भारतीय प्रौद्योगिकी संस्थान तिरुपति

TIRUPATI

Indian Institute of Technology Tirupati

Image Processing Lab

Lab sheet. No: 05

Name: Pedasingu Rajesh **Roll. No:** EE21M019

1. Canny Edge Detection:

Algorithm :

1. *Smooth the image with a Gaussian filter.*
2. *Compute the gradient of the image (both magnitude M(m,n) and direction ϑ (m,n)) using an edge detector like Sobel.*
3. *The edge direction angle is rounded to one of the eight angles representing vertical (90° ± 22.5°, 270° ± 22.5°), horizontal (0° ± 22.5°, 180° ± 22.5°), and the two diagonals (one with 45° ± 22.5° and 225° ± 22.5°, and the other with 135° ± 22.5° and 315° ± 22.5°).*
4. *Suppress non-maxima pixels in the edges obtained above to thin the edge ridges. To do so, check whether each gradient magnitude is greater than its two neighbors along the gradient direction. If so, keep it unchanged, otherwise, set it to 0.*
5. *Apply two different thresholds t1 and t2 (where t1< t2) to obtain two binary images Image1 and Image2. (Note that Image2 with greater t2 has less noise and fewer false edges but greater gaps between edge segments, when compared to Image1 with smaller t1.)*
6. *Link edge segments in Image2 to form continuous edges. To do so, trace each segment in Image2 to its end and then search its neighbors in Image1 to find any edge segment in Image1 to bridge the gap until reaching another edge segment in Image2.*

Write a program that smooths an image and then detects and enhances the edges in the image using "Canny Edge Detector". Your program should output the final result as well as intermediate results at each step.

a) Images with only horizontal and vertical edges. Perform the experiment on "fig1.png" and "fig2.png".

b) Images having lines that are not vertical and horizontal. Perform the experiment on "fig3.png" and "fig4.png".

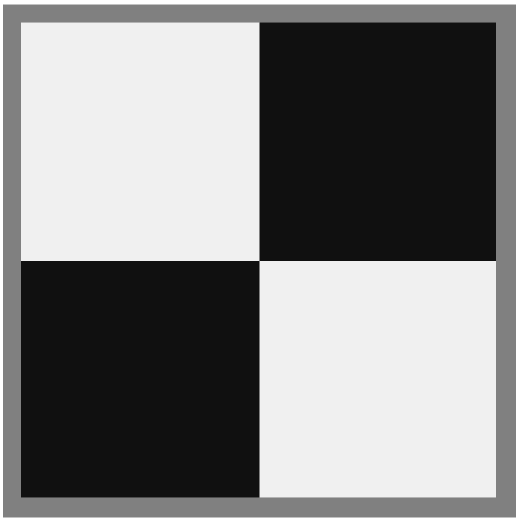
c) Natural Images "fig5.png" and "fig6.png".

d) Modify your Canny edge detection algorithm such that you can detect only diagonal edges. Apply your algorithm on fig4.png and fig5.png. Write your inferences about the results.

e) Compare your results in prob. (a), (b), (c) with results obtained from MATLAB's inbuilt command for Canny edge detection. Comment on your results.

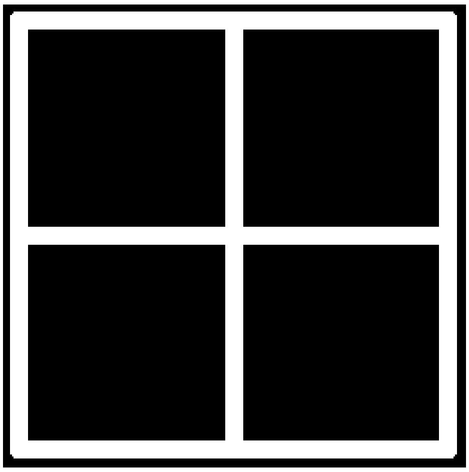
Aim: To find the edges in the given image using cany edge detector algorithm for all given images and also find only the diagonal edges in the given image fig4.png and fig3.png.

Output:

