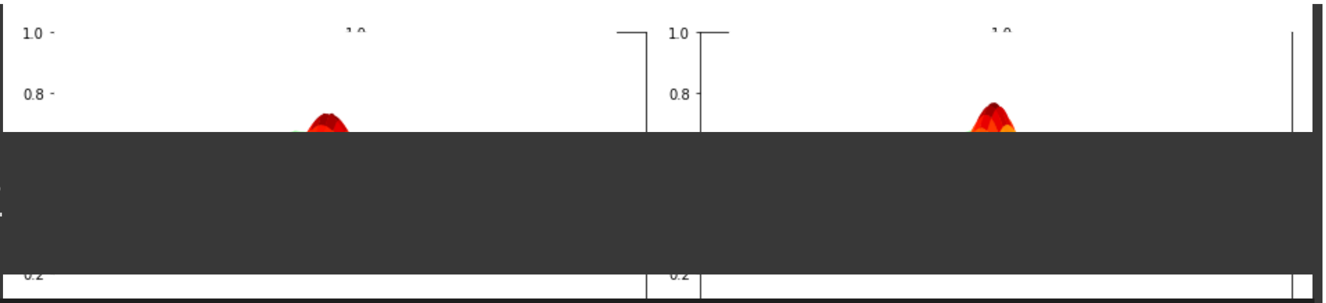
```
from google.colab import drive
drive.mount('/content/drive')
% cd '/content/drive/MyDrive/Academics/UoM-Course-Work/Semester-04/Computer-Vision/Inclass
! ls
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m
/content/drive/MyDrive/Academics/UoM-Course-Work/Semester-04/Computer-Vision/Inclass/
building.tif
```

```
import cv2 as cv
import matplotlib.pyplot as plt
from matplotlib import cm
import numpy as np
from skimage.feature import peak_local_max
```

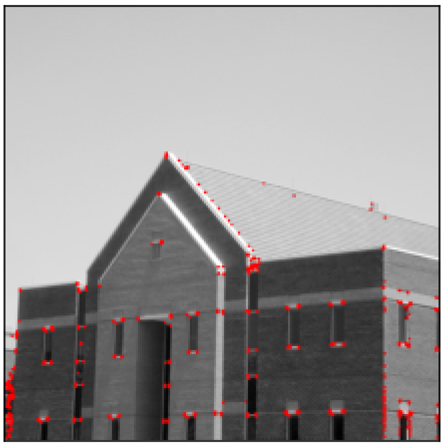
Q1

```
delta = 0.1
XX, YY = np.meshgrid(np.arange(-5, 5 + delta, delta), np.arange(-5, 5 + delta, delta))

sigma = 1
g = np.exp(-(XX**2 + YY**2)/(2*sigma**2))
g /= np.sum(g)
sobel_v = np.array([[-1,-2,-1],[0, 0, 0], [1, 2, 1]], dtype=np.float32)
g_x = cv.filter2D(g, -1, sobel_v)
sobel_h = np.array([[-1,0,1],[-2, 0, 2], [-1, 0, 1]], dtype=np.float32)
g_y = cv.filter2D(g, -1, sobel_h)

num_axs = 4
fig, ax = plt.subplots(1, num_axs, figsize=(4*num_axs, 4))
ax1 = fig.add_subplot(121, projection='3d')
ax2 = fig.add_subplot(122, projection='3d')

ax1.plot_surface(XX, YY, g_x, cmap=cm.jet, linewidth=0, antialiased=True)
ax2.plot_surface(XX, YY, g_y, cmap=cm.jet, linewidth=0, antialiased=True)

ax1.axis('off')
ax2.axis('off')

plt.show()
```



