import cv2

import numpy as np

image = cv2.imread('images/input.jpg',0)

height, width = image.shape

sobel\_x = cv2.Sobel(image, cv2.CV\_64F, 0, 1, ksize=3)

sobel\_y = cv2.Sobel(image, cv2.CV\_64F, 1, 0, ksize=3)

cv2.imshow('Original', image)

cv2.waitKey(0)

cv2.imshow('Sobel X', sobel\_x)

cv2.waitKey(0)

cv2.imshow('Sobel Y', sobel\_y)

cv2.waitKey(0)

sobel\_OR = cv2.bitwise\_or(sobel\_x, sobel\_y)

cv2.imshow('sobel\_OR', sobel\_OR)

cv2.waitKey(0)

laplacian = cv2.Laplacian(image, cv2.CV\_64F)

cv2.imshow('Laplacian', laplacian)

cv2.waitKey(0)

canny = cv2.Canny(image, 50, 120)

cv2.imshow('Canny', canny)

cv2.waitKey(0)

cv2.destroyAllWindows()

import cv2

import numpy as np

# Our sketch generating function

def sketch(image):

# Convert image to grayscale

img\_gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

# Clean up image using Guassian Blur

img\_gray\_blur = cv2.GaussianBlur(img\_gray, (5,5), 0)

# Extract edges

canny\_edges = cv2.Canny(img\_gray\_blur, 10, 70)

# Do an invert binarize the image

ret, mask = cv2.threshold(canny\_edges, 70, 255, cv2.THRESH\_BINARY\_INV)

return mask

cap = cv2.VideoCapture(0)

while True:

ret, frame = cap.read()

cv2.imshow('Our Live Sketcher', sketch(frame))

if cv2.waitKey(1) == 13: #13 is the Enter Key

break

# Release camera and close windows

cap.release()

cv2.destroyAllWindows()