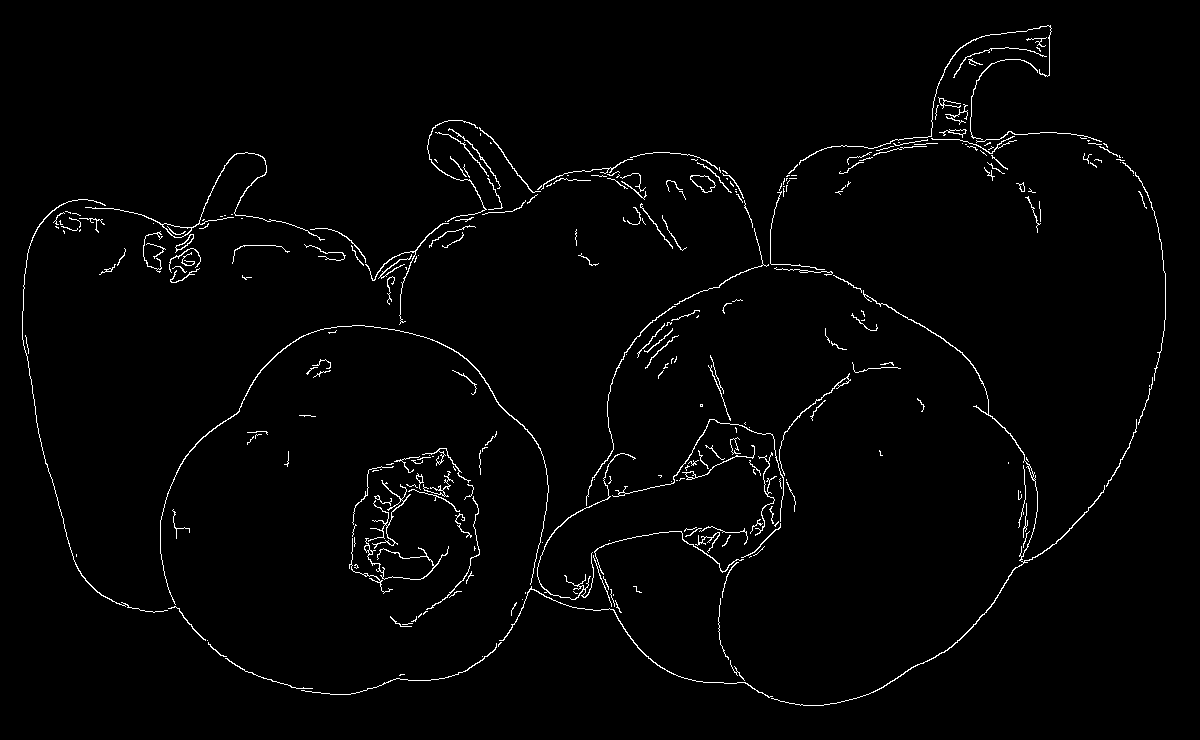
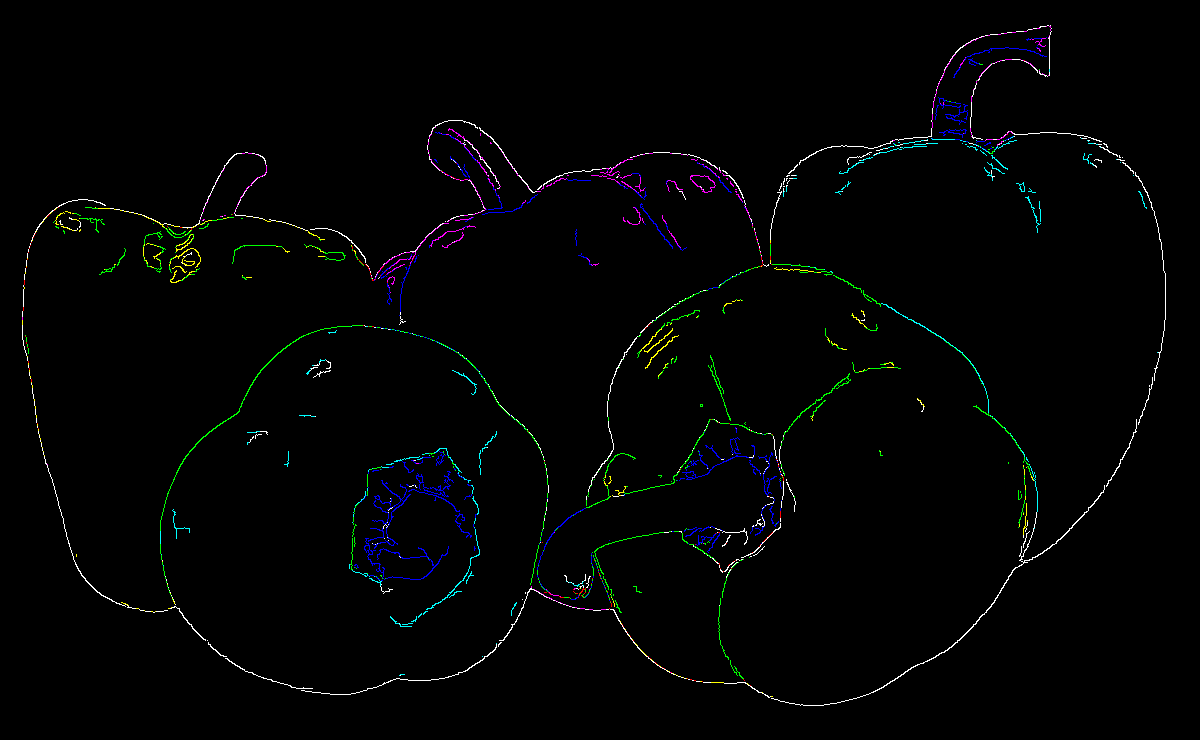
**Report on the program A and B from Assignment 3  
Computer Vision (CS-559)**