Q2

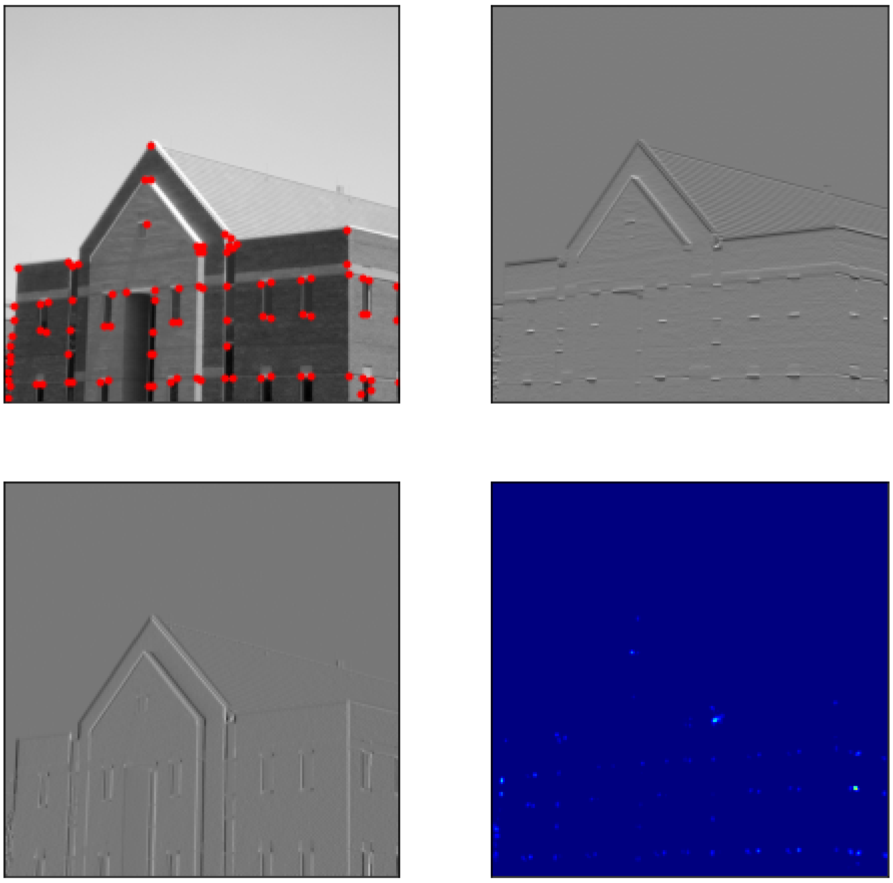
```
img = cv.imread('building.tif', cv.IMREAD_COLOR)
assert img is not None

gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
gray = np.float32(gray)
dst = cv.cornerHarris(gray, 2, 3, 0.04)

dst = cv.dilate(dst, None)
img[dst > 0.01*dst.max()] = [255, 0, 0]

plt.imshow(img)
plt.xticks([])
plt.yticks([])

plt.show()
```



Q3

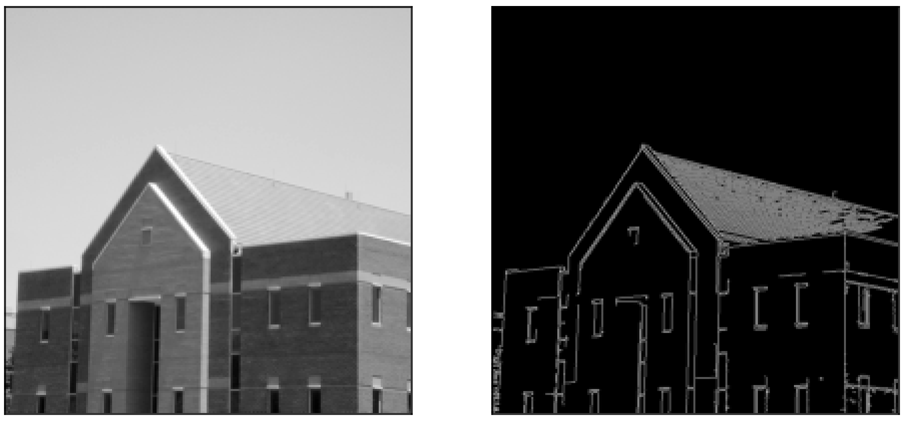
```
img = cv.imread('building.tif', cv.IMREAD_COLOR)
assert img is not None

I = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
I = np.float32(I)
sobel_v = np.array([[-1, -2, -1], [0, 0, 0], [1, 2, 1]])
sobel_h = np.array([[-1, 0, 1], [-2, 0, 2], [-1, 0, 1]])

Ix = cv.filter2D(I, -1, sobel_v)
Iy = cv.filter2D(I, -1, sobel_h)

sigma = 3
```

```
ksize = 7

m11 = cv.GaussianBlur(Ix*Ix, (ksize, ksize), sigma) # size of the image
m12 = cv.GaussianBlur(Ix*Iy, (ksize, ksize), sigma)
m21 = m12
m22 = cv.GaussianBlur(Iy*Iy, (ksize, ksize), sigma)

det = m11*m22 - m12*m21
trace = m11 + m22
alpha = 0.04
R = det - alpha*trace**2
R[R < 1e8] = 0
coordinates = peak_local_max(R, min_distance=2)

fig, ax = plt.subplots(2, 2, figsize=(8,8))

ax[0,0].imshow(img, cmap = 'gray')
ax[0,0].plot(coordinates[:, 1], coordinates[:, 0], 'r.')
ax[0,1].imshow(Ix + 127, cmap='gray')
ax[1,0].imshow(Iy + 127, cmap='gray')
ax[1,1].imshow(R + 127, cmap=cm.jet)

for i in range(2):
  for j in range(2):
    ax[i, j].set_xticks([])
    ax[i, j].set_yticks([])

plt.show()
```



Q4

```
img = cv.imread('building.tif', cv.IMREAD_GRAYSCALE)
assert img is not None

edges = cv.Canny(img, 100, 200)

fig, ax = plt.subplots(1,2, figsize=(8,4))

ax[0].imshow(img, cmap='gray')
ax[1].imshow(edges, cmap='gray')

for i in range(2):
  ax[i].set_xticks([])
  ax[i].set_yticks([])

plt.show()
```



✓ 0s completed at 2:57 PM