coursework_01

January 29, 2023

1 Coursework 1: Image filtering

In this coursework you will practice techniques for image filtering. The coursework includes coding questions and written questions. Please read both the text and the code in this notebook to get an idea what you are expected to implement.

1.1 What to do?

- Complete and run the code using `jupyter-lab` or `jupyter-notebook` to get the results.
- Export (File | Save and Export Notebook As…) the notebook as a PDF file, which contains your code, results and answers, and upload the PDF file onto Scientia.
- Instead of clicking the Export button, you can also run the following command instead: `jupyter nbconvert coursework_01_solution.ipynb --to pdf`
- If Jupyter complains about some problems in exporting, it is likely that pandoc (https://pandoc.org/installing.html) or latex is not installed, or their paths have not been included. You can install the relevant libraries and retry. Alternatively, use the Print function of your browser to export the PDF file.
- If Jupyter-lab does not work for you at the end (we hope not), you can use Google Colab to write the code and export the PDF file.

1.2 Dependencies:

You need to install Jupyter-Lab (https://jupyterlab.readthedocs.io/en/stable/getting_started/installation.html) and other libraries used in this coursework, such as by running the command: `pip3 install [package_name]`

[46]:

```
import sys
!{sys.executable} -m pip install imageio
!{sys.executable} -m pip install matplotlib
!{sys.executable} -m pip install scipy
!{sys.executable} -m pip install torch torchvision torchaudio
!{sys.executable} -m pip install nbconvert
```

```
Requirement already satisfied: imageio in
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages
(2.25.0)
Requirement already satisfied: pillow>=8.3.2 in
```

```
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages
(from imageio) (9.4.0)
Requirement already satisfied: numpy in
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages
(from imageio) (1.21.6)

[notice] A new release of pip
available: 22.3 -> 22.3.1
[notice] To update, run:
python3.7 -m pip install --upgrade pip
Requirement already satisfied: matplotlib in
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages
(3.5.3)
Requirement already satisfied: cycler>=0.10 in
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages
(from matplotlib) (0.11.0)
Requirement already satisfied: numpy>=1.17 in
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages
(from matplotlib) (1.21.6)
Requirement already satisfied: pyparsing>=2.2.1 in
/Users/ana_raicu/Library/Python/3.7/lib/python/site-packages (from matplotlib)
(3.0.9)
Requirement already satisfied: packaging>=20.0 in
/Users/ana_raicu/Library/Python/3.7/lib/python/site-packages (from matplotlib)
(21.3)
Requirement already satisfied: fonttools>=4.22.0 in
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages
(from matplotlib) (4.38.0)
Requirement already satisfied: python-dateutil>=2.7 in
/Users/ana_raicu/Library/Python/3.7/lib/python/site-packages (from matplotlib)
(2.8.2)
Requirement already satisfied: pillow>=6.2.0 in
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages
(from matplotlib) (9.4.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages
(from matplotlib) (1.4.4)
Requirement already satisfied: typing-extensions in
/Users/ana_raicu/Library/Python/3.7/lib/python/site-packages (from
kiwisolver>=1.0.1->matplotlib) (4.4.0)
Requirement already satisfied: six>=1.5 in
/Users/ana_raicu/Library/Python/3.7/lib/python/site-packages (from python-
dateutil>=2.7->matplotlib) (1.16.0)

[notice] A new release of pip
available: 22.3 -> 22.3.1
[notice] To update, run:
python3.7 -m pip install --upgrade pip
```

```
Requirement already satisfied: scipy in
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages
(1.7.3)
Requirement already satisfied: numpy<1.23.0,>=1.16.5 in
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages
(from scipy) (1.21.6)

[notice] A new release of pip
available: 22.3 -> 22.3.1
[notice] To update, run:
python3.7 -m pip install --upgrade pip
Requirement already satisfied: torch in
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages
(1.13.1)
Requirement already satisfied: torchvision in
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages
(0.14.1)
Requirement already satisfied: torchaudio in
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages
(0.13.1)
Requirement already satisfied: typing-extensions in
/Users/ana_raicu/Library/Python/3.7/lib/python/site-packages (from torch)
(4.4.0)
Requirement already satisfied: requests in
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages
(from torchvision) (2.28.2)
Requirement already satisfied: numpy in
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages
(from torchvision) (1.21.6)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages
(from torchvision) (9.4.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages
(from requests->torchvision) (1.26.14)
Requirement already satisfied: idna<4,>=2.5 in
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages
(from requests->torchvision) (3.4)
Requirement already satisfied: charset-normalizer<4,>=2 in
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages
(from requests->torchvision) (3.0.1)
Requirement already satisfied: certifi>=2017.4.17 in
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages
(from requests->torchvision) (2022.12.7)

[notice] A new release of pip
available: 22.3 -> 22.3.1
[notice] To update, run:
```

```
python3.7 -m pip install --upgrade pip
Collecting nbconvert
  Downloading nbconvert-7.2.9-py3-none-any.whl (274 kB)
                           274.9/274.9

kB 4.2 MB/s eta 0:00:00
Collecting tinycss2
  Using cached tinycss2-1.2.1-py3-none-any.whl (21 kB)
Collecting importlib-metadata>=3.6
  Using cached importlib_metadata-6.0.0-py3-none-any.whl (21 kB)
Collecting beautifulsoup4
  Using cached beautifulsoup4-4.11.1-py3-none-any.whl (128 kB)
Collecting bleach
  Downloading bleach-6.0.0-py3-none-any.whl (162 kB)
                           162.5/162.5

kB 6.2 MB/s eta 0:00:00
Requirement already satisfied: jupyter-core>=4.7 in
/Users/ana_raicu/Library/Python/3.7/lib/python/site-packages (from nbconvert)
(4.11.2)
Collecting jinja2>=3.0
  Using cached Jinja2-3.1.2-py3-none-any.whl (133 kB)
Requirement already satisfied: pygments>=2.4.1 in
/Users/ana_raicu/Library/Python/3.7/lib/python/site-packages (from nbconvert)
(2.13.0)
Collecting markupsafe>=2.0
  Downloading MarkupSafe-2.1.2-cp37-cp37m-macosx_10_9_x86_64.whl (13 kB)
Requirement already satisfied: traitlets>=5.0 in
/Users/ana_raicu/Library/Python/3.7/lib/python/site-packages (from nbconvert)
(5.5.0)
Collecting nbclient>=0.5.0
  Downloading nbclient-0.7.2-py3-none-any.whl (71 kB)
                           72.0/72.0 kB
2.7 MB/s eta 0:00:00
Collecting pandocfilters>=1.4.1
  Using cached pandocfilters-1.5.0-py2.py3-none-any.whl (8.7 kB)
Requirement already satisfied: packaging in
/Users/ana_raicu/Library/Python/3.7/lib/python/site-packages (from nbconvert)
(21.3)
Collecting defusedxml
  Using cached defusedxml-0.7.1-py2.py3-none-any.whl (25 kB)
Collecting nbformat>=5.1
  Downloading nbformat-5.7.3-py3-none-any.whl (78 kB)
                           78.1/78.1 kB
3.0 MB/s eta 0:00:00
Collecting mistune<3,>=2.0.3
  Using cached mistune-2.0.4-py2.py3-none-any.whl (24 kB)
Collecting jupyterlab-pygments
```

```
  Using cached jupyterlab_pygments-0.2.2-py2.py3-none-any.whl (21 kB)
Collecting zipp>=0.5
  Using cached zipp-3.12.0-py3-none-any.whl (6.6 kB)
Requirement already satisfied: typing-extensions>=3.6.4 in
/Users/ana_raicu/Library/Python/3.7/lib/python/site-packages (from importlib-
metadata>=3.6->nbconvert) (4.4.0)
Requirement already satisfied: jupyter-client>=6.1.12 in
/Users/ana_raicu/Library/Python/3.7/lib/python/site-packages (from
nbclient>=0.5.0->nbconvert) (7.4.7)
Collecting jupyter-core>=4.7
  Downloading jupyter_core-4.12.0-py3-none-any.whl (89 kB)
                           89.9/89.9 kB
3.2 MB/s eta 0:00:00
Collecting jsonschema>=2.6
  Downloading jsonschema-4.17.3-py3-none-any.whl (90 kB)
                           90.4/90.4 kB
3.4 MB/s eta 0:00:00
Collecting fastjsonschema
  Using cached fastjsonschema-2.16.2-py3-none-any.whl (22 kB)
Collecting soupsieve>1.2
  Using cached soupsieve-2.3.2.post1-py3-none-any.whl (37 kB)
Requirement already satisfied: six>=1.9.0 in
/Users/ana_raicu/Library/Python/3.7/lib/python/site-packages (from
bleach->nbconvert) (1.16.0)
Collecting webencodings
  Using cached webencodings-0.5.1-py2.py3-none-any.whl (11 kB)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/Users/ana_raicu/Library/Python/3.7/lib/python/site-packages (from
packaging->nbconvert) (3.0.9)
Collecting attrs>=17.4.0
  Downloading attrs-22.2.0-py3-none-any.whl (60 kB)
                           60.0/60.0 kB
2.0 MB/s eta 0:00:00
Collecting pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0
  Downloading pyrsistent-0.19.3-cp37-cp37m-macosx_10_9_x86_64.whl (69 kB)
                           69.8/69.8 kB
2.6 MB/s eta 0:00:00
Collecting importlib-resources>=1.4.0
  Downloading importlib_resources-5.10.2-py3-none-any.whl (34 kB)
Collecting pkgutil-resolve-name>=1.3.10
  Using cached pkgutil_resolve_name-1.3.10-py3-none-any.whl (4.7 kB)
Requirement already satisfied: pyzmq>=23.0 in
/Users/ana_raicu/Library/Python/3.7/lib/python/site-packages (from jupyter-
client>=6.1.12->nbclient>=0.5.0->nbconvert) (24.0.1)
Requirement already satisfied: tornado>=6.2 in
/Users/ana_raicu/Library/Python/3.7/lib/python/site-packages (from jupyter-
client>=6.1.12->nbclient>=0.5.0->nbconvert) (6.2)
Requirement already satisfied: python-dateutil>=2.8.2 in
```

```
/Users/ana_raicu/Library/Python/3.7/lib/python/site-packages (from jupyter-
client>=6.1.12->nbclient>=0.5.0->nbconvert) (2.8.2)
Requirement already satisfied: entrypoints in
/Users/ana_raicu/Library/Python/3.7/lib/python/site-packages (from jupyter-
client>=6.1.12->nbclient>=0.5.0->nbconvert) (0.4)
Requirement already satisfied: nest-asyncio>=1.5.4 in
/Users/ana_raicu/Library/Python/3.7/lib/python/site-packages (from jupyter-
client>=6.1.12->nbclient>=0.5.0->nbconvert) (1.5.6)
Installing collected packages: webencodings, mistune, fastjsonschema, zipp,
tinycss2, soupsieve, pyrsistent, pkgutil-resolve-name, pandocfilters,
markupsafe, jupyterlab-pygments, jupyter-core, defusedxml, bleach, attrs,
jinja2, importlib-resources, importlib-metadata, beautifulsoup4, jsonschema,
nbformat, nbclient, nbconvert
  Attempting uninstall: jupyter-core
    Found existing installation: jupyter_core 4.11.2
    Uninstalling jupyter_core-4.11.2:
      Successfully uninstalled jupyter_core-4.11.2
  WARNING: The scripts jupyter, jupyter-migrate and jupyter-troubleshoot
are installed in '/Library/Frameworks/Python.framework/Versions/3.7/bin' which
is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this
warning, use --no-warn-script-location.
  WARNING: The script jsonschema is installed in
'/Library/Frameworks/Python.framework/Versions/3.7/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this
warning, use --no-warn-script-location.
  WARNING: The script jupyter-trust is installed in
'/Library/Frameworks/Python.framework/Versions/3.7/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this
warning, use --no-warn-script-location.
  WARNING: The script jupyter-execute is installed in
'/Library/Frameworks/Python.framework/Versions/3.7/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this
warning, use --no-warn-script-location.
  WARNING: The scripts jupyter-dejavu and jupyter-nbconvert are
installed in '/Library/Frameworks/Python.framework/Versions/3.7/bin' which is
not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this
warning, use --no-warn-script-location.
Successfully installed attrs-22.2.0 beautifulsoup4-4.11.1 bleach-6.0.0
defusedxml-0.7.1 fastjsonschema-2.16.2 importlib-metadata-6.0.0 importlib-
resources-5.10.2 jinja2-3.1.2 jsonschema-4.17.3 jupyter-core-4.12.0 jupyterlab-
pygments-0.2.2 markupsafe-2.1.2 mistune-2.0.4 nbclient-0.7.2 nbconvert-7.2.9
nbformat-5.7.3 pandocfilters-1.5.0 pkgutil-resolve-name-1.3.10 pyrsistent-0.19.3
soupsieve-2.3.2.post1 tinycss2-1.2.1 webencodings-0.5.1 zipp-3.12.0

[notice] A new release of pip
available: 22.3 -> 22.3.1
[notice] To update, run:
python3.7 -m pip install --upgrade pip
```

