator

Finite difference approximations:

$$\frac{\partial I}{\partial x} \approx \frac{1}{2\varepsilon}\left(\left(I_{i+1,j+1} - I_{i,j+1}\right) + \left(I_{i+1,j} - I_{i,j}\right)\right)$$

$$\frac{\partial I}{\partial y} \approx \frac{1}{2\varepsilon}\left(\left(I_{i+1,j+1} - I_{i+1,j}\right) + \left(I_{i,j+1} - I_{i,j}\right)\right)$$

$I_{i,j+1}$	$I_{i+1,j+1}$
$I_{i,j}$	$I_{i+1,j}$

ε

Can be implemented as Convolution!

$$\frac{\partial}{\partial x} \approx \frac{1}{2\varepsilon}\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \qquad \frac{\partial}{\partial y} \approx \frac{1}{2\varepsilon}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

Note: Convolution flips have been applied

Slide from “Edge Detection Using Gradients | Edge Detection”. First Principles of Computer Vision. Shree Nayar. URL: https://www.youtube.com/watch?v=lOEBsQodtEQ

Gradients & Edge Detection (5)

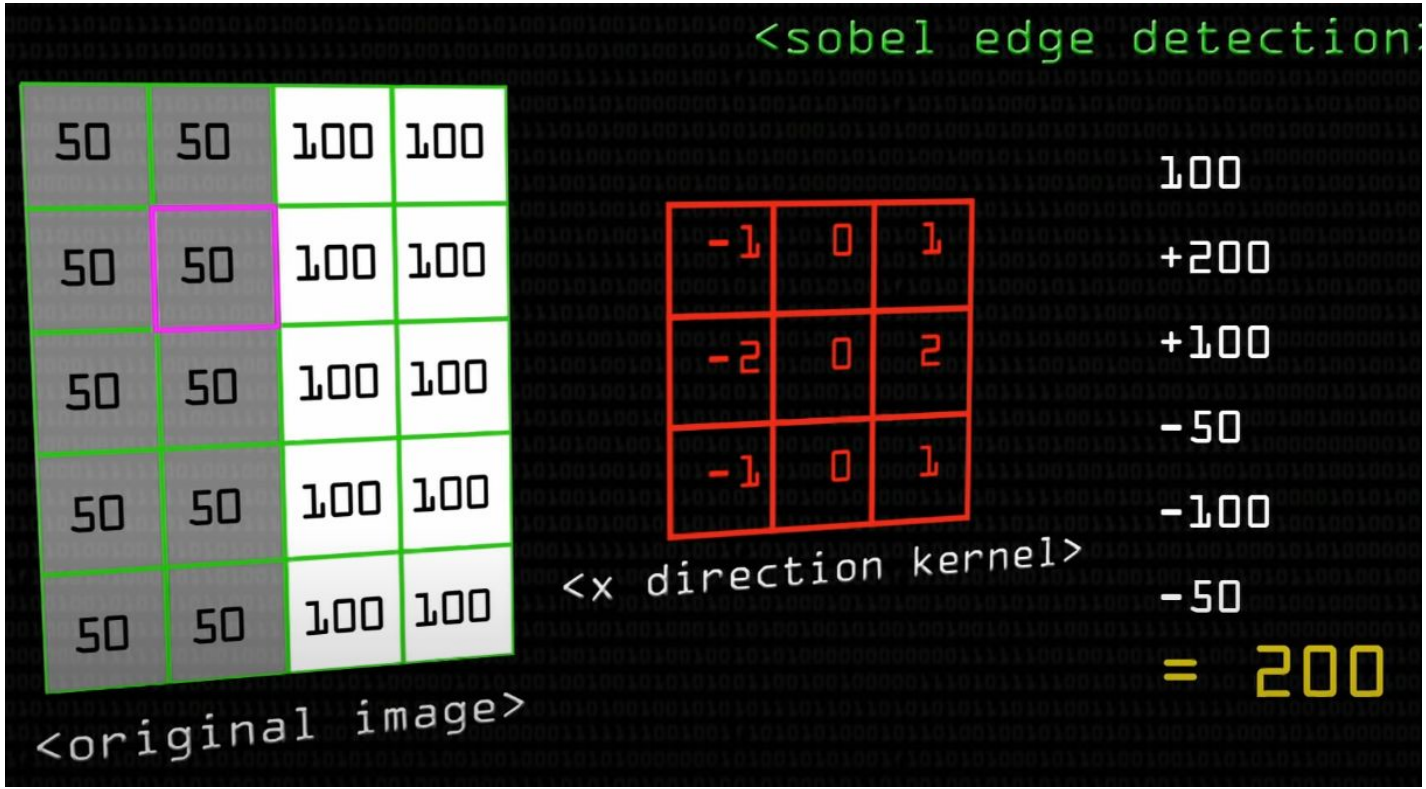


Image would be BW and smoothed

Picture from “Finding the Edges (Sobel Operator) - Computerphile”. Computerphile. Mike Pound. URL: https://www.youtube.com/watch?v=uihBwtPIBxM

Sobel Results

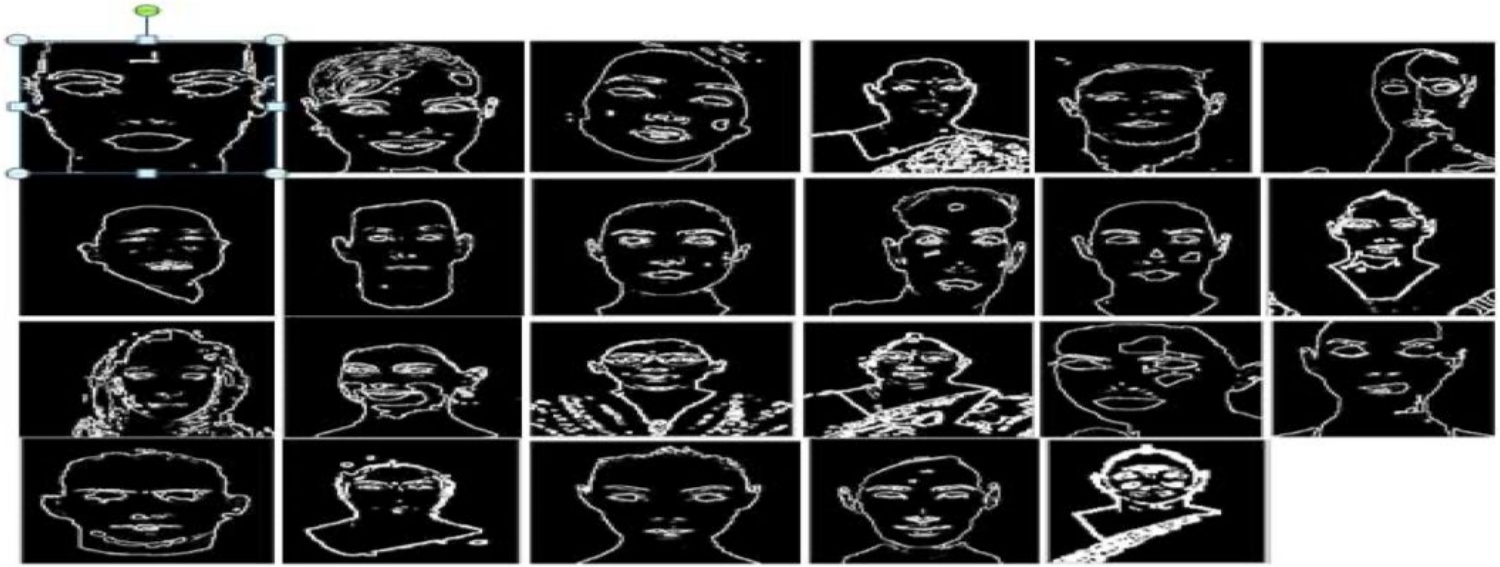
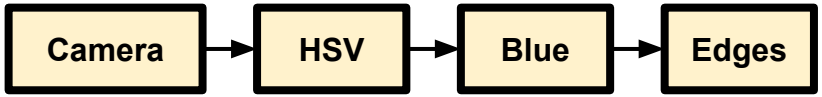


