Assignment3

March 26, 2019

1 Name: Li Gen

2 Student ID: 2017210037

3 Assignment 02

4 Link: https://github.com/ligen0423/Ass3.git

```
In [1]: import matplotlib.pyplot as plt
        import numpy as np
        from skimage import io,color
        from skimage import exposure
        import torch
        import torch.nn as nn
```

5 Uploading the image data and transform into gray image data

```
In [2]: file_name = 'cau-1.jpg'
        im_color = io.imread(file_name)
        im_gray = color.rgb2gray(im_color) #print(im_color)
        print(im_gray)
```

```
[[0.4203251  0.42424667 0.42424667 ... 0.65200824 0.65200824 0.65200824]
 [0.41640353 0.4203251  0.42424667 ... 0.65845255 0.65845255 0.65845255]
 [0.41640353 0.4203251  0.42424667 ... 0.66237412 0.66237412 0.66237412]
 ...
 [0.1039698  0.10202039 0.10986353 ... 0.10981961 0.11179176 0.10338314]
 [0.09978039 0.09978039 0.10484078 ... 0.10533255 0.09946157 0.09161843]
 [0.10370196 0.1115451  0.10540627 ... 0.09748941 0.08769686 0.10281765]]
```

6 Creating the alpha map

```
In [3]: alpha = np.ones((im_gray.shape))

        im_gray.shape
```

```
Out[3]: (102, 150)
```

7 Mutiphy alpha and image data

```
In [4]: alpha[:,:] = 0.35
        alpha[30:70,50:100] = 0.4
        alpha[35,51]
        im_gray = im_gray * alpha
        im_gray = im_gray.reshape(1,1,102,150)
        print(im_gray)
```

```
[[[[0.14711378 0.14848633 0.14848633 ... 0.22820288 0.22820288
    0.22820288]
   [0.14574124 0.14711378 0.14848633 ... 0.23045839 0.23045839
    0.23045839]
   [0.14574124 0.14711378 0.14848633 ... 0.23183094 0.23183094
    0.23183094]
   ...
   [0.03638943 0.03570714 0.03845224 ... 0.03843686 0.03912712
    0.0361841 ]
   [0.03492314 0.03492314 0.03669427 ... 0.03686639 0.03481155
    0.03206645]
   [0.03629569 0.03904078 0.0368922  ... 0.03412129 0.0306939
    0.03598618]]]]
```

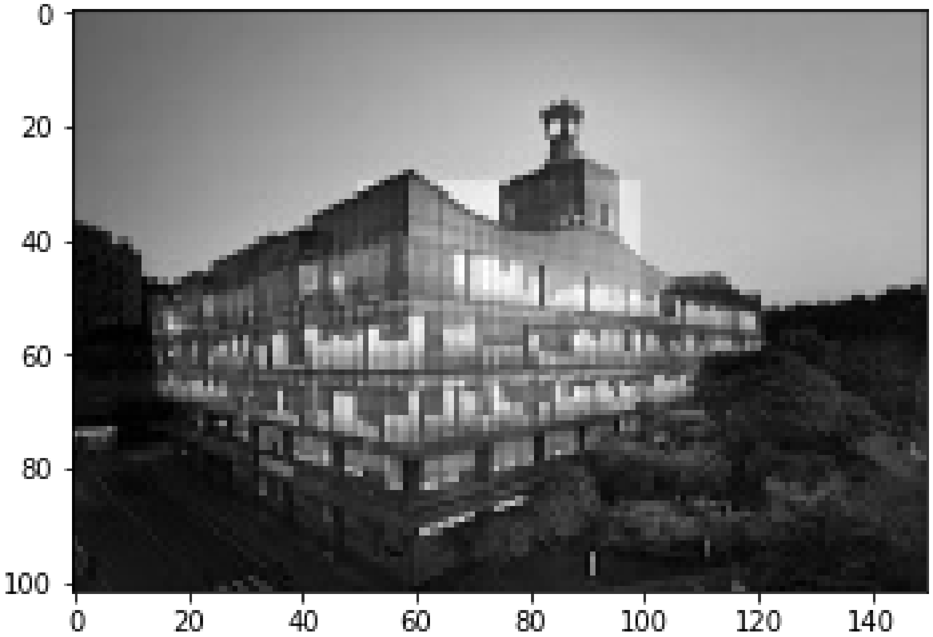
```
In [5]: t = 1/2
        #ker = np.array(([0,t,0],[t,4*t,t],[0,t,0]))
        #ker = torch.Tensor(([0,t,0],[t,4*t,t],[0,t,0]))
        #im_gray = torch.FloatTensor(im_gray)
```