[2]:

```python
# Import libaries (provided)
import imageio.v3 as imageio
import numpy as np
import matplotlib.pyplot as plt
import noise
import scipy
import scipy.signal
import math
import time
```

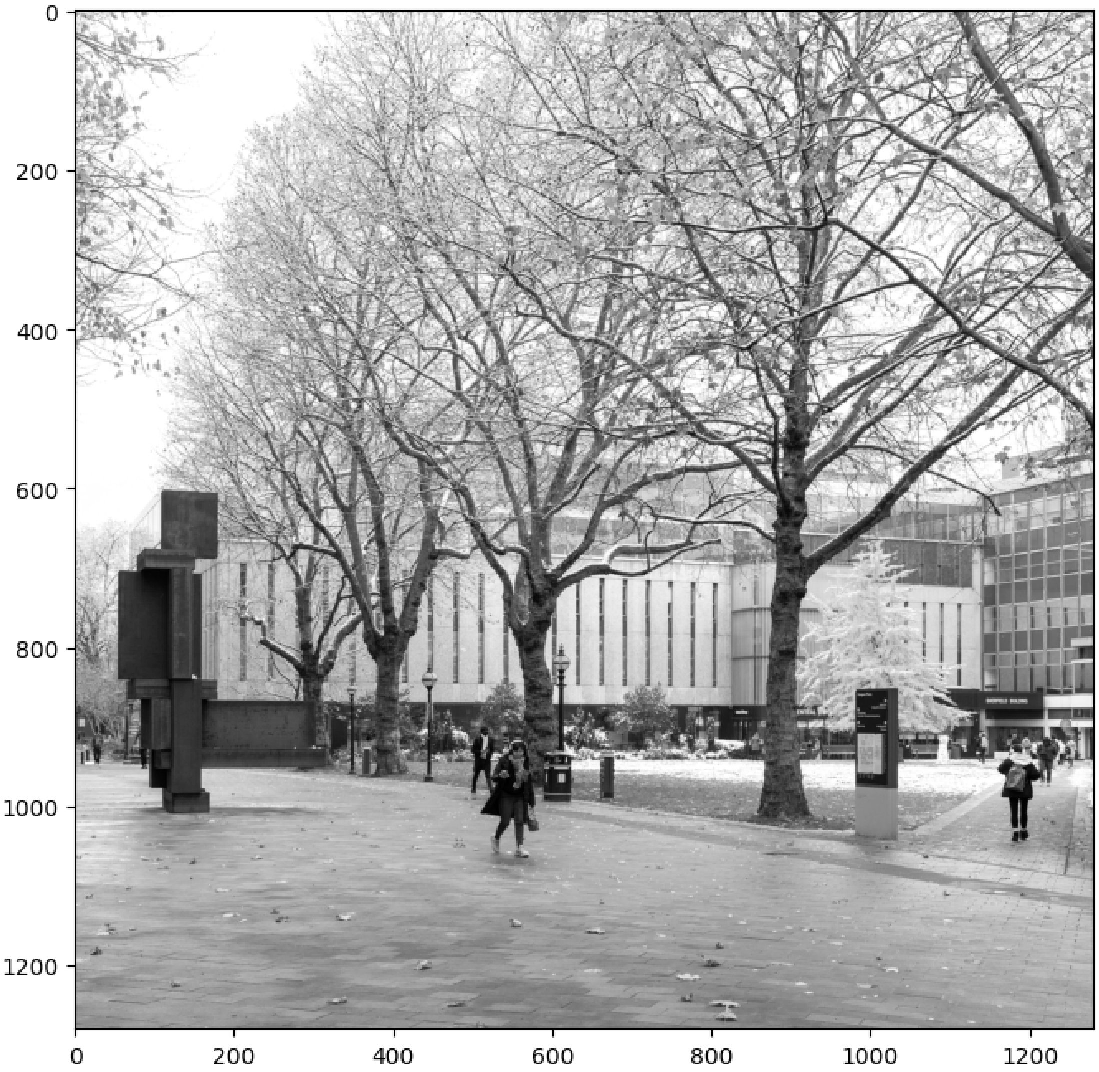
1.3 1. Moving average filter (20 points).