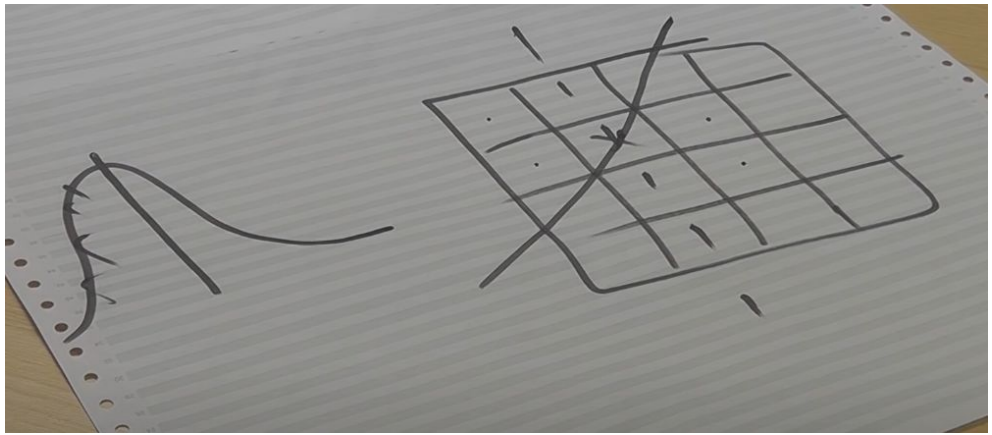
Figure 3: Sobel edge detection

Vijayarani & Vinupriya. International Journal of Innovative Research in Computer and Communication Engineering, Vol. 1 (2013). URL: Link.