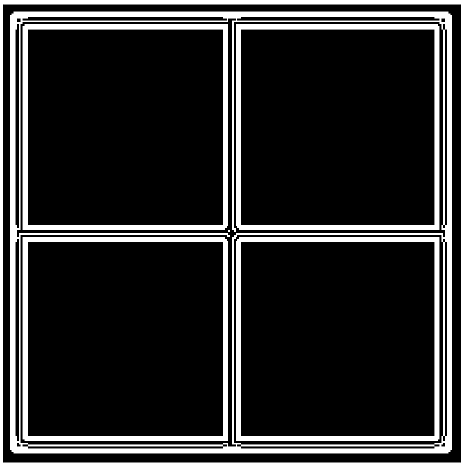


applying gaussian filter

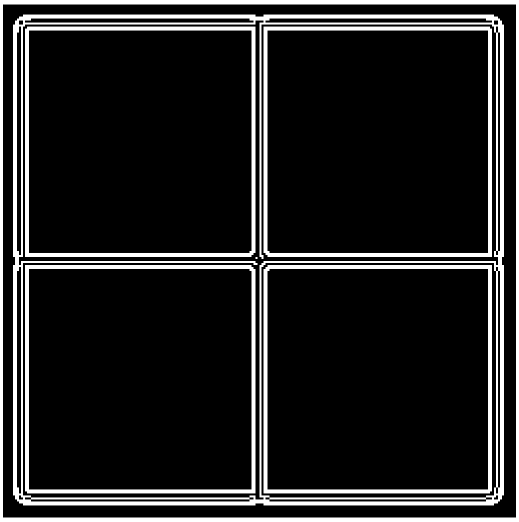
magnitude of gradients

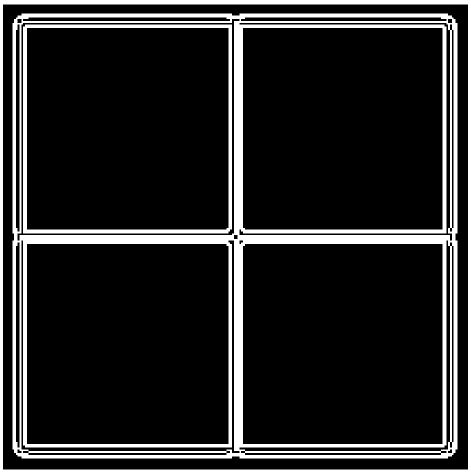


after Suppress non-maxima pixels

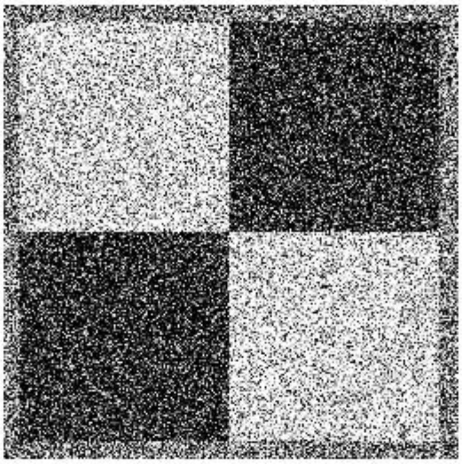


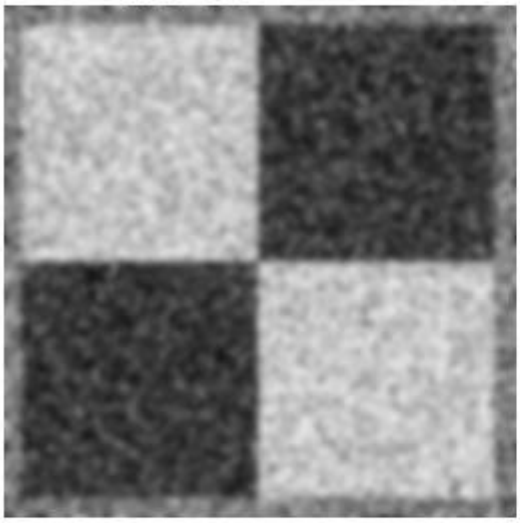
after double thresholding





Canny edge detection algorithm

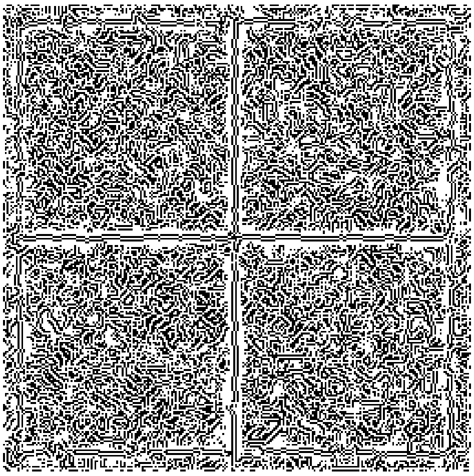




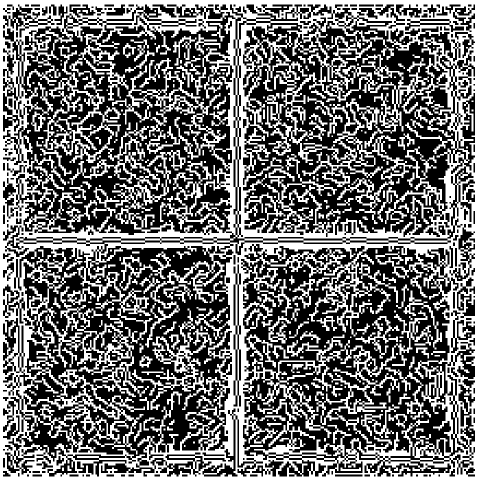
applying gaussian filter