Read the provided input image, add noise to the image and design a moving average filter for denoising.

You are expected to design the kernel of the filter and then perform 2D image filtering using the function `scipy.signal.convolve2d()`.

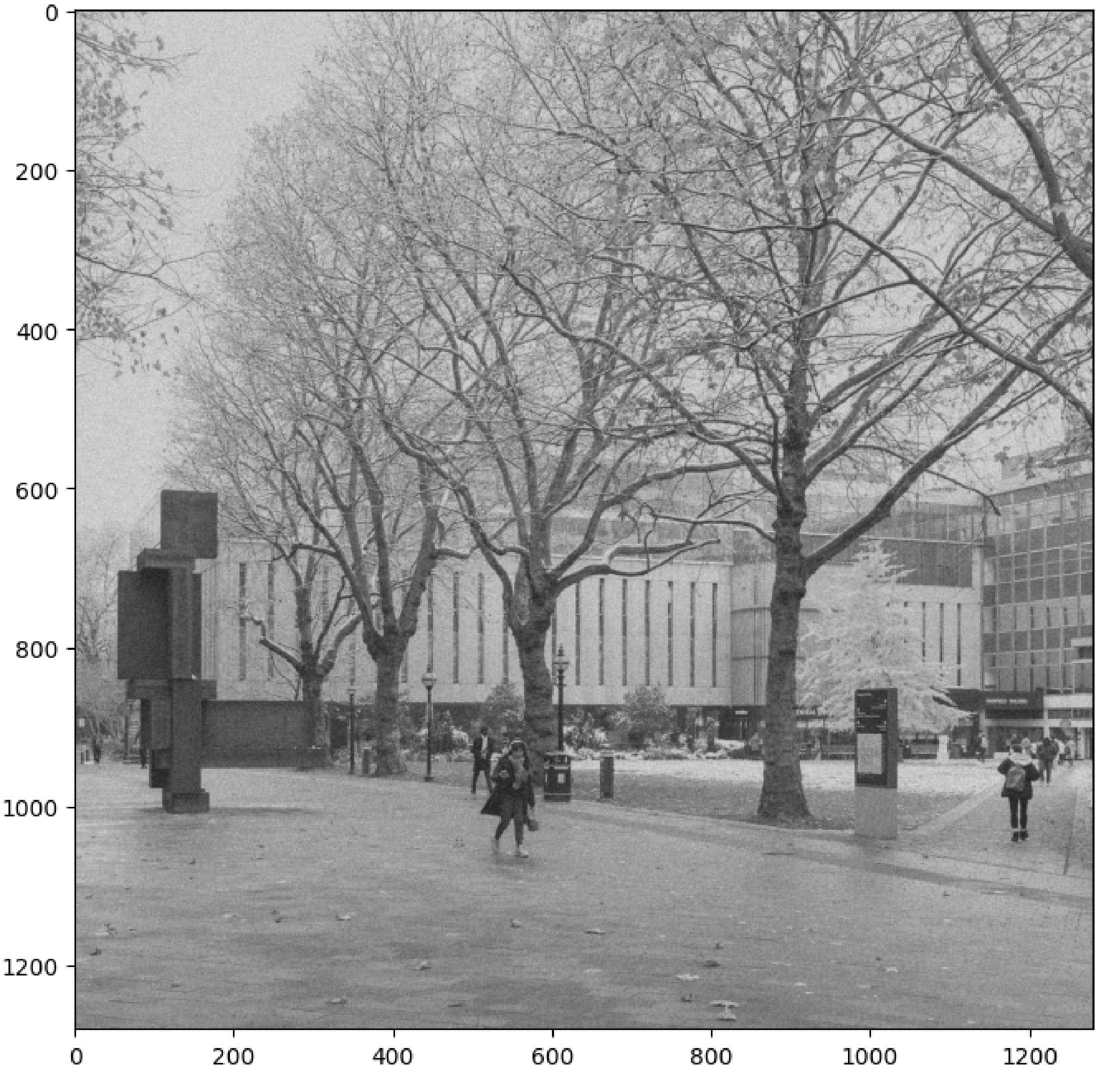
[3]:

```python
# Read the image (provided)
image = imageio.imread('campus_snow.jpg')
plt.imshow(image, cmap='gray')
plt.gcf().set_size_inches(8, 8)
```



[6]:

```python
# Corrupt the image with Gaussian noise (provided)
image_noisy = noise.add_noise(image, 'gaussian')
plt.imshow(image_noisy, cmap='gray')
plt.gcf().set_size_inches(8, 8)
```



1.3.1 Note: from now on, please use the noisy image as the input for the filters.

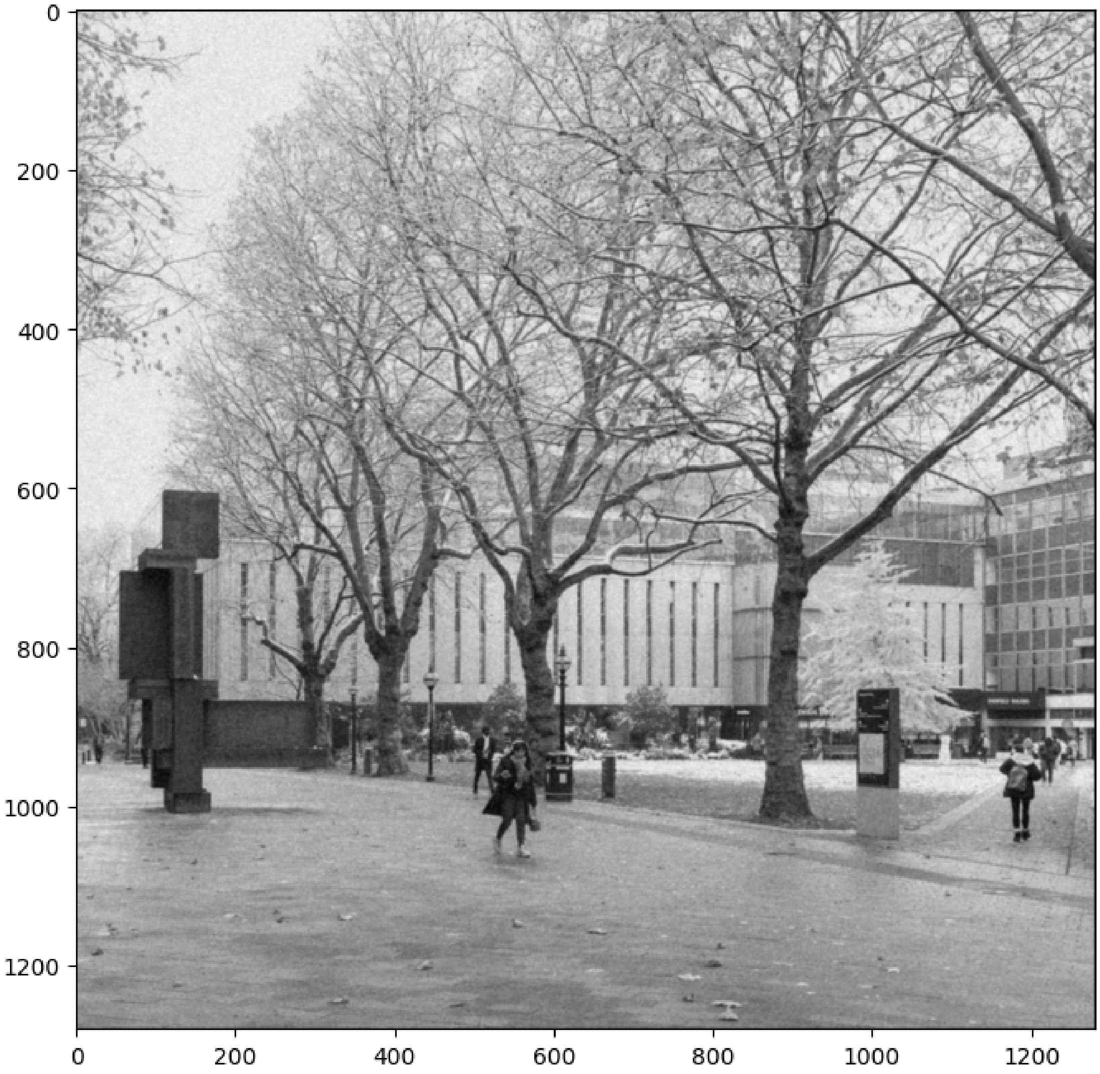
1.3.2 1.1 Filter the noisy image with a 3x3 moving average filter. Show the filtering results.