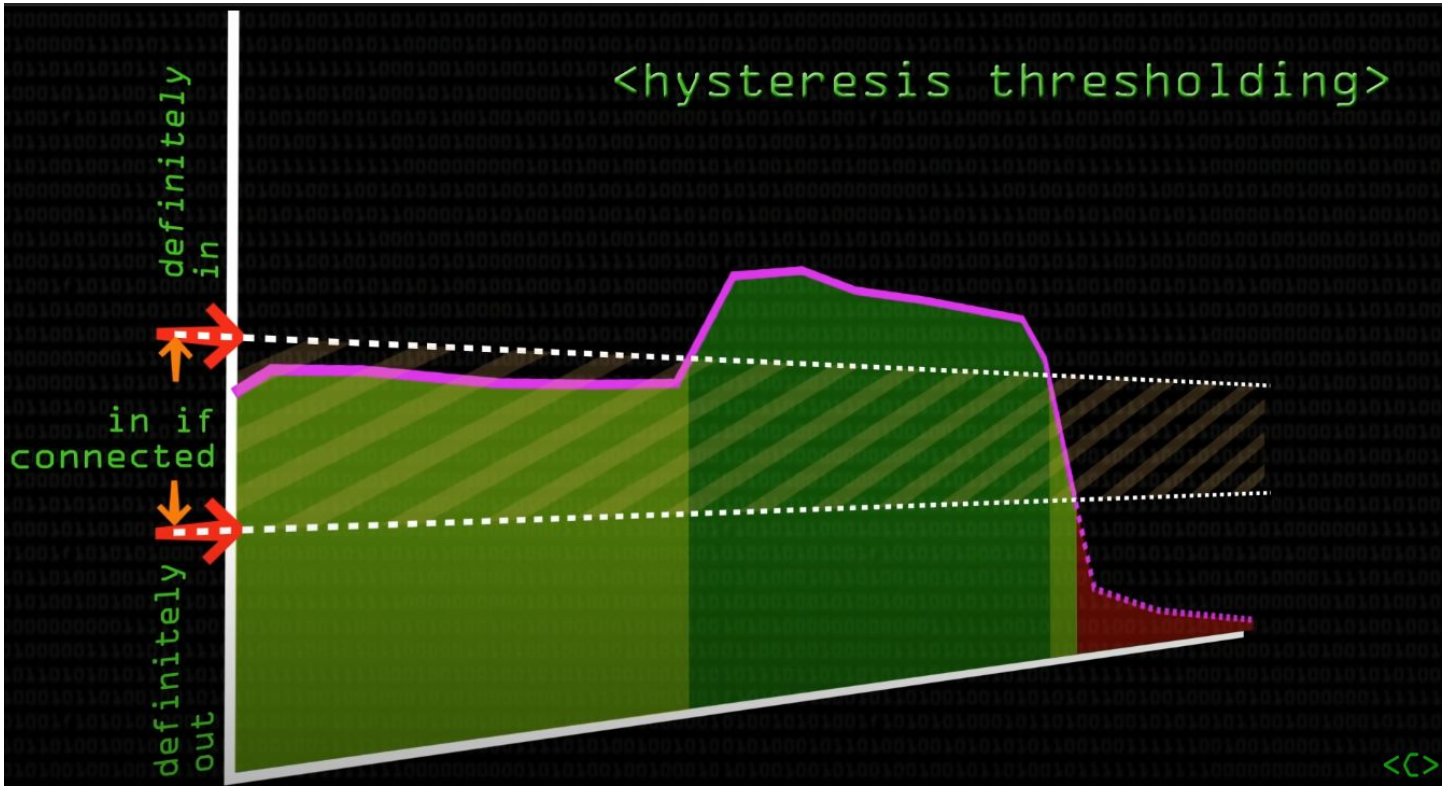
Canny Edge Detection

- Improves upon Sobel
- Sobel feeds into Canny edge detector
- Removes weak edges
- Two steps:
  - Makes edges 1 pixel wide
  - **Hysteresis** - two thresholds



Info & screenshot from: “Canny Edge Detector - Computerphile”. Computerphile. Mike Pound. URL: https://www.youtube.com/watch?v=sRFM5IEqR2w

Canny Edge Detection: Hysteresis Thresholding



Screenshot from: “Canny Edge Detector - Computerphile”. Computerphile. Mike Pound. URL: https://www.youtube.com/watch?v=sRFM5IEqR2w