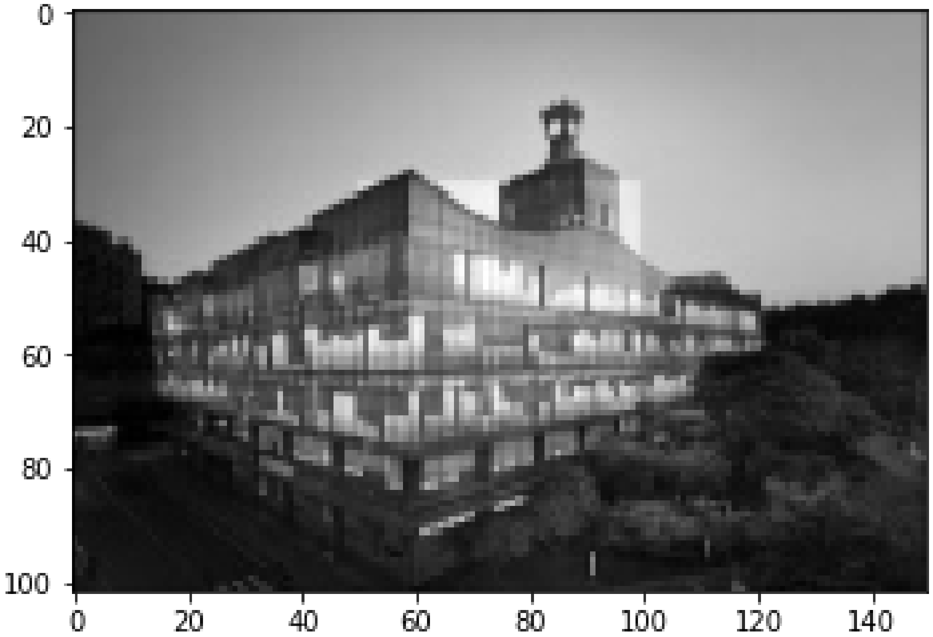
8 Defination of heatequation

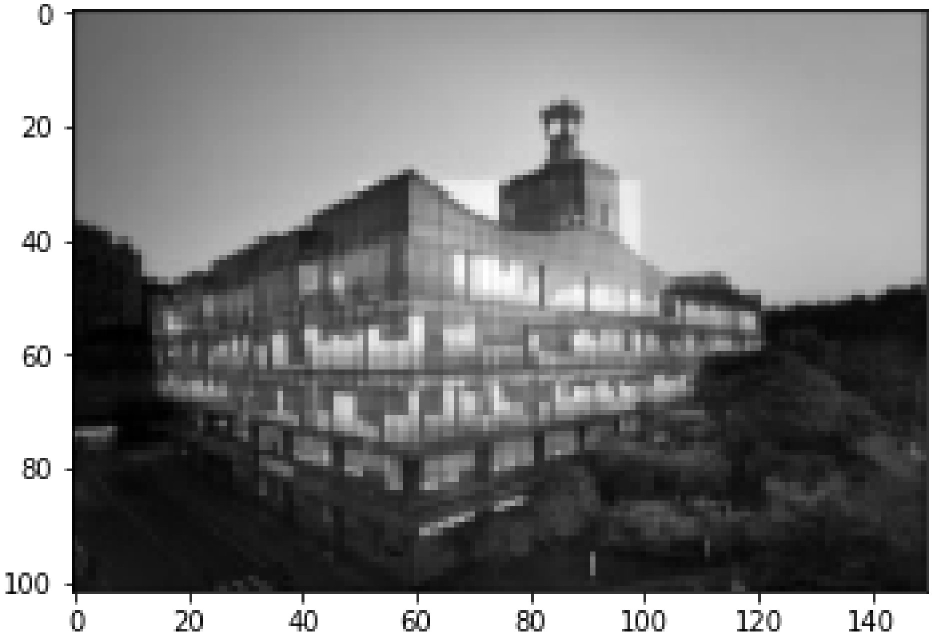
```
In [6]: def heatequation(im,t):
            ker =[[0,0.25 * t,0],[0.25 *t,0.25 *4*t,0.25*t],[0,0.25*t,0]]
            ker = torch.FloatTensor(ker).unsqueeze(0).unsqueeze(0)
            weight = nn.Parameter(data=ker, requires_grad=False)
            im = torch.FloatTensor(im)
            out = im
            for i in range(9):
                out = out + nn.functional.conv2d(out,ker,padding = 1)
                plt.imshow(out.reshape(102,150),cmap = 'gray')
                plt.show()
            return out
```

