

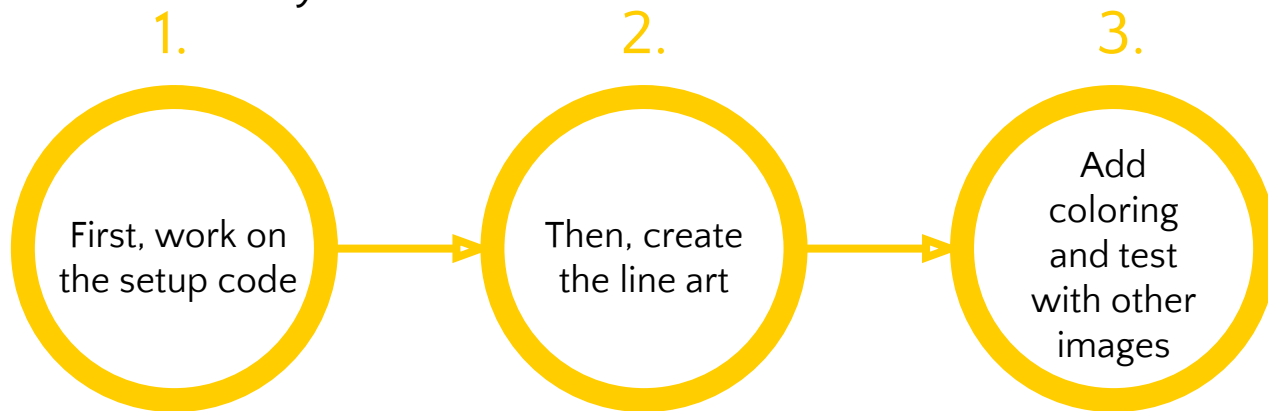
EQ: How can we apply edge detection in real life?



Lesson 09 - Edge detection Project pt. 2

Do Now

With your partner, discuss your plan for working on the project today. Talk about which functions you want to write/revise and how you think they might work. No code yet!



Roll for **confidence!**





Line Art Generator - Reference Sheet

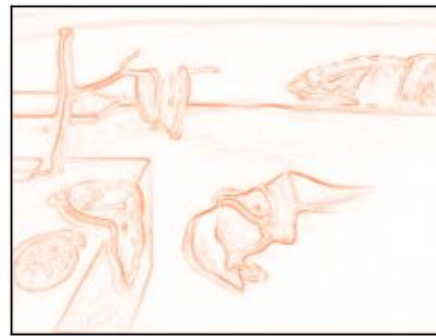
Good places to look for help:

- Smoothing out an image with a Gaussian kernel - [L5, S16](#)
- Creating a Gaussian kernel - [L6, S2](#)
- Merging x and y edges - [L7, S10](#)
- Prewitt kernel - [L7, S2](#)
- Sobel kernel - [L7, S4](#)

Original



Line Art





Line Art Generator - Reference Sheet

Prewitt x kernel

-1	0	1
-1	0	1
-1	0	1

Prewitt y kernel

1	1	1
0	0	0
-1	-1	-1

Sobel x kernel

-1	0	1
-2	0	2
-1	0	1

Sobel y kernel

1	2	1
0	0	0
-1	-2	-1

Scharr x kernel

-3	0	3
-10	0	10
-3	0	3

Scharr y kernel

3	10	3
0	0	0
-3	-10	-3



Let's get **movin'**!



*Any **questions** ?*

Project description and starter code on
Google Classroom.

Project due Monday.



Line Art Generator - Extra credit

Using [this video](#) and other resource you find online, research the Laplacian operator. Answer the following questions in a few sentences about edge detection using the Laplacian and add the Laplacian kernel to your line art generator.

1. What does the second derivative represent in terms of edge detection? Is any information lost using the second derivative?
2. Describe three ways in which the Laplacian differs from the other edge detection kernels we've seen. (Exclude second derivative.)