magnitude of gradients

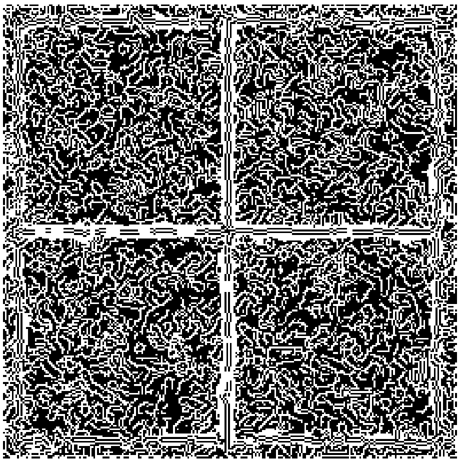
after Suppress non-maxima pixels



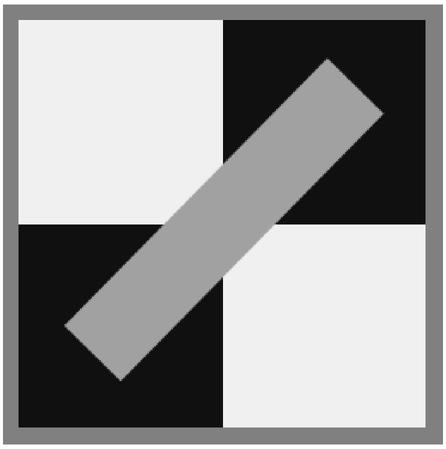
after double thresholding



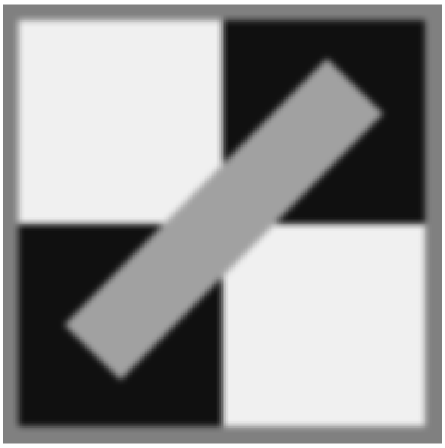
Canny edge detection algorithm

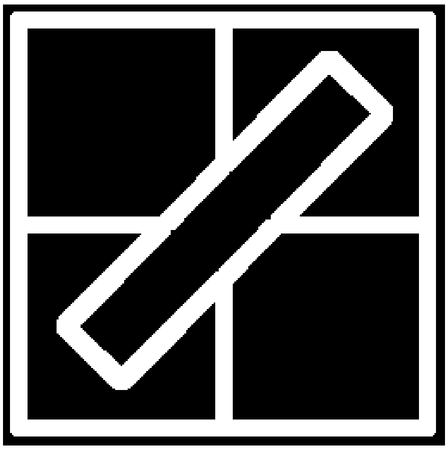


For fig3.png and fig4.png

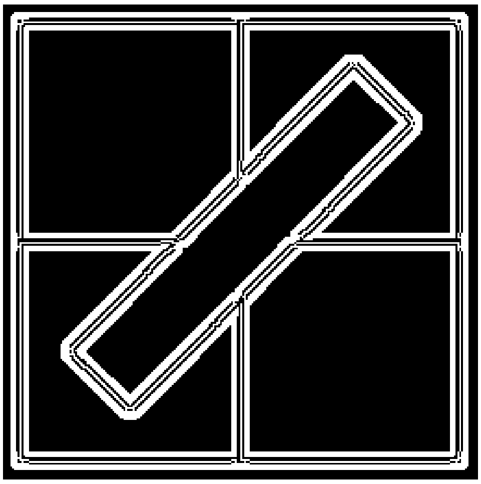


applying gaussian filter

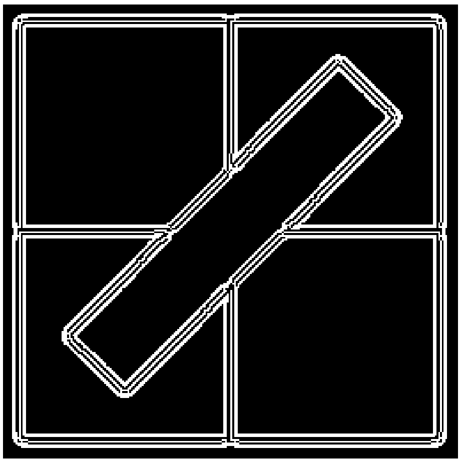




magnitude of gradients

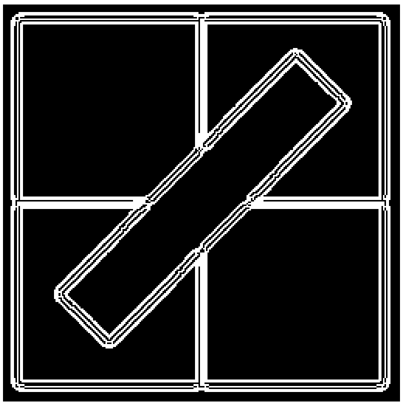