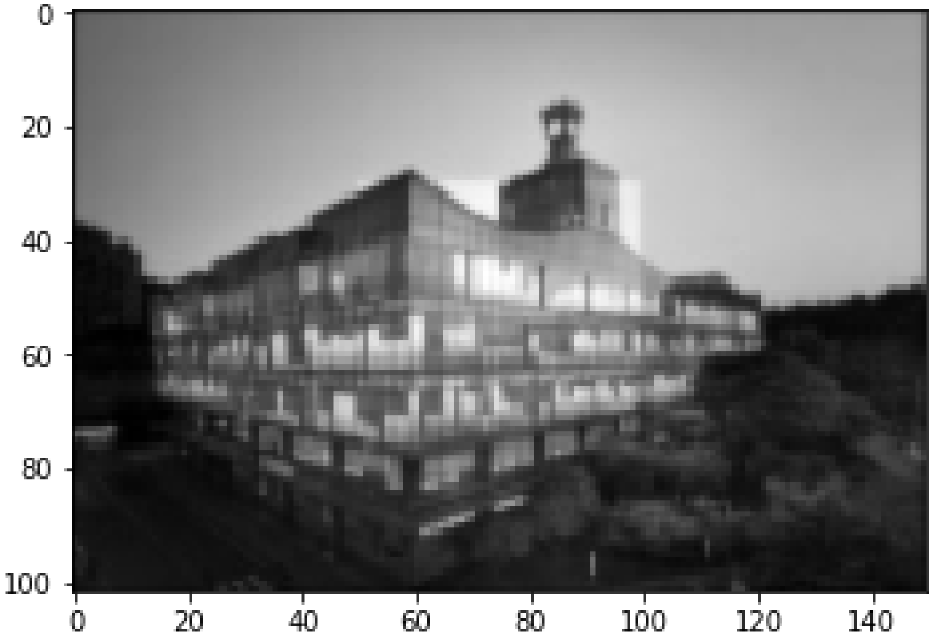
```
In [7]: he = heatequation(im_gray,0.4)
        #he = he.reshape(102,150)
```

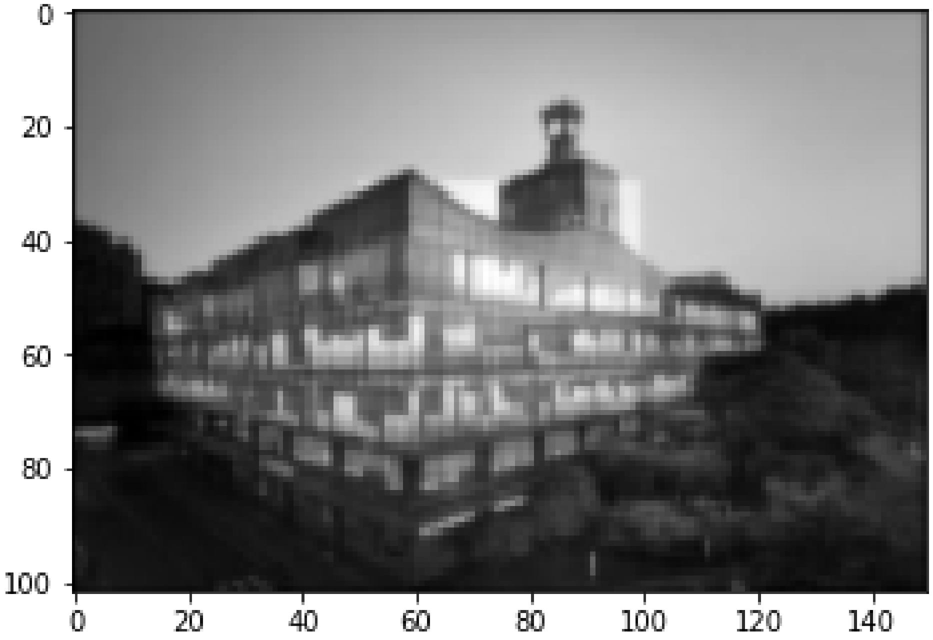
```
#he = im_gray.reshape(102,150) + he
#print(he)
```