Sobel Vs. Canny Edge Detection

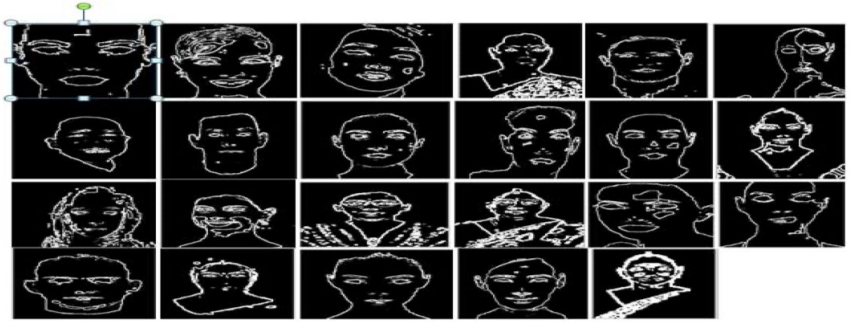


Figure 3: Sobel edge detection

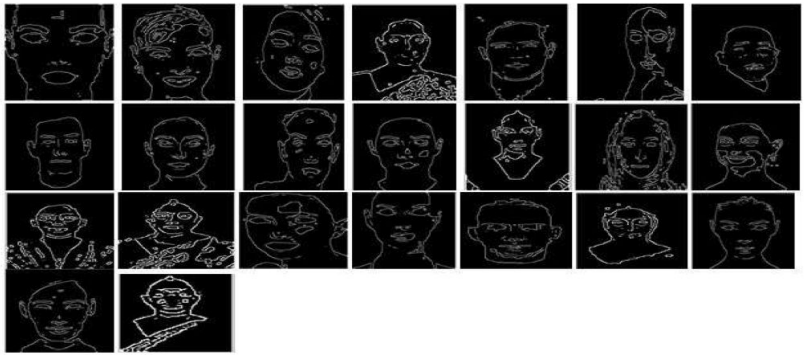
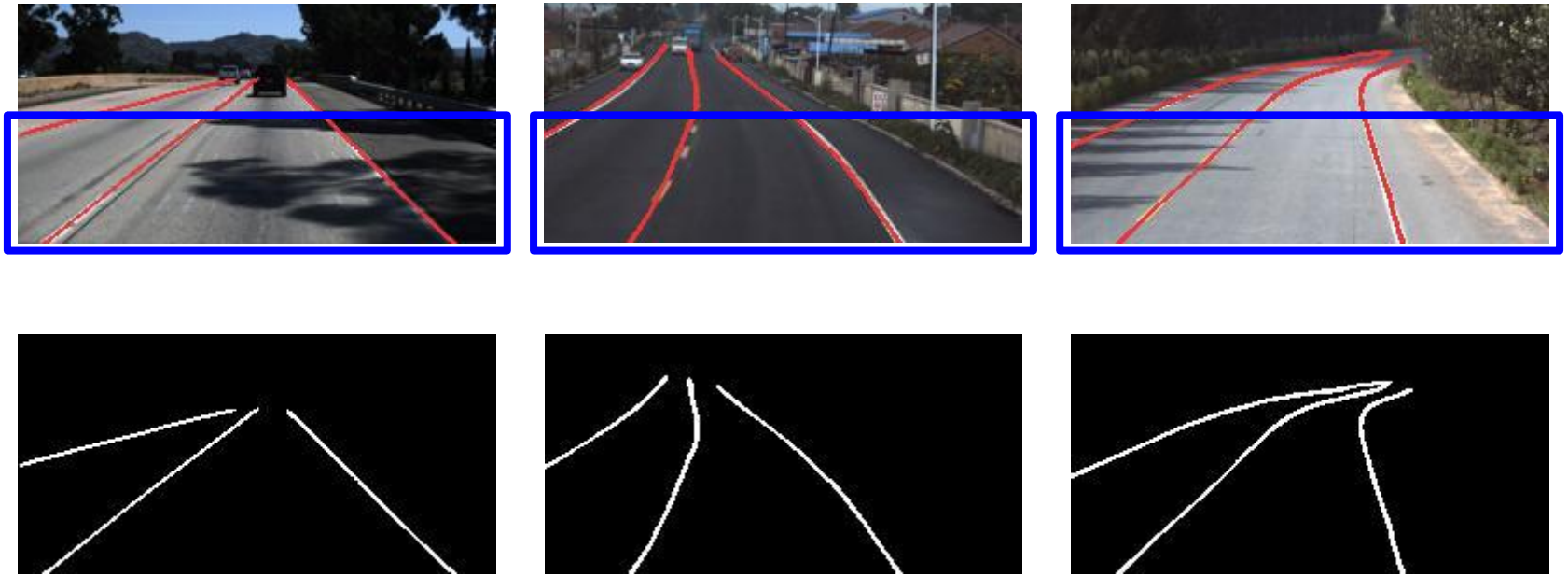


Figure 4: Canny edge detection

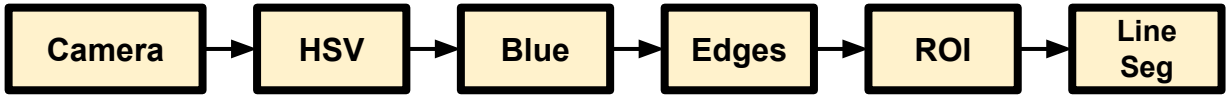
Vijayarani & Vinupriya. International Journal of Innovative Research in Computer and Communication Engineering, Vol. 1 (2013). URL: Link.

ROI: Region of Interest

Do We Need The Whole Scene?



Images from: “Robust Lane Detection From Continuous Driving Scenes Using Deep Neural Networks”. URL: https://github.com/qinnzou/Robust-Lane-Detection



Line Segment Detection

- How do we infer and present a shape?
- We have edges
- But there are holes