[7]:

```
# Design the filter h
### Insert your code ###
h = np.ones((3,3)) / 9

# Convolve the corrupted image with h using scipy.signal.convolve2d function
### Insert your code ###
image_filtered = scipy.signal.convolve2d(image_noisy, h, mode='same', boundary='fill', fillvalue=0)
```

```
# Print the filter (provided)
print('Filter h:')
print(h)

# Display the filtering result (provided)
plt.imshow(image_filtered, cmap='gray')
plt.gcf().set_size_inches(8, 8)
```

```
Filter h:
[[0.11111111 0.11111111 0.11111111]
 [0.11111111 0.11111111 0.11111111]
 [0.11111111 0.11111111 0.11111111]]
```

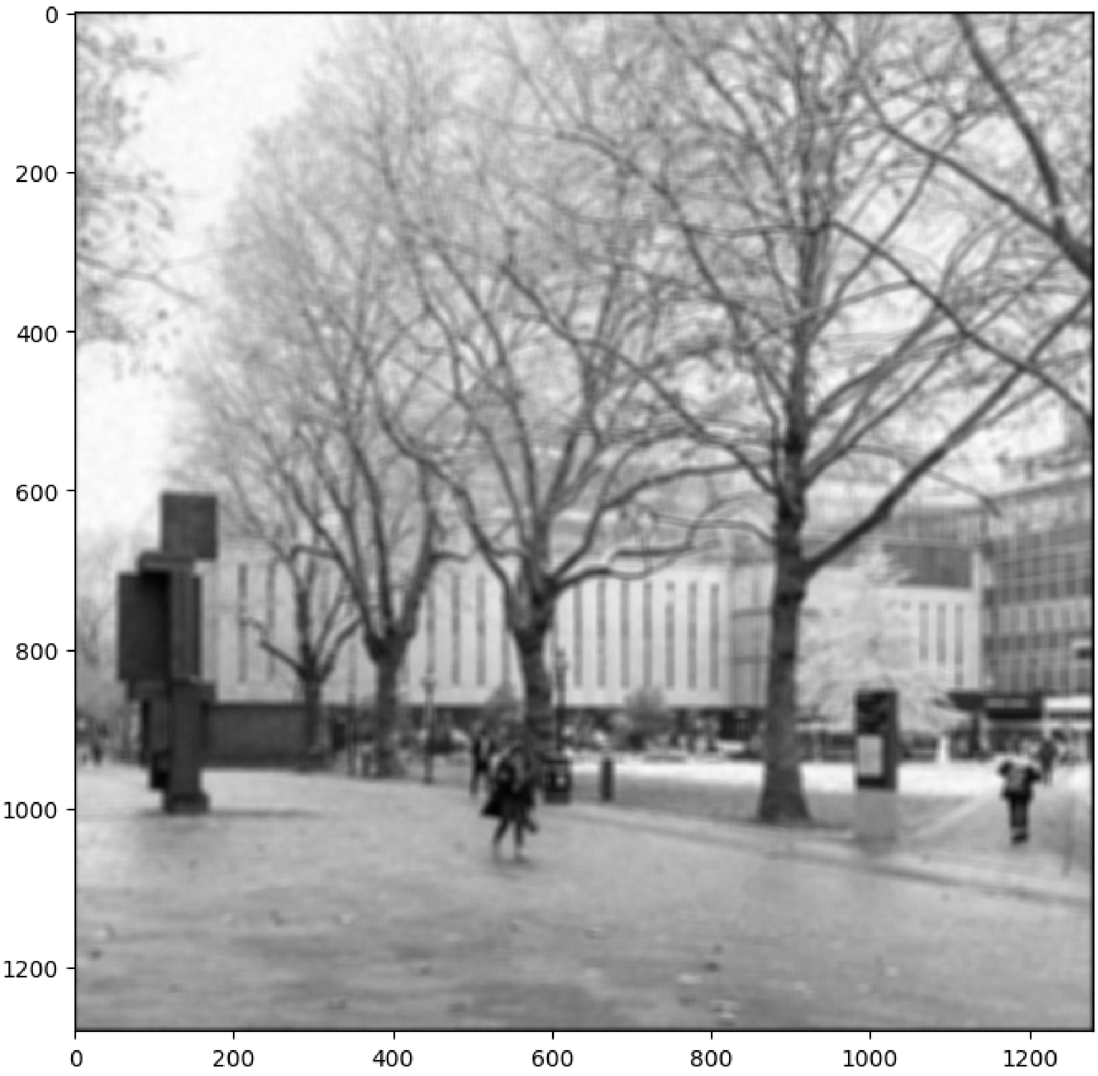


1.3.3 1.2 Filter the noisy image with a 11x11 moving average filter.

[8]:

```
# Design the filter h
### Insert your code ###
h = np.ones((11,11)) / 121

# Convolve the corrupted image with h using scipy.signal.convolve2d function
### Insert your code ###
image_filtered = scipy.signal.convolve2d(image_noisy, h, mode='same',␣
 ↪boundary='fill', fillvalue=0)

# Print the filter (provided)
print('Filter h:')
print(h)

# Display the filtering result (provided)
plt.imshow(image_filtered, cmap='gray')
plt.gcf().set_size_inches(8, 8)
```

```
Filter h:
[[0.00826446 0.00826446 0.00826446 0.00826446 0.00826446 0.00826446
  0.00826446 0.00826446 0.00826446 0.00826446 0.00826446]
 [0.00826446 0.00826446 0.00826446 0.00826446 0.00826446 0.00826446
  0.00826446 0.00826446 0.00826446 0.00826446 0.00826446]
 [0.00826446 0.00826446 0.00826446 0.00826446 0.00826446 0.00826446
  0.00826446 0.00826446 0.00826446 0.00826446 0.00826446]
 [0.00826446 0.00826446 0.00826446 0.00826446 0.00826446 0.00826446
  0.00826446 0.00826446 0.00826446 0.00826446 0.00826446]
 [0.00826446 0.00826446 0.00826446 0.00826446 0.00826446 0.00826446
  0.00826446 0.00826446 0.00826446 0.00826446 0.00826446]
 [0.00826446 0.00826446 0.00826446 0.00826446 0.00826446 0.00826446
  0.00826446 0.00826446 0.00826446 0.00826446 0.00826446]
 [0.00826446 0.00826446 0.00826446 0.00826446 0.00826446 0.00826446
  0.00826446 0.00826446 0.00826446 0.00826446 0.00826446]
 [0.00826446 0.00826446 0.00826446 0.00826446 0.00826446 0.00826446
  0.00826446 0.00826446 0.00826446 0.00826446 0.00826446]
 [0.00826446 0.00826446 0.00826446 0.00826446 0.00826446 0.00826446
  0.00826446 0.00826446 0.00826446 0.00826446 0.00826446]
 [0.00826446 0.00826446 0.00826446 0.00826446 0.00826446 0.00826446
  0.00826446 0.00826446 0.00826446 0.00826446 0.00826446]
 [0.00826446 0.00826446 0.00826446 0.00826446 0.00826446 0.00826446
  0.00826446 0.00826446 0.00826446 0.00826446 0.00826446]]
```



1.3.4 1.3 Comment on the filtering results. How do different kernel sizes influence the filtering results?

[29]:

```
### Insert your answer ###
# The kernel size represents the window size of pixels used
# for calculating the average value of each pixel of the output image.
# Hence modifying the kernel size indicates the smoothness of the image.
# Increasing the kernel size results in a more smoothed image,
# since the "window" size is bigger, meaning that more neighbouring pixels
# are being considered.
# In the example from 1.1 we can notice that some noise
# has been reduced from its input. However, in the example from 1.2, the significantly
# increased window size results in a very blurred image, where details can no␣
 ↪longer be easily observed.
```

1.4 2. Edge detection (56 points).