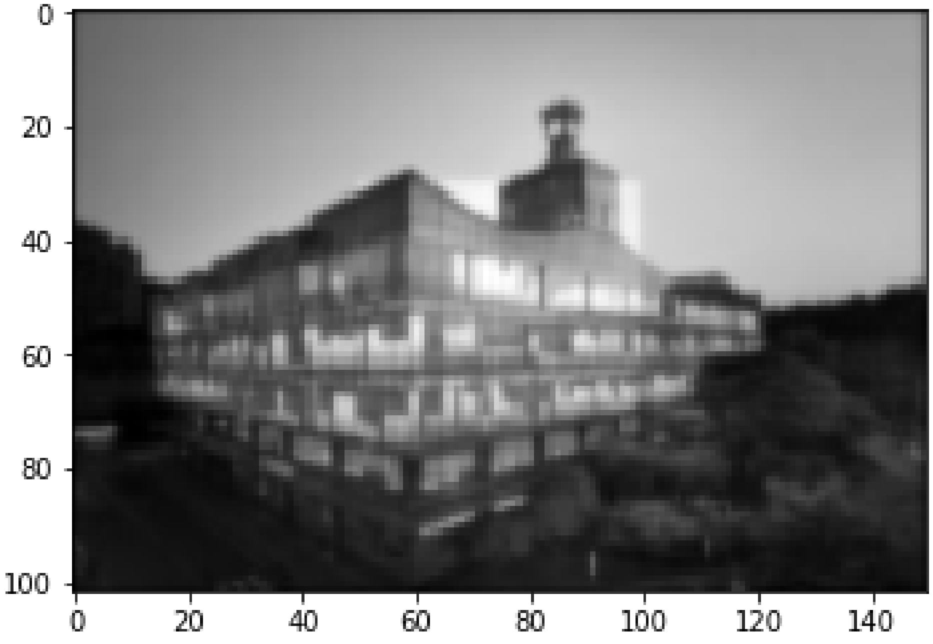


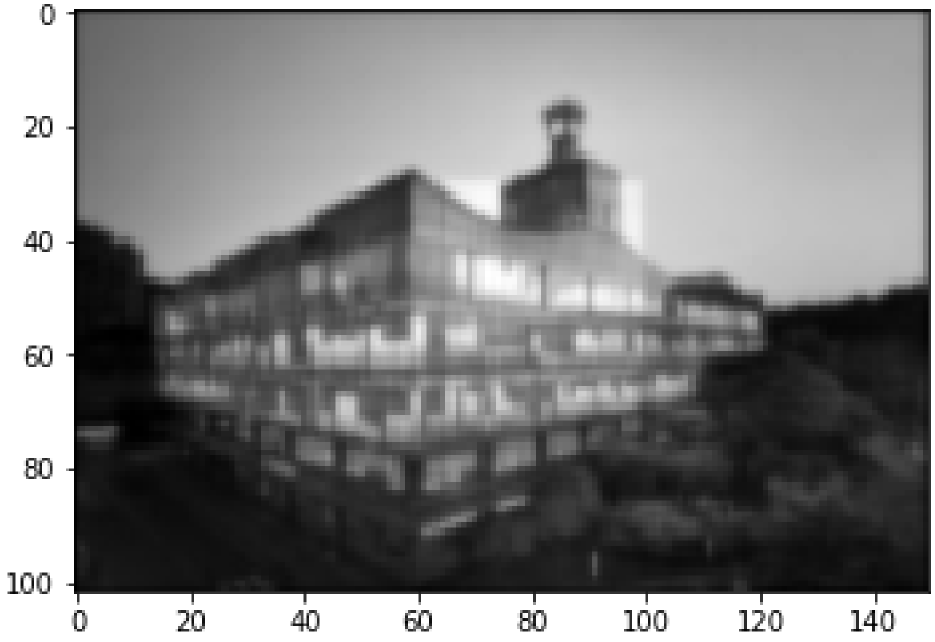



0
20
40
60
80
100
0
20
40
60
80
100
120
140

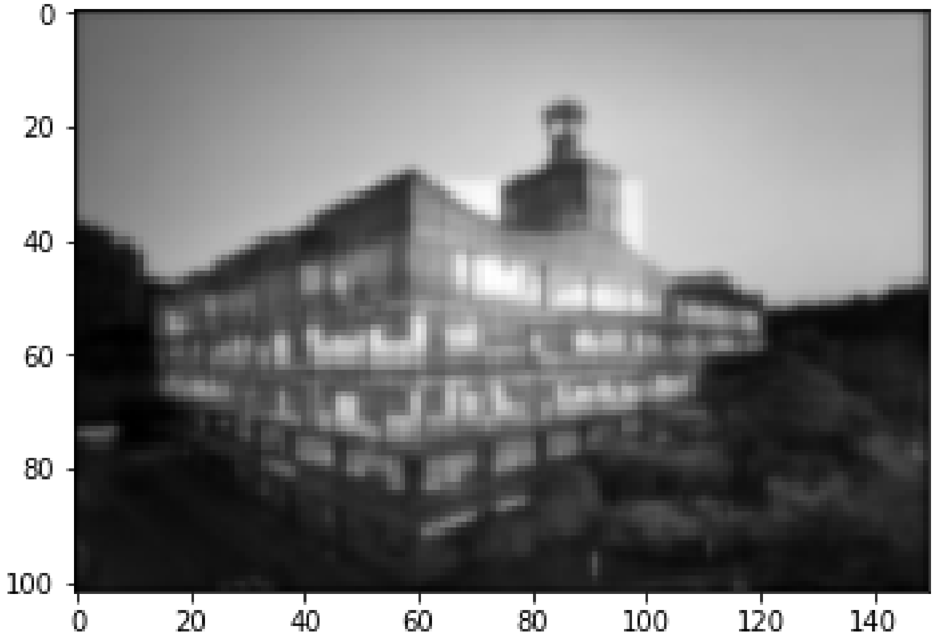

0
20
40
60
80
100
0
20
40
60
80
100
120
140


0
20
40
60