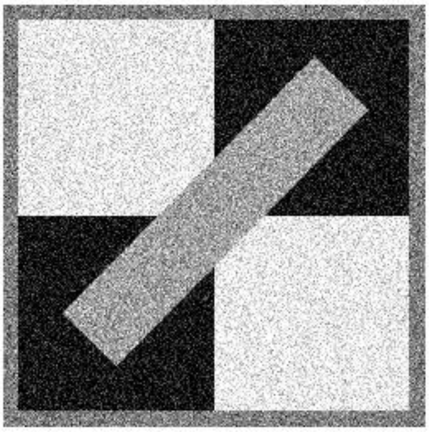
after Suppress non-maxima pixels



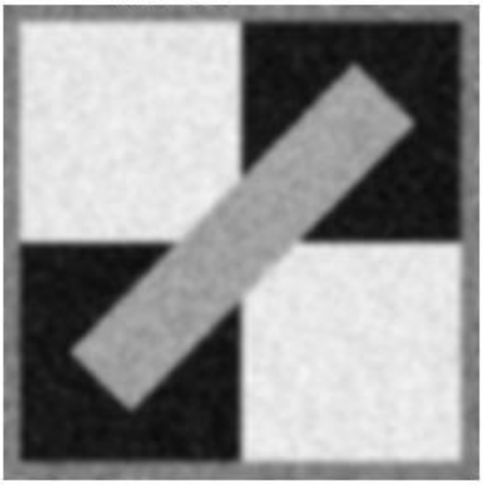
after double thresholding

Canny edge detection algorithm

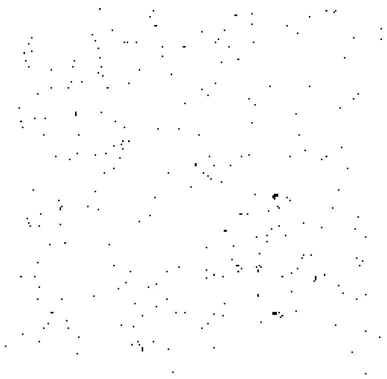




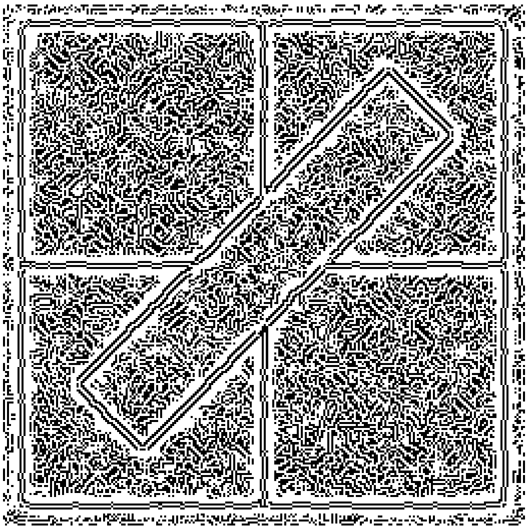
applying gaussian filter



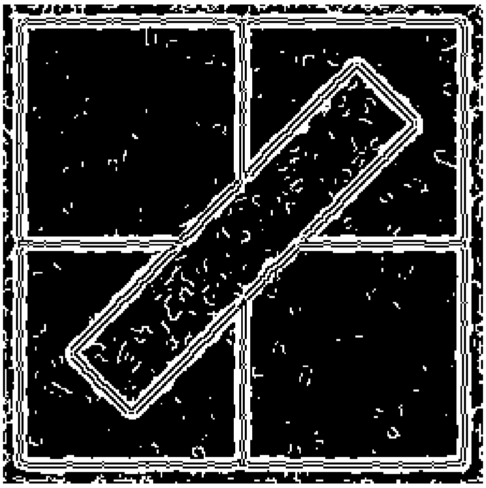
magnitude of gradients

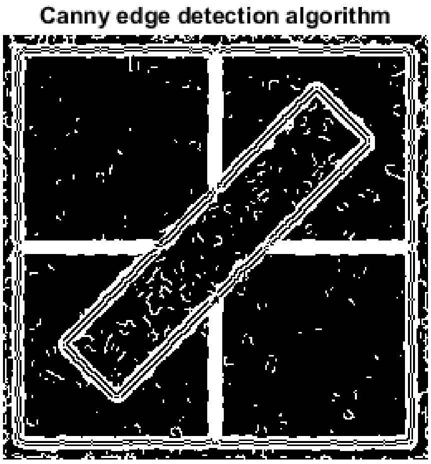


after Suppress non-maxima pixels



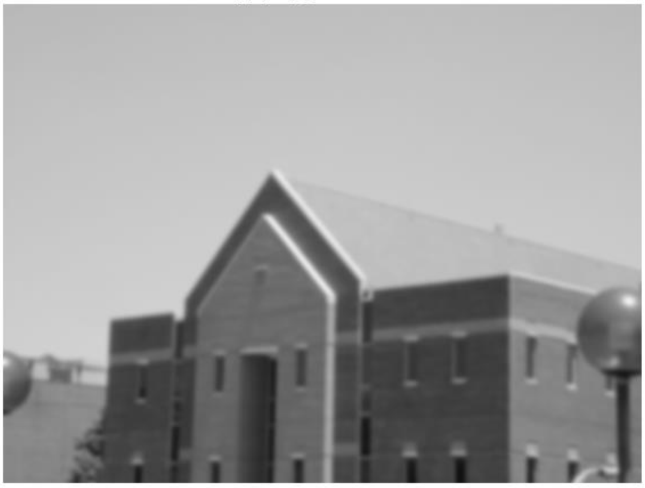