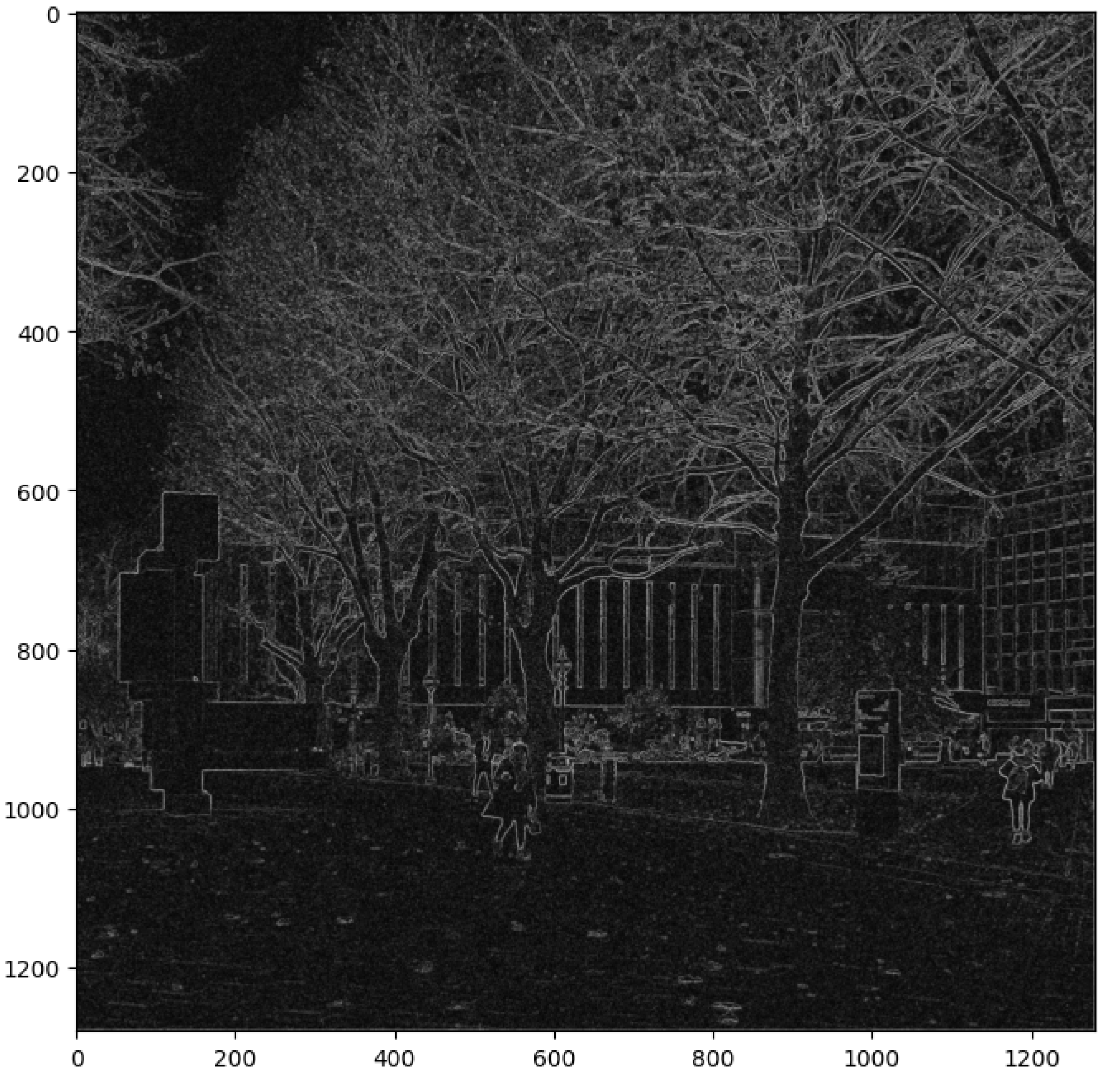
Perform edge detection using Sobel filtering, as well as Gaussian + Sobel filtering.

1.4.1 2.1 Implement 3x3 Sobel filters and convolve with the noisy image.

[9]:

```
# Design the filters
### Insert your code ###
sobel_x = np.array([[1, 0, -1], [2, 0, -2], [1, 0, -1]])
sobel_y = np.array([[1, 2, 1], [0, 0, 0], [-1, -2, -1]])

# Image filtering
image_sobel_x = scipy.signal.convolve2d(image_noisy, sobel_x, mode='same',␣
 ↪boundary='fill', fillvalue=0)
image_sobel_y = scipy.signal.convolve2d(image_noisy, sobel_y, mode='same',␣
 ↪boundary='fill', fillvalue=0)

# Calculate the gradient magnitude
grad_mag = np.sqrt(np.square(image_sobel_x) + np.square(image_sobel_y))

# Print the filters (provided)
print('sobel_x:')
print(sobel_x)
print('sobel_y:')
print(sobel_y)

# Display the magnitude map (provided)
plt.imshow(grad_mag, cmap='gray')
plt.gcf().set_size_inches(8, 8)
```

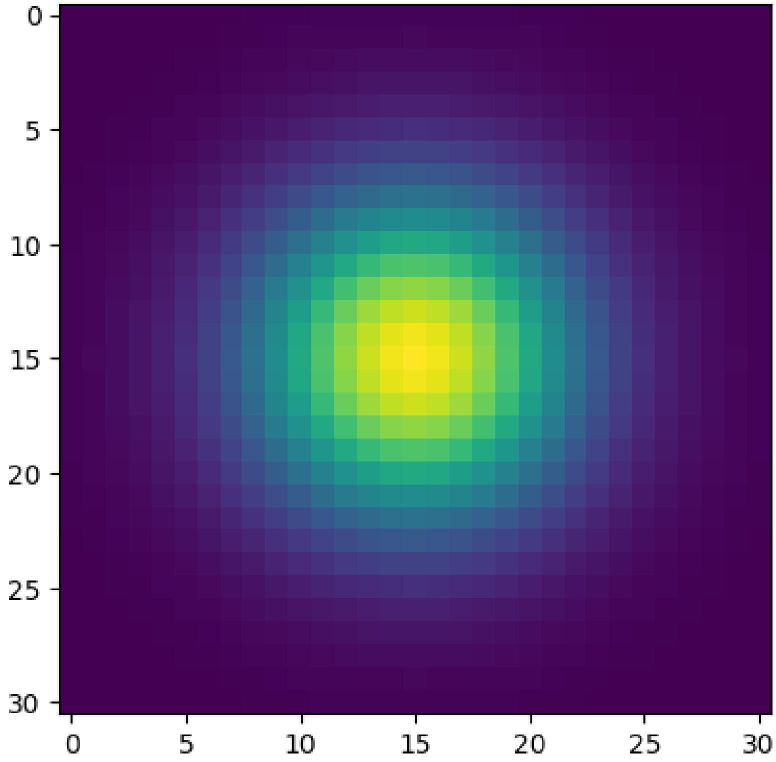
```
sobel_x:
[[ 1  0 -1]
 [ 2  0 -2]
 [ 1  0 -1]]
sobel_y:
[[ 1  2  1]
 [ 0  0  0]
 [-1 -2 -1]]
```



1.4.2 2.2 Implement a function that generates a 2D Gaussian filter given the parameter σ.

[10]:
```
# Design the Gaussian filter
def gaussian_filter_2d(sigma):
    # sigma: the parameter sigma in the Gaussian kernel (unit: pixel)
    #
    # return: a 2D array for the Gaussian kernel

    ### Insert your code ###
    x, y = np.mgrid[-3*sigma:3*sigma+1, -3*sigma:3*sigma+1]
    h = 1/np.sqrt((2*np.pi)*sigma) * np.exp(-(x**2 + y**2)/(2*sigma**2))
    h = h / h.sum()
    return h

# Visualise the Gaussian filter when sigma = 5 pixel (provided)
sigma = 5
h = gaussian_filter_2d(sigma)
plt.imshow(h)
```