80
100
0
20
40
60
80
100
120
140

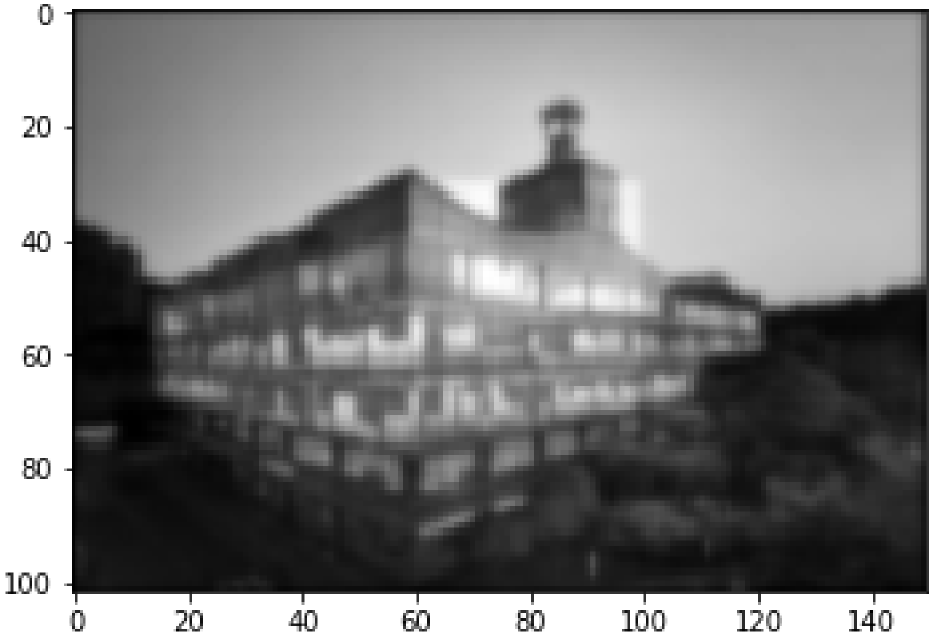

0
20
40
60
80
100
0
20
40
60
80
100
120
140



0
20
40
60
80
100
0
20
40
60
80
100
120
140



0
20
40
60
80
100
0
20
40
60
80
100
120
140



```
In [ ]:
```