after double thresholding



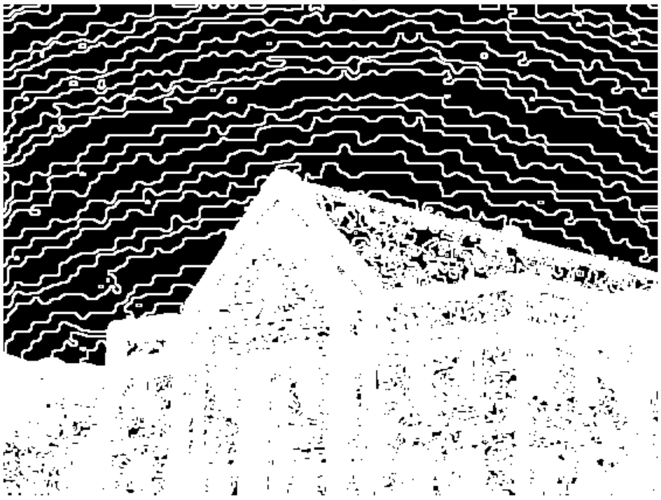


Canny edge detection algorithm

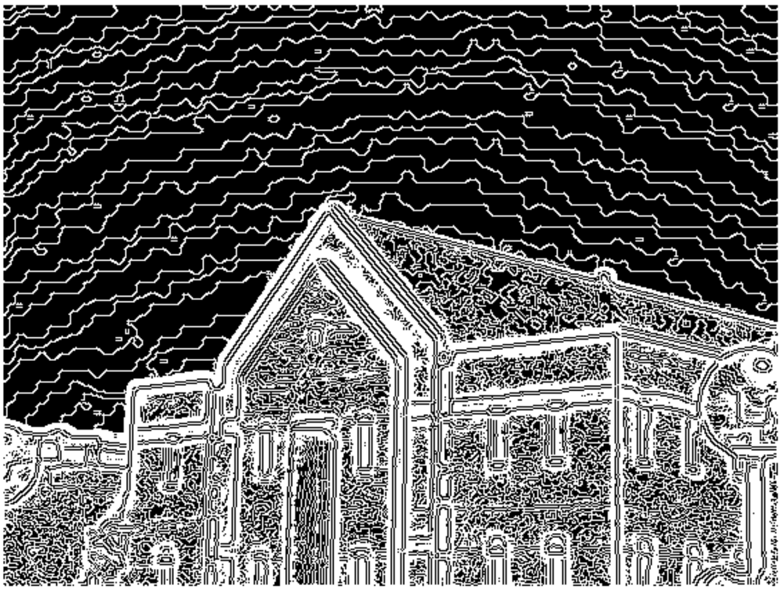
For fig5.png and fig6.png



applying gaussian filter

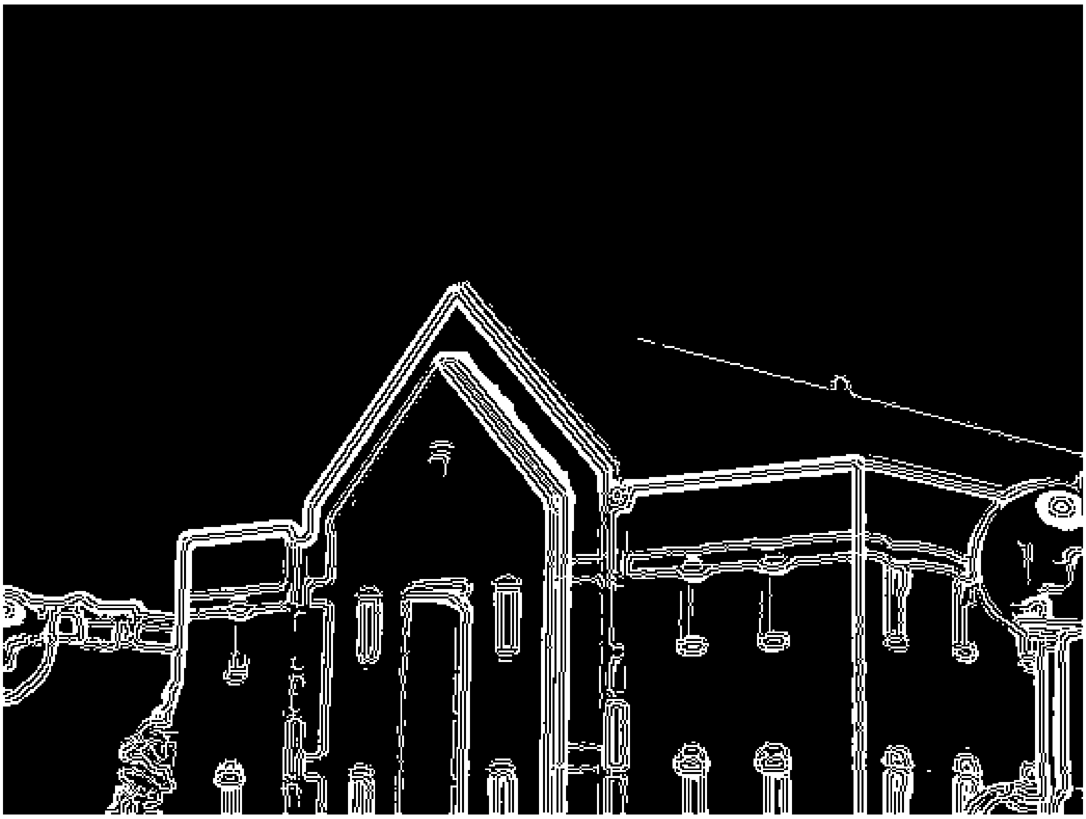


magnitude of gradients

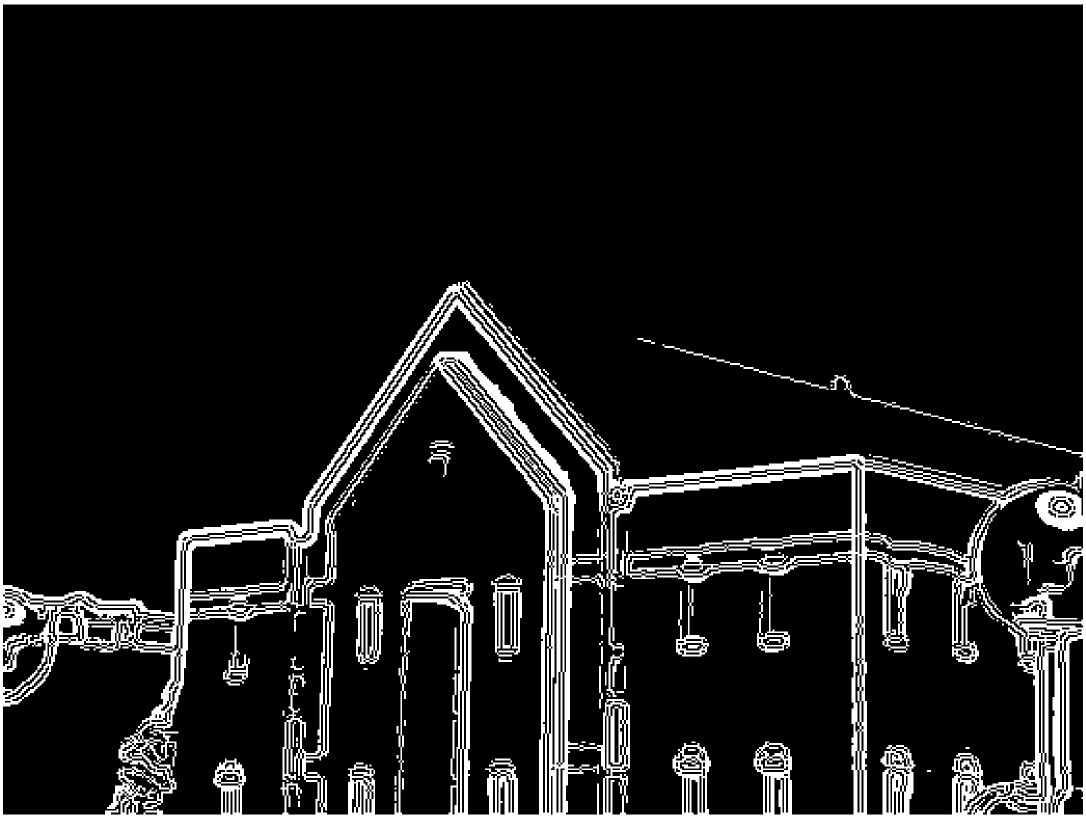


after Suppress non-maxima pixels

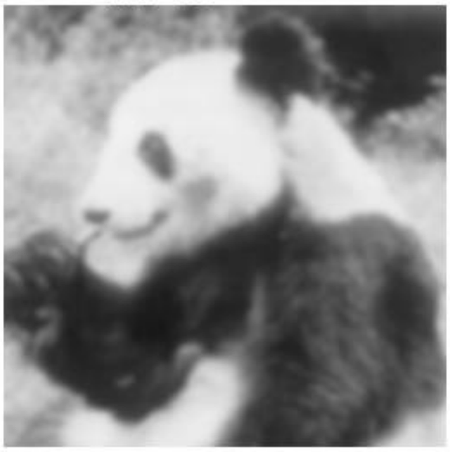
after double thresholding



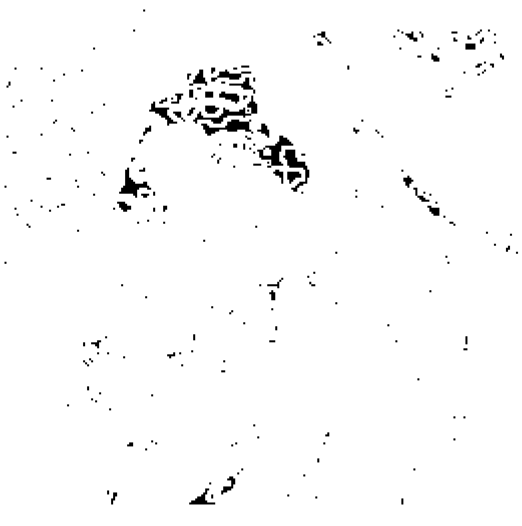