[10]:
```
<matplotlib.image.AxesImage at 0x7fa666f02a90>
```

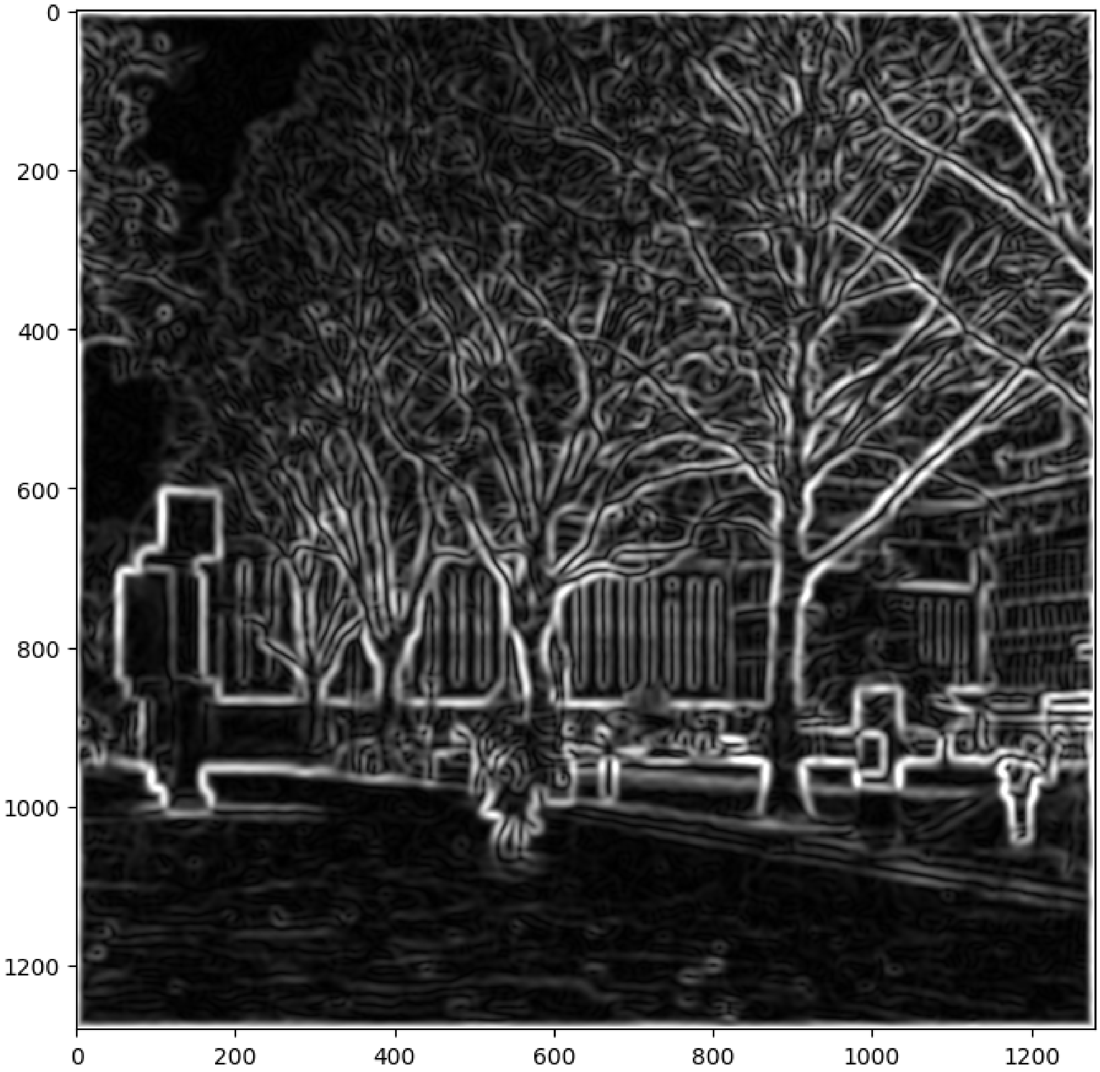


1.4.3 2.3 Perform Gaussian smoothing ($\sigma = 5$ pixels) and evaluate the computational time for Gaussian smoothing. After that, perform Sobel filtering and show the gradient magintude map.

[11]:
```
# Construct the Gaussian filter
sigma = 5
h = gaussian_filter_2d(sigma)

# Perform Gaussian smoothing and count time
start = time.time()
image_gaussian = scipy.signal.convolve2d(image_noisy, h, mode='same',␣
 ↪boundary='fill', fillvalue=0)
end = time.time()
```

```
elapsed = end - start
print("Gaussian filtering time: ")
print(elapsed)

# Image filtering
image_sobel_x = scipy.signal.convolve2d(image_gaussian, sobel_x, mode='same',␣
 ↪boundary='fill', fillvalue=0)
image_sobel_y = scipy.signal.convolve2d(image_gaussian, sobel_y, mode='same',␣
 ↪boundary='fill', fillvalue=0)

# Calculate the gradient magnitude
grad_mag = np.sqrt(np.square(image_sobel_x) + np.square(image_sobel_y))

# Display the gradient magnitude map (provided)
plt.imshow(grad_mag, cmap='gray', vmin=0, vmax=100)
plt.gcf().set_size_inches(8, 8)
```

```
Gaussian filtering time:
2.740772247314453
```

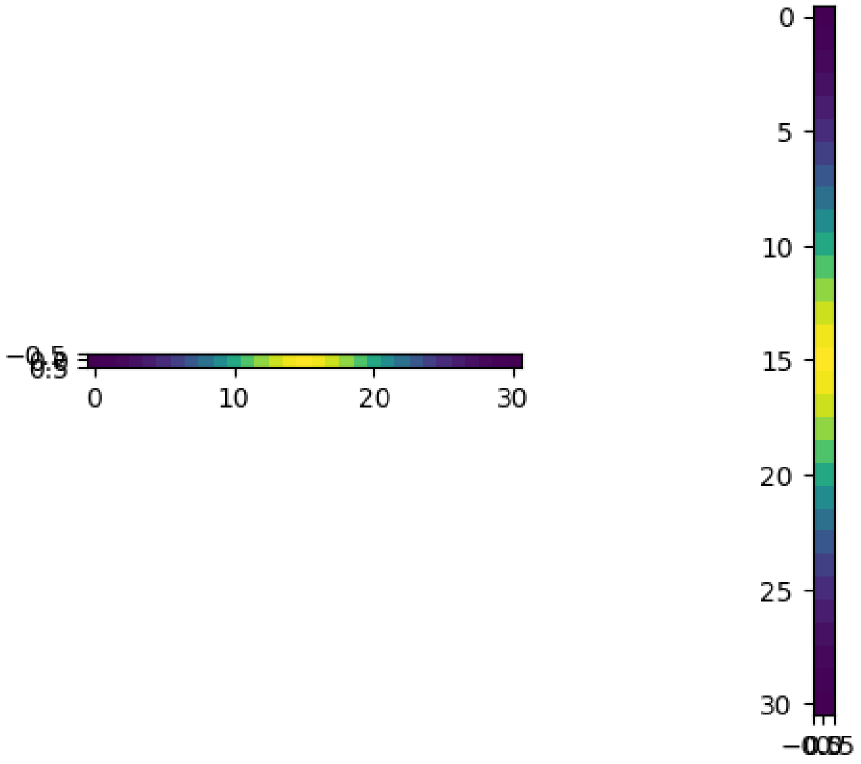


1.4.4 2.4 Implement a function that generates a 1D Gaussian filter given the parameter σ. Generate 1D Gaussian filters along x-axis and y-axis respectively.

[21]:
```
# Design the Gaussian filter
def gaussian_filter_1d(sigma):
    # sigma: the parameter sigma in the Gaussian kernel (unit: pixel)
    #
    # return: a 1D array for the Gaussian kernel

    ### Insert your code ###
    grid = np.mgrid[-3*sigma:3*sigma+1]
    h = h = 1/np.sqrt((2*np.pi)*sigma) * np.exp(-(grid**2)/(2*sigma**2))
    return h
```

```
# sigma = 5 pixel (provided)
sigma = 5

# The Gaussian filter
gaussian_filter = gaussian_filter_1d(sigma)

# The Gaussian filter along x-axis. Its shape is (1, sz).
### Insert your code ###
h_x = np.expand_dims(gaussian_filter, 0)
# np.expand_dims(np.expand_dims(noisy_image, 0), 0)
# The Gaussian filter along y-axis. Its shape is (sz, 1).
### Insert your code ###
h_y = np.expand_dims(gaussian_filter, 1)

# Visualise the filters (provided)
plt.subplot(1, 2, 1)
plt.imshow(h_x)
plt.subplot(1, 2, 2)
plt.imshow(h_y)
```

```
[21]: <matplotlib.image.AxesImage at 0x7fa66d557ed0>
```

