

# REPORT

[**Assumption** : Address of the folder given to CollageCreate(Address folder) will contain at least 6 images as mentioned in the assignment.]

## How images are sorted

- **Canny edge detector** is used to find the **edge information** from grayscale images of all the 6 images.
- For noisy images, canny edge detector detect wrong edges. For that, **gaussian filter of  $\sigma=3$  is applied** so that only meaningful edges information will be collected. For lower  $\sigma$  values filter, canny edge detector consider more edge details but the information may not be meaningful.
- In canny edge detector of skimage, double thresholding is applied to detect low, weak and non-relevant pixels which is further followed by hysteresis to give **binary pixel image** (contain either low or high intensity pixels only).
- All images are sorted on the basis of total number of high intensity pixel counts in an image. **High intensity pixels resembles more boundaries detected.**

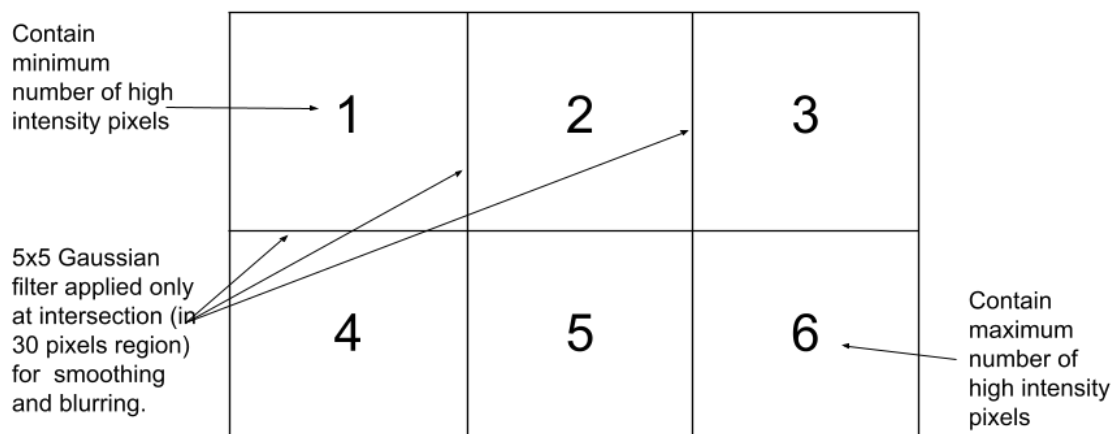


Fig. 1 Grid layout of collage

- In the Fig.1 grid, **image at location 1 will contain minimum number of high intensity pixel counts and image 6 will contain maximum number of high intensity pixel counts.**

### **How images are combined**

- Output resolution of collage is 639X480
- In collage each image is of equal size i.e., 213X240 by resizing.
- After sorting, all respective rgb images are placed on collage.
- **Only intersecting edges are smoothed and blurred by using a 5X5 gaussian filter to provide continuity (within the nearest 30 pixel region).**

### **Output**



**Fig. 2 Final Collage**

Collage of the canny edge detector images sorted on the basis of set pixel count of binary image

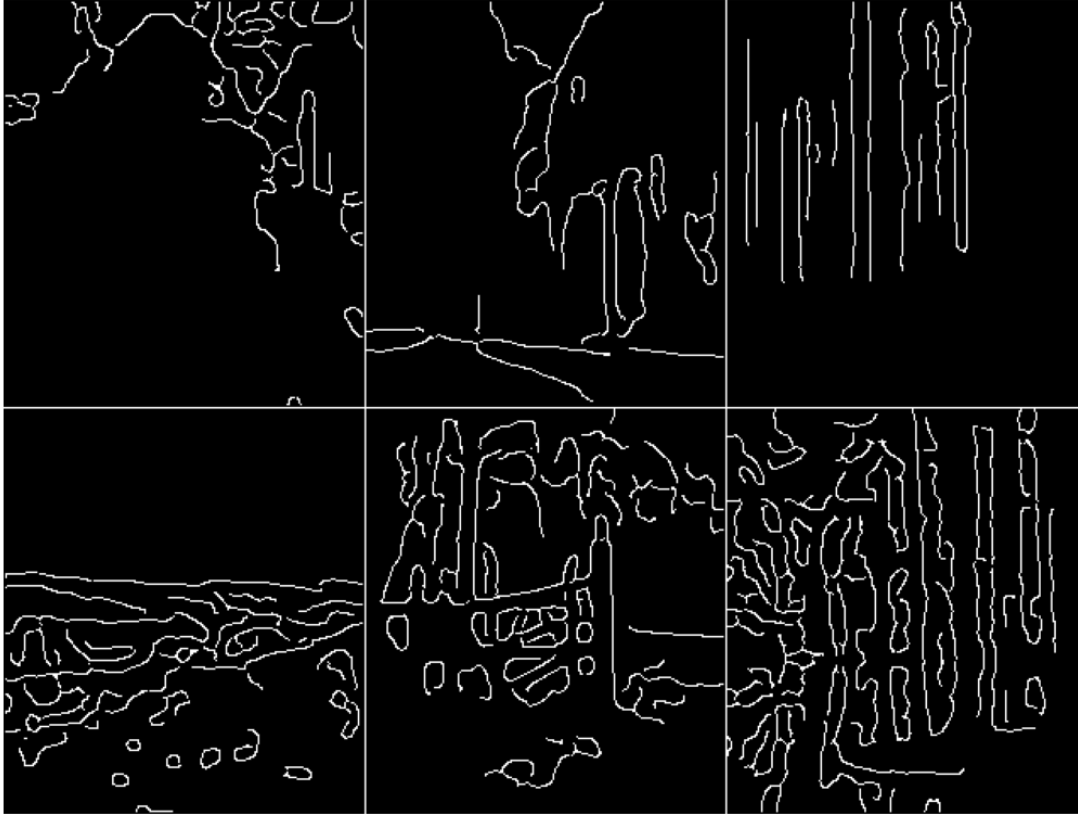


Fig. 3 Final Collage (Canny edge detector view) using high  $\sigma$  values ( $\sigma = 3$ )

### Run Command at terminal:-

```
import CollageCreator as cc  
cc.CollageCreate("frames")
```