Canny edge detection algorithm



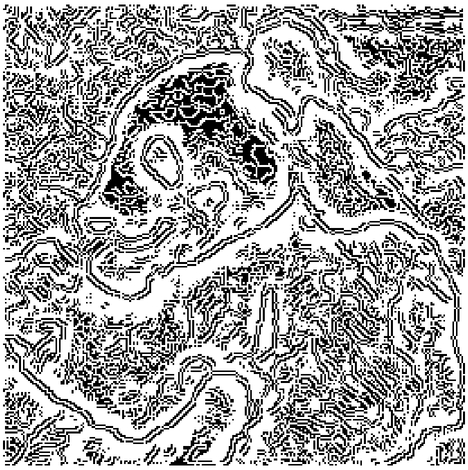
applying gaussian filter



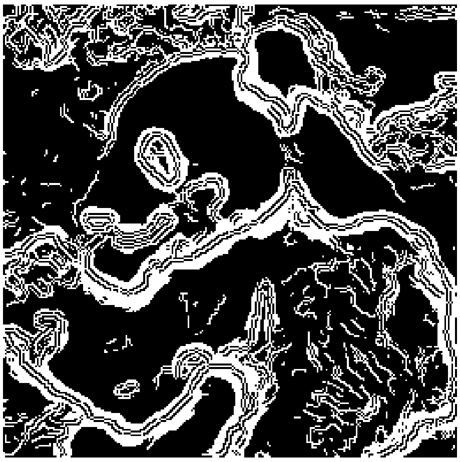
magnitude of gradients



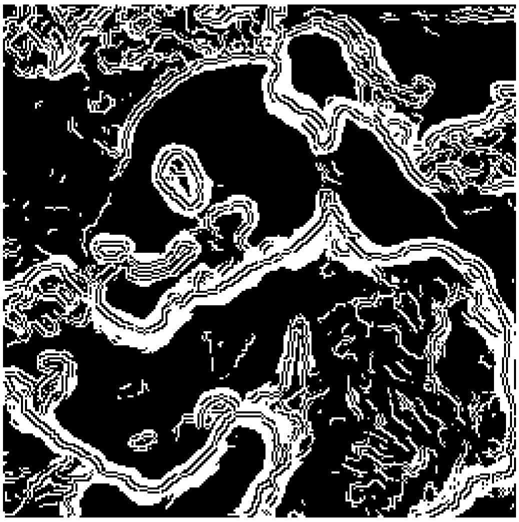
after Suppress non-maxima pixels



after double thresholding

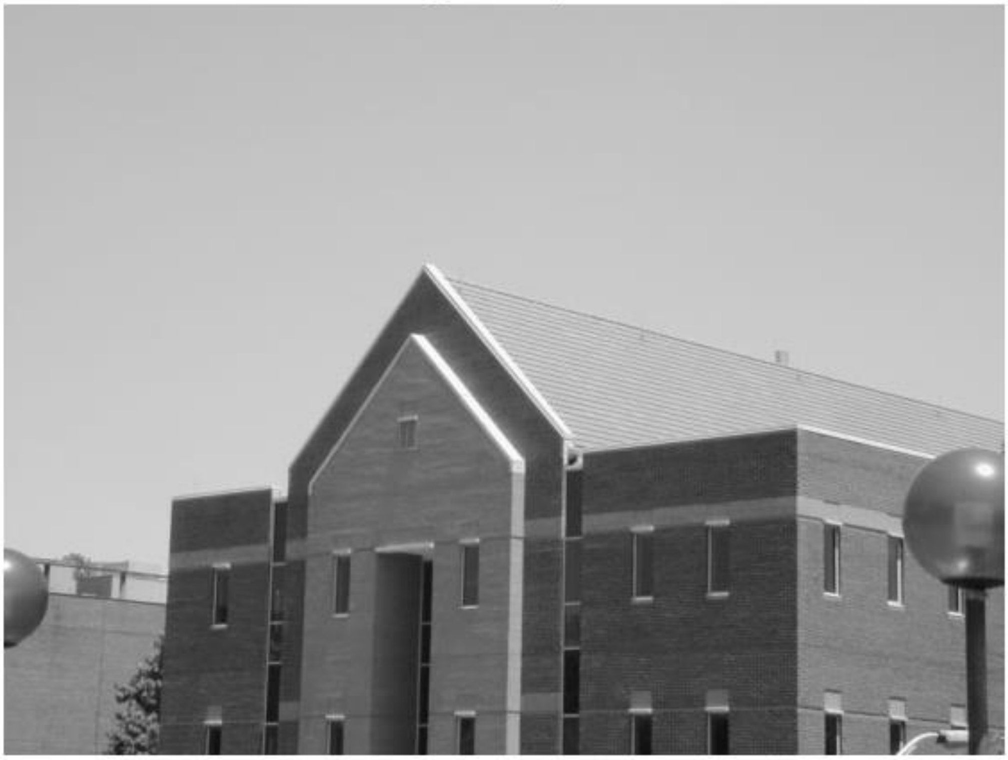


Canny edge detection algorithm

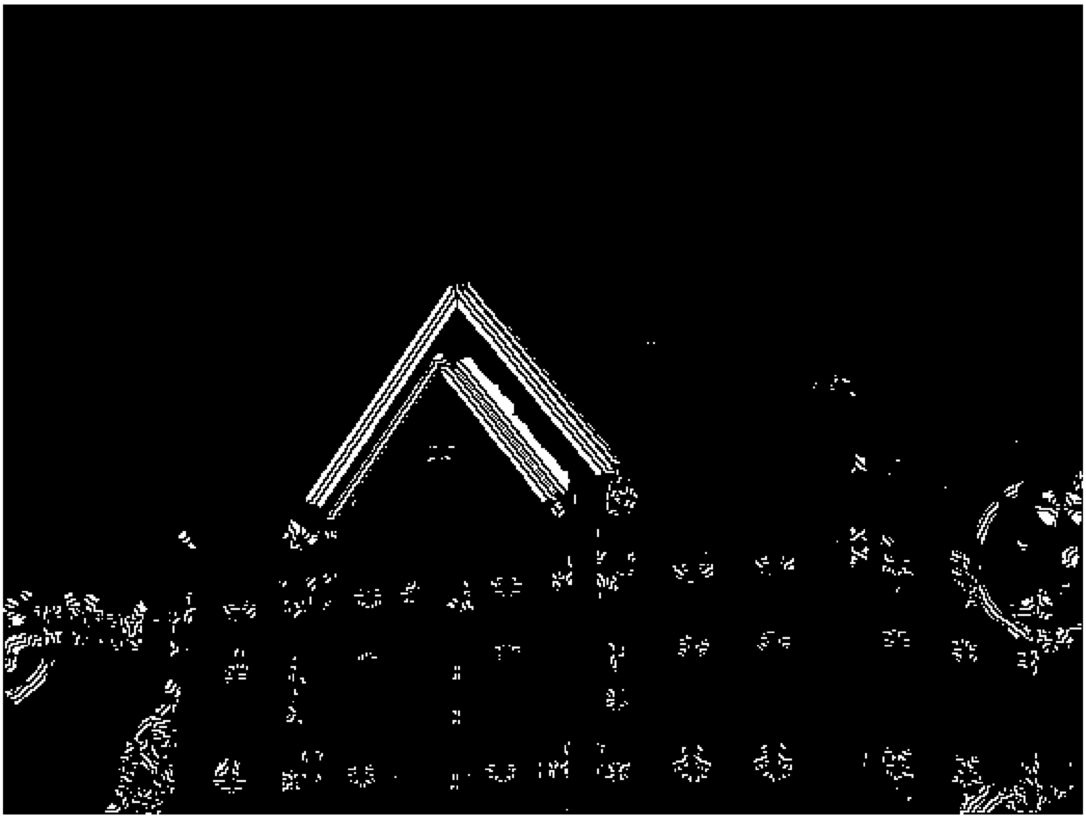


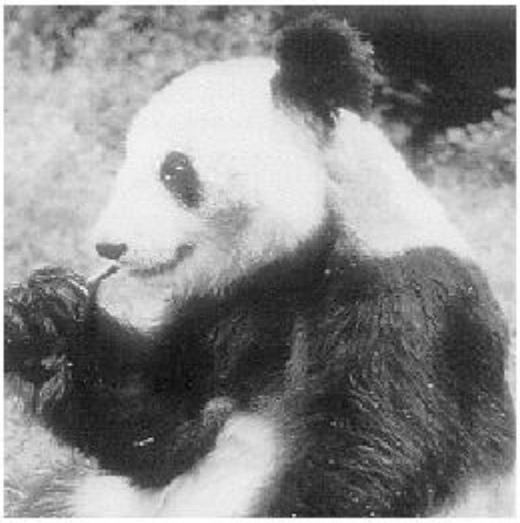
For diagonal edges using cany edge detection algorithm

