Q 31-A Edge Detection

a) The gradient is preffered for edge detection because edge detection requires you to analyze the rate of change of neighbouring pixels to the seed pixel. Since the gradient is the derivative, it provides this. In addition, the Laplacian exhibits the "double edge" effect which has a negative impact on edge detection.

b) the goal of hysteresis thresholding in Canny Edge Detection is to provide a more robust method to threshold along the edge compared to the rest of the image. It works by looping through an image until it finds a pixel whose value is greater than T_H. It then looks at neighbours to check them as well. Pixels with values less than T_L are set to a different value.

c) Main Steps of Canny Edge Detection

1. Smoothing
2. Sharpening / differentiation / Gaussian
3. Non maximum suppression
4. Hysteresis thresholding

d) Main steps in general edge detection:

1. smoothing
2. Sharpening
3. connected components algorithm

Q31-B segmentation

a) main steps of k-means clustering for colour quantization:

- randomly select k seed points, identified by their colour channels
  ↳ these are the initial centres.
- Create Vernoi regions by looping through all pixels in image and associating them with the closest centre. Given that we are dealing in color quantization, the distance metric will use the colour channels. Position may also be required if there are multiple regions with the same colour
- once all pixels are distributed amongst the centres, look at each Vernoi region. Recalculate the centre value. once the centre value minimizes the distance between the centre value and all pixels in region, we are finished. Note: all vernoi regions must meet this condition in order to be finished.

b) two ways to choose the value of k in k-means:

- look at histogram. # of peaks = k.
- use a trackbar and see what looks best (!!)

c)

- select seed
- look at neighbours
- each neighbour that is within thres, add to stack

↳ complete description on next page

c) Main steps for region growing segmentation:

- select a seed pixel a
- look at neighbours of seed pixel (either 4 or 8 connectivity)
- if neighbouring pixel is close in value to seed (given some threshold), add it to the stack. Add all neighbours that satisfy threshold to stack. Mark pixels once they have been added.
- one at a time, pop each neighbour off stack. this becomes the new "seed". look at neighbours of new seed and push them to stack if they satisfy thresholding criteria (and mark them as added)
- continue popping pixels to stack and pushing neighbours to stack until the stack is empty. Then you're finished.