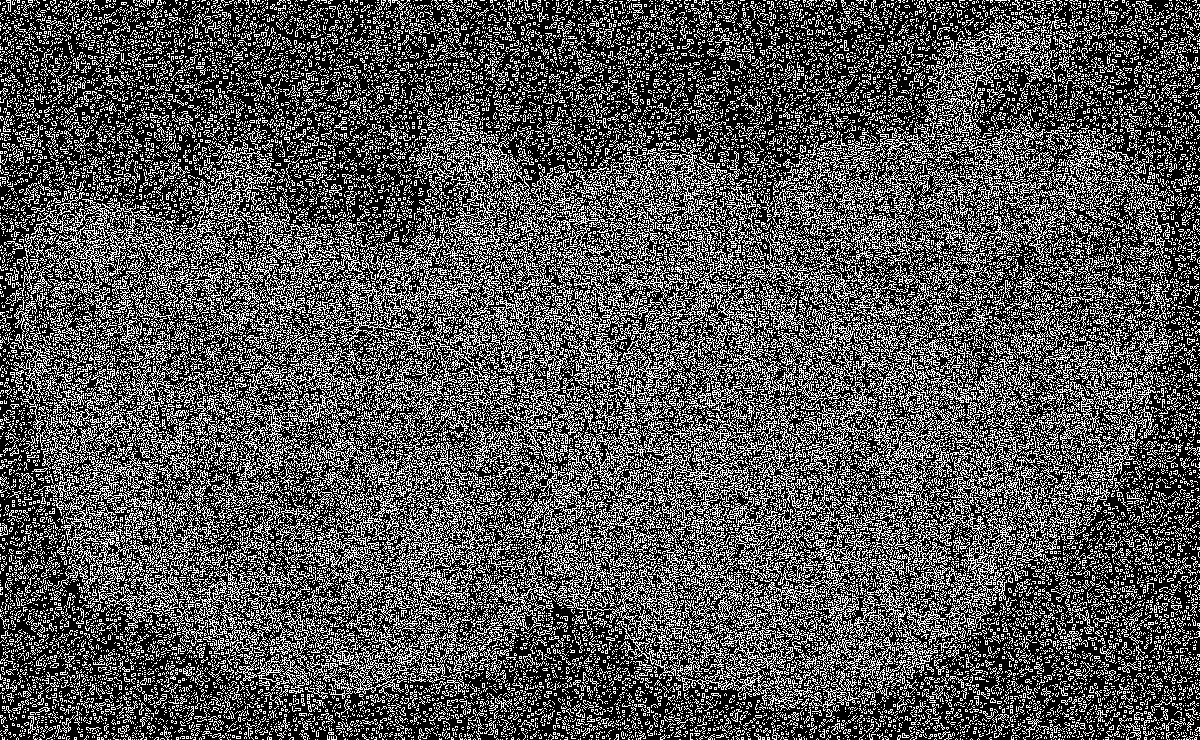
**Name: Dhaval Harish Sharma  
Red ID: 824654344**