given image

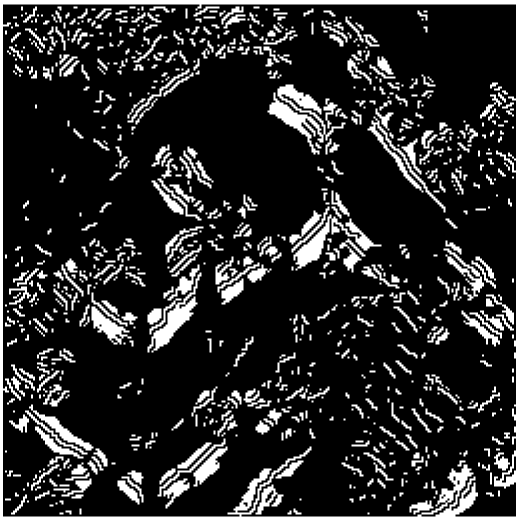


diagonal edges using Canny algorithm



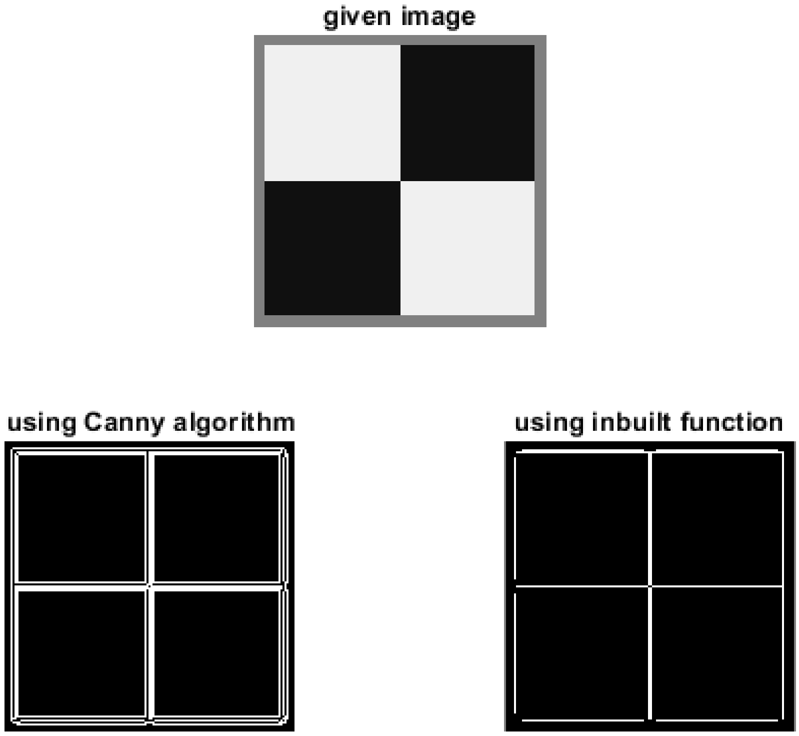


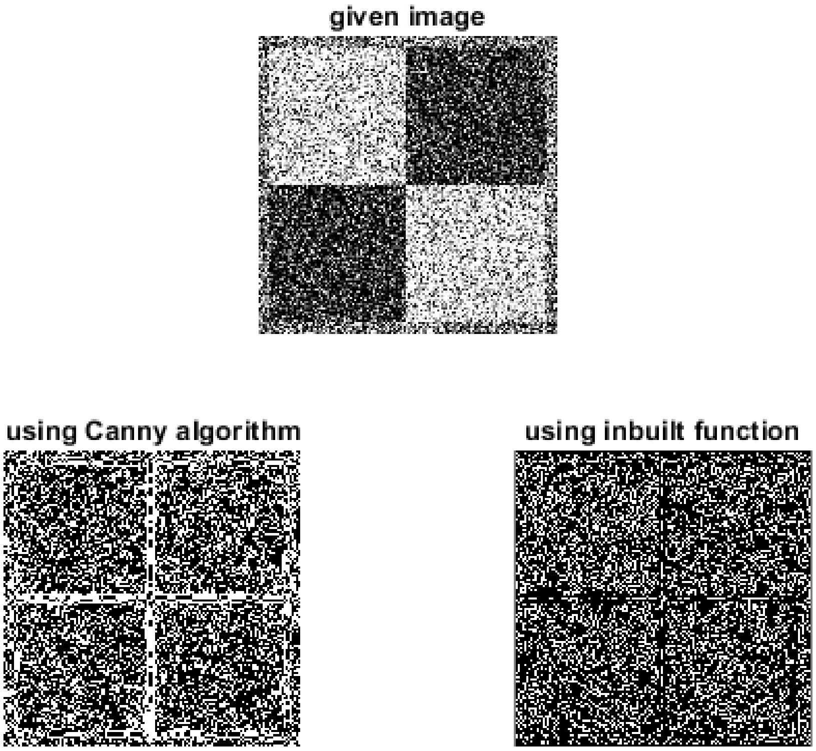
given image

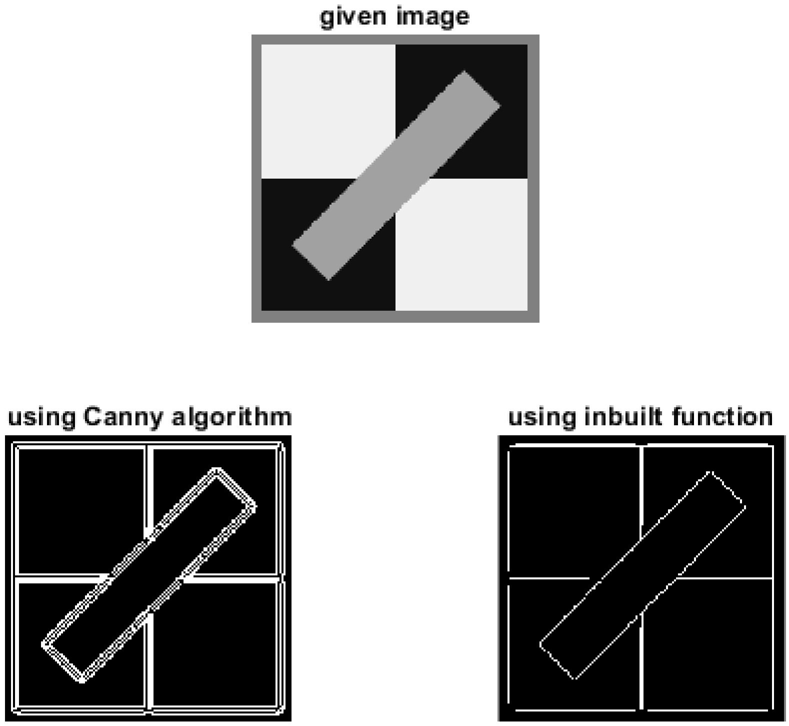


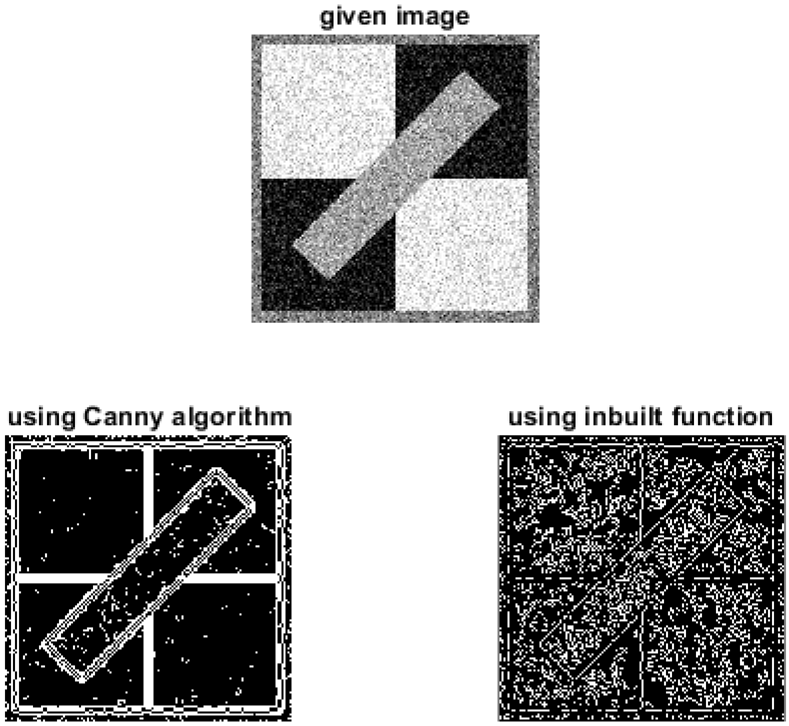
diagonal edges using Canny algorithm

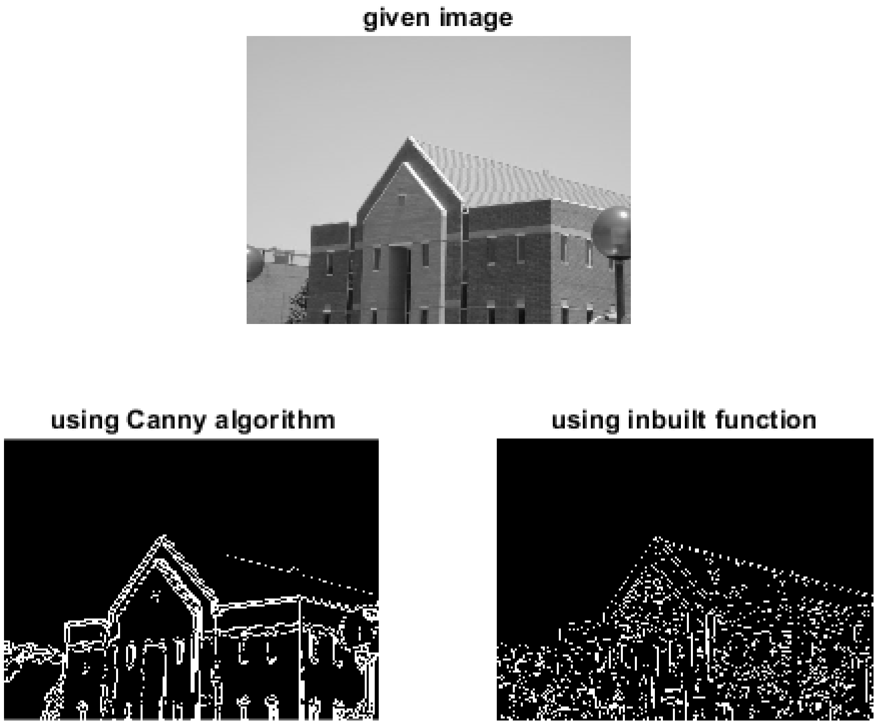
Comparison using inbuilt command:






given image
using Canny algorithm
using inbuilt function

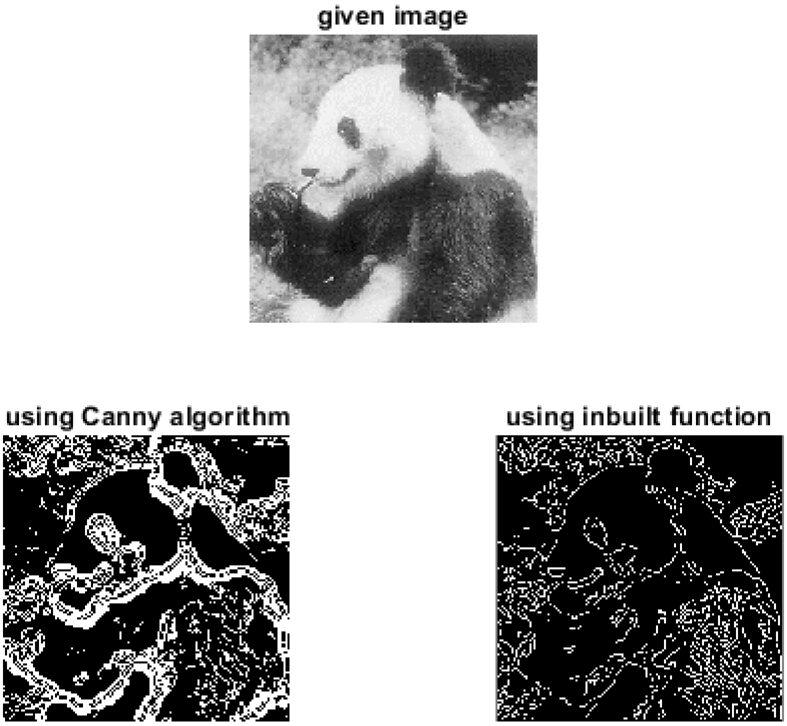

given image
using Canny algorithm
using inbuilt function



given image

using Canny algorithm

using inbuilt function



given image

using Canny algorithm

using inbuilt function

Inferences:

1) After assigning the angle of edges into different 8 bins with a each bin size of 20 degrees, we supress the non maximum pixel for the magnitude of gradients using the angle of gradients.
2) Then we perform the double thresholding using different threshold values.
3) We can get the perfect edge that depends on the selectin of the threshold two different values.
4) After double thresholding we perform hysteresis operation that will ensures the continuity in the edges in the image.
5) In canny edge detection , edge width is less compared to other edge detection techniques
6) Also edge detection is precise in canny edge detection.