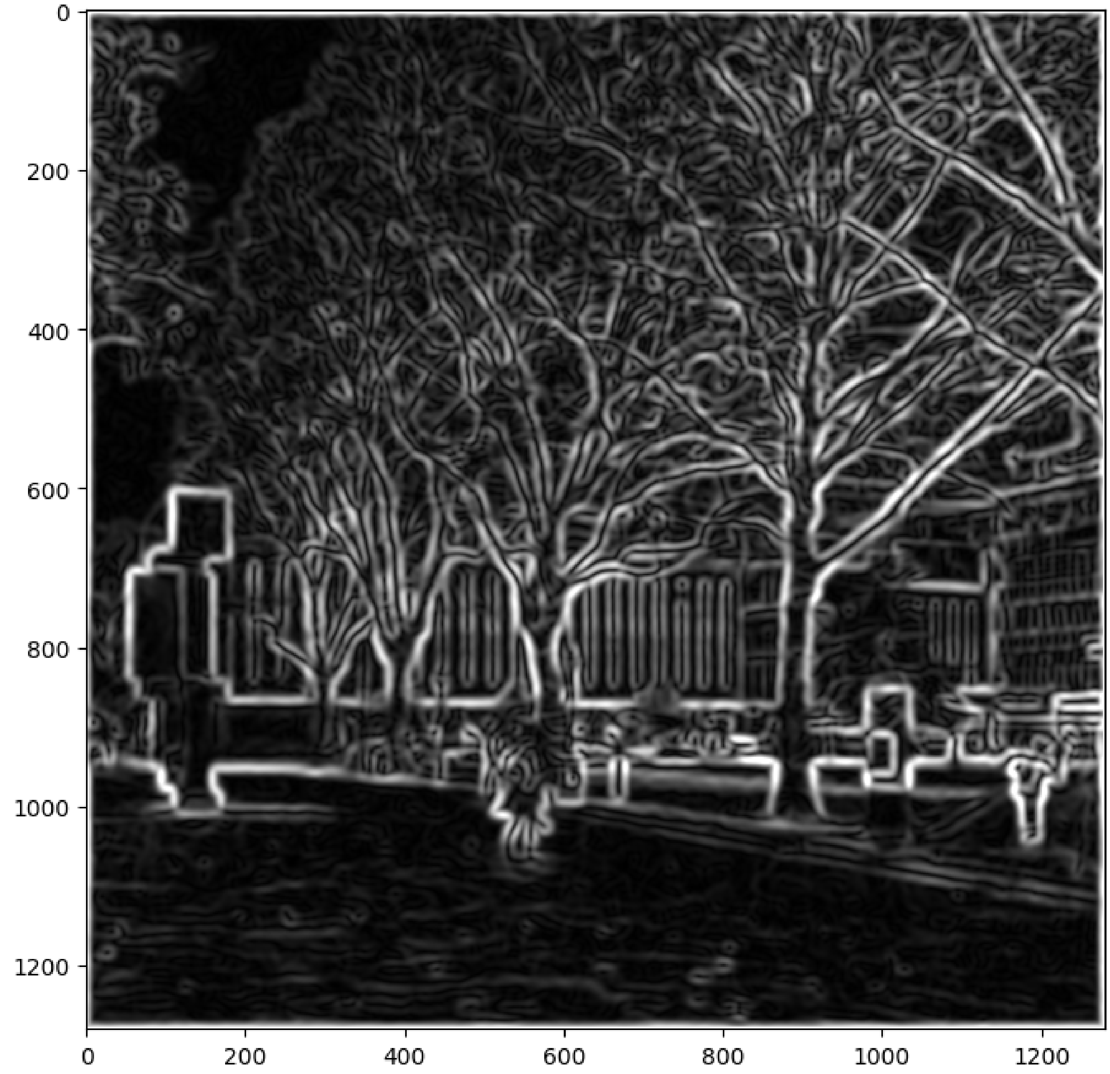


1.4.5 2.6 Perform Gaussian smoothing ($\sigma = 5$ pixels) using two separable filters and evaluate the computational time for separable Gaussian filtering. After that, perform Sobel filtering, show the gradient magnitude map and check whether it is the same as the previous one without separable filtering.

[13]:

```
# Perform separable Gaussian smoothing and count time
### Insert your code ###

# Image filtering
### Insert your code ###
start = time.time()
image_gaussian_x = scipy.signal.convolve(image_noisy, h_x)
image_gaussian_y = scipy.signal.convolve(image_noisy, h_y)
end = time.time()
elapsed = end - start
print("Gaussian filtering time with separable filters: ")
print(elapsed)

# Calculate the gradient magnitude
### Insert your code ###
image_sobel_x = scipy.signal.convolve2d(image_gaussian, sobel_x, mode='same',␣
 ↪boundary='fill', fillvalue=0)
image_sobel_y = scipy.signal.convolve2d(image_gaussian, sobel_y, mode='same',␣
 ↪boundary='fill', fillvalue=0)
grad_mag2 = np.sqrt(np.square(image_sobel_x) + np.square(image_sobel_y))

# Display the gradient magnitude map (provided)
plt.imshow(grad_mag2, cmap='gray', vmin=0, vmax=100)
plt.gcf().set_size_inches(8, 8)

# Check the difference between the current gradient magnitude map
# and the previous one produced without separable filtering. You
# can report the mean difference between the two.
### Insert your code ###
```

```
Gaussian filtering time with separable filters:
0.09035110473632812
```



1.4.6 2.7 Comment on the Gaussian + Sobel filtering results and the computational time.

Insert your answer ### Filtering with Gaussian and Sobel filters involve smoothing the image and emphasizing the edges detected in the image, respectively. Hence applying them both, one after another, returns an image with well dfeined edges and smoothened regions because noise in the gradient magnitude gets surpressed by Gaussian filtering. When applying Gaussian filtering in 2D, then computation time(~3.007) is significantly higher than the computation time observed in separable Gaussian filtering(~0.093). That is because the 2D Gaussian filter is being applied at each pixel of the photo, performing an overall large number of dot product. With separable filtering, we create 2 1D filters (one on the x-axis and one on the y-axis) each of which is being applied only once over each row/column, resulting in a much smaller number of dot product operations being performed. Another thing to note is that applying 2D filtering is much more precise than separable filtering since each operation is occuring on the pixel level. Hence the picture obtained through

2D Gaussian might be more precise and smoothened compared to the other. Using 2D might be preffered when we care less about computation time and more about image accuracy.

1.5 3. Challenge: Implement 2D image filters using Pytorch (24 points).

Pytorch is a machine learning framework that supports filtering and convolution.

The Conv2D operator takes an input array of dimension NxC1xXxY, applies the filter and outputs an array of dimension NxC2xXxY. Here, since we only have one image with one colour channel, we will set N=1, C1=1 and C2=1. You can read the documentation of Conv2D for more detail.

[14]:

```
# Import libaries (provided)
import torch
```

1.5.1 3.1 Expand the dimension of the noisy image into 1x1xXxY and convert it to a Pytorch tensor.