**Introduction:**This program takes an image as input and finds its edges using canny edge detection algorithm. It converts the edges of different colours into corresponding colours that is green to blue, etc. After that, it adds salt and pepper noise in to the input image, applies the same edge detection algorithm again and tries to remove the noise using median filter.

**Working of the program:**The program starts by importing the necessary libraries for its working. Skimage is used for converting the image to a numpy array. Matplotlib is used to display the results of the operations in the output. Colorsys is used for converting the image from rgb to hsv in order to sort the pixels for applying median filter. CV2 is used for applying the canny edge detection algorithm. Initially, the image is taken as input and converted to a numpy array containing the corresponding pixel values in the image. We apply the canny edge detection algorithm from cv2 library for getting the edges in the image. But, the rgb image is converted to a greyscale image after applying the canny edge detection algorithm. So, to get the original colours in the image, we traverse the whole image, replacing the black and white pixels with the original pixels in the input image. At the same time, we are saving the rgb values for all the edges which can be used to find the mean and standard deviation later. After getting all the rgb values, we find the mean and standard deviation of the corresponding channels which will be helpful for getting an idea about the average number of pixel values in each channel. We again traverse through the image, finding the colours and replacing them with the corresponding colours. In this program, red is converted to green, green is converted to blue, blue is converted to red, cyan is converted to magenta, magenta is converted to yellow and yellow is converted to cyan. All the colours except these colours are turned to white. The result of the above operations gives us the edged image with different colours.

We then add salt and pepper noise to the input image. First of all, we count the number of salt and pepper pixels in the image which is 20% of the total pixels. Then, we randomly replace the pixels with either black or white. After adding salt and pepper noise, we use the canny edge detection from cv2 to find the edges of the image with salt and pepper noise. Then, we apply the median filter on the image affected with salt and pepper noise. We do this by maintaining a window of size 5 x 5 and getting the middle element from the sorted pixels. Lastly, we display the resulting images in the output using matplotlib.

The results of each operation are as follows:  
   
 Original Image Canny edge detection  
   
 Coloured Edges Salt and pepper

   
 Canny edge detection on salt and pepper Filtered image

**Findings:**Edge detection is a very good technique to capture important events and changes in properties of the world. Canny edge detection is a powerful edge detection method to detect the edges in an image accurately. The idea of adding colours to the different edges is a bit challenging as it involves many minute details to work on. But, once applied it can be very helpful for finding different objects from an image. The salt and pepper noise can be added unintentionally in an image due to faulty transmission, equipments, etc. The canny edge detection doesn’t work as it is supposed to on the images with salt and pepper noise. The salt and pepper noise can be easily removed without losing much of the details from the original image by applying median filter.

**Conclusion:**In conclusion, finding the edges in an image is an important operation as it can help to find several details in the image. We can find different objects in an image by finding different colours for different edges. The salt and pepper noise is inevitable to avoid in several applications such as satellite transmission, space communication, etc but it can be easily removed using ordered filters.