Edge Detection

1. Compute and display edge magnitude image and edge angle image of Lena.png
    - Read the image as gray-scale image
    - Use Sobel edge operator
2. Compute canny edge image of Background.mp4
    - Read each frame as gray-scale image
    - Change th_l and th_h values and see what happens

Line Detection

1. Detect lanes in “Downtown_road.mp4”
   - Read each frame as a gray-scale image and perform line detection
   - Draw lines corresponding to lanes in red in the result image (result image should be displayed in color image)
     - Hint: Proper setting of ROI region is necessary
   - You may get several lines. Try to merge them

Face Detection

1. Perform face detection on “faces.mp4”. Try to detect the face that is closest to the camera
2. Make an integral image for ‘lena.png’ and compute sum of pixel values in the following area
   - Read the image as a gray-scale image
   - LT: (100,100), BR: (200,200)
   - LT: (150,150), BR: (170,170)
   - LT: (200,200), BR: (210,210)

Pedestrian Detection

1. Perform pedestrian detection on “pedestrian_detection.mp4” and print out the number of pedestrians on the image.
    - Draw a bounding rectangle on each pedestrian.

2. Perform pedestrian detection on “pedestrian_detection.mp4” and print out the number of moving pedestrians on the image.
    - Draw a bounding rectangle on each moving pedestrian. Set the color of rectangle as green.
    - Draw a filled-bounding rectangle on each non-moving pedestrian. Set the color of rectangle as red.

Object Detection using deep learning

1. Perform object detection using “Desk.mp4”. Try to change confidence_threshold and see what happens.

2. Perform object detection using on “Go_1.mp4”. Based on detection result, estimate whether the distance between your car and the front car is i)same, ii)larger, and iii)smaller

Tracking

1. Perform face detection/tracking on “faces.mp4”. Try to detect the face that is closest to the camera. Once detection is done, perform tracking. Compare the result of detection/tracking with that of detection on every frame.
2. Compute optical flows of the region in “background.mp4” that is specified by mouse dragging.

Image transformation

1. Display the contents of “contest.mp4” on the board of “Timesquare.mp4”. Set the position of the board by mouse clicking.

2. Do the same thing with prob. 1 by explicitly setting the position of the board (not by mouse clicking)