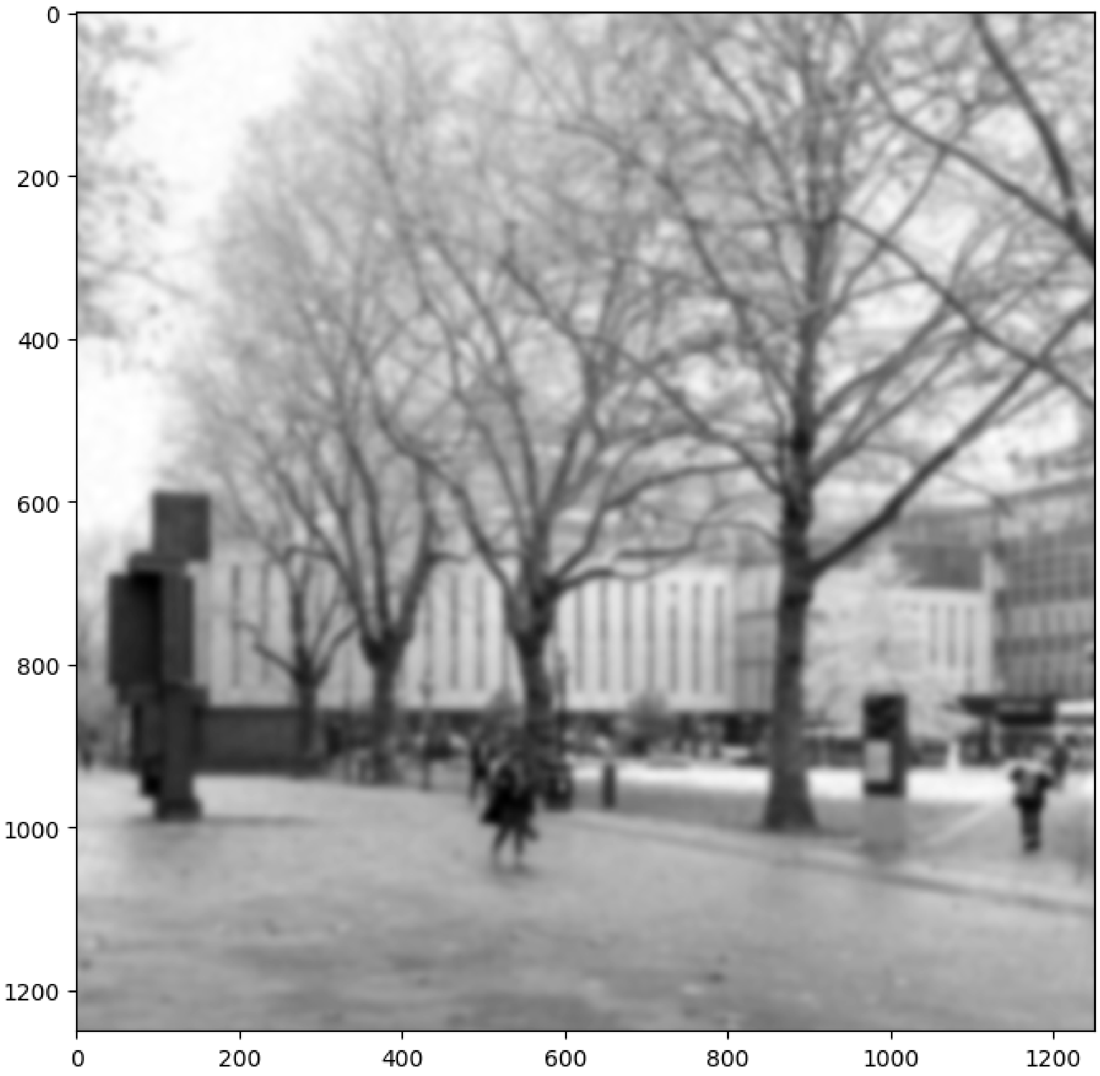
[33]:

```
# Expand the dimension of the numpy array
### Insert your code ###
image_noisy_expanded = np.expand_dims(np.expand_dims(image_noisy, 0), 0)

# Convert to a Pytorch tensor using torch.from_numpy
image_noisy_torch = torch.from_numpy(np.float32(image_noisy_expanded)
```

1.5.2 3.2 Create a Pytorch Conv2D filter, set its kernel to be a 2D Gaussian filter and perform filtering.

[37]:

```
# A 2D Gaussian filter when sigma = 5 pixel (provided)
sigma = 5
h = gaussian_filter_2d(sigma)

# Create the Conv2D filter
conv_filter = torch.nn.Conv2d(in_channels=1, out_channels=1, kernel_size=sigma)

# Setting the kernel
kernel = np.array([[h]])
conv_filter.weight.data = torch.from_numpy(np.float32(kernel))

# Filtering
image_filtered = conv_filter(image_noisy_torch).detach().numpy()[0][0]

# Display the filtering result (provided)
plt.imshow(image_filtered, cmap='gray')
plt.gcf().set_size_inches(8, 8)
```

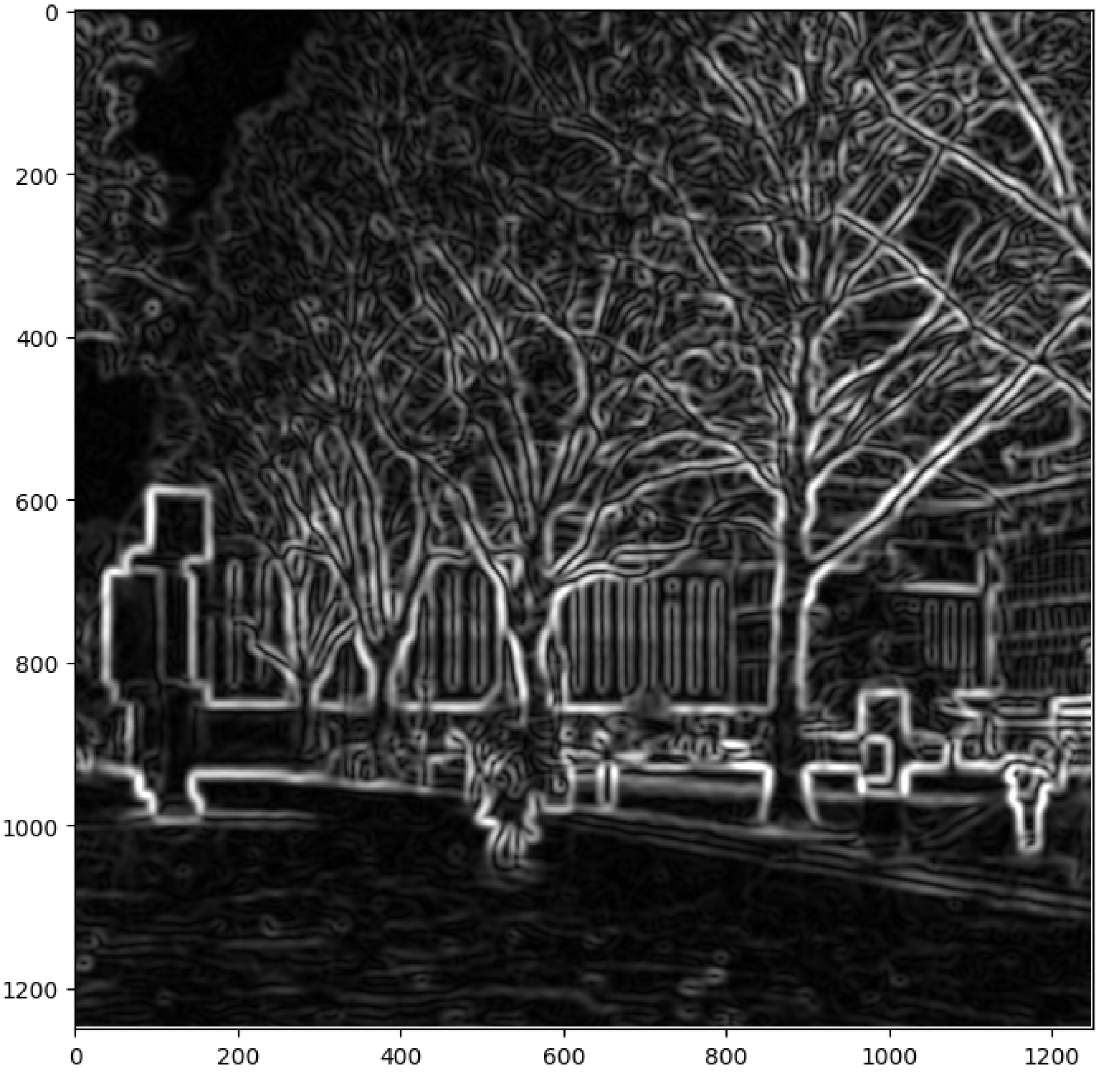


1.5.3 3.3 Implement Pytorch Conv2D filters to perform Sobel filtering on Gaussian smoothed images, show the gradient magnitude map.

[45]:

```
sigma = 5
h = gaussian_filter_2d(sigma)
# Create Conv2D filters
### Insert your code ###
conv_filter = torch.nn.Conv2d(in_channels=1, out_channels=1, kernel_size=sigma)

# Perform filtering
kernel = np.array([[h]])
conv_filter.weight.data = torch.from_numpy(np.float32(kernel))
image_filtered = conv_filter(image_noisy_torch)
```

```
# Calculate the gradient magnitude map
### Insert your code ###
sobel_x = torch.tensor([[1, 0, -1], [2, 0, -2], [1, 0, -1]], dtype=torch.
 ↪float32)
sobel_y = torch.tensor([[1, 2, 1], [0, 0, 0], [-1, -2, -1]], dtype=torch.
 ↪float32)
conv_x = torch.nn.Conv2d(1, 1, 3, padding=1, bias=False)
conv_x.weight.data = sobel_x.repeat(1, 1, 1, 1)
conv_y = torch.nn.Conv2d(1, 1, 3, padding=1, bias=False)
conv_y.weight.data = sobel_y.repeat(1, 1, 1, 1)
grad_mag3 = torch.sqrt(conv_x(image_filtered)**2 + conv_y(image_filtered)**2).
 ↪detach().numpy()[0][0]

# Visualise the gradient magnitude map (provided)
plt.imshow(grad_mag3, cmap='gray', vmin=0, vmax=100)
plt.gcf().set_size_inches(8, 8)
```


0
200
400
600
800
1000
1200
0
200
400
600
800
1000
1200

[ ]: