R00182510 Machine Vision Assignment 1

Both Task 1 and Task 2 were implemented in a single file R00182510_MV_A1_Code.py

To execute Task_1, in the main() function please assign the value ‘Task_1’ to the execute variable.
To execute Task_2, in the main() function please assign the value ‘Task_2’ to the execute variable.

Task 1

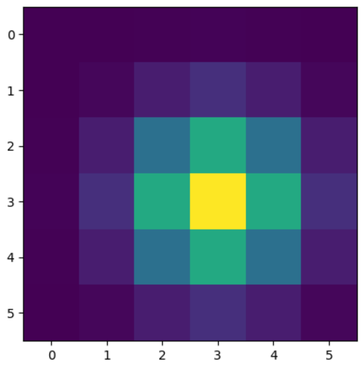
Implementation of scale and rotation invariant point feature extraction algorithm inspired by SIFT to identify a set of interest points in an image together with their respective scales and rotations.

All the output images from Task-1 will be saved to a folder as given in the ‘picPath’.

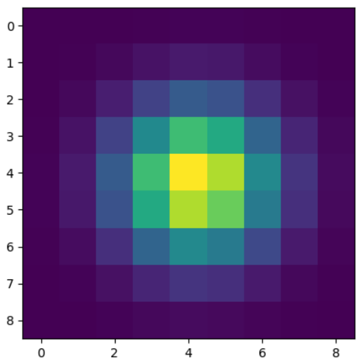
```
# The output path to save all the figures/output images generated in Task 1
picFolderName = 'R00182510_Task1_Output_Images'
picPath = os.path.join(os.getcwd(),picFolderName)
if not os.path.exists(picPath):
    os.makedirs(picPath)
```

Subtask B – Output Images

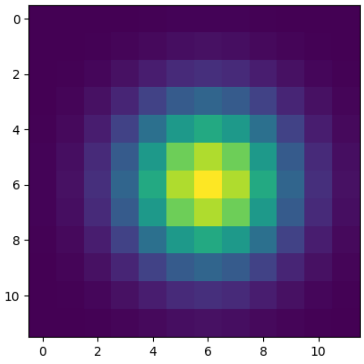
Gaussian Kernel 1



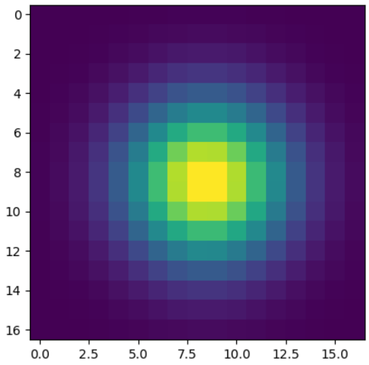
Gaussian Kernel 2



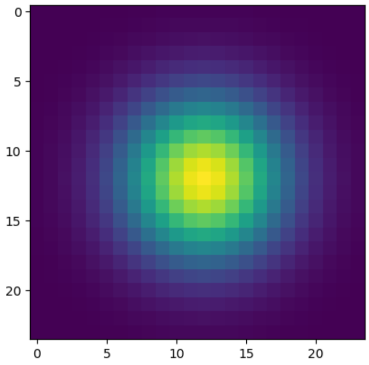
Gaussian Kernel 3



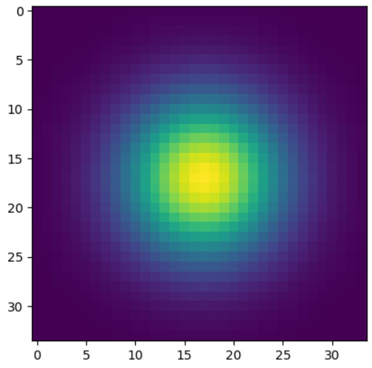
Gaussian Kernel 4



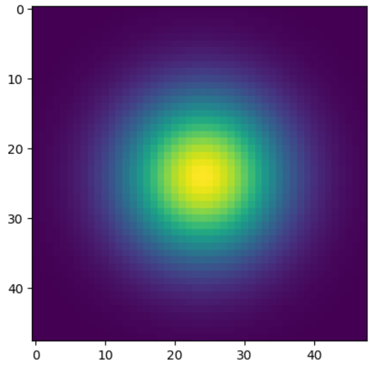
Gaussian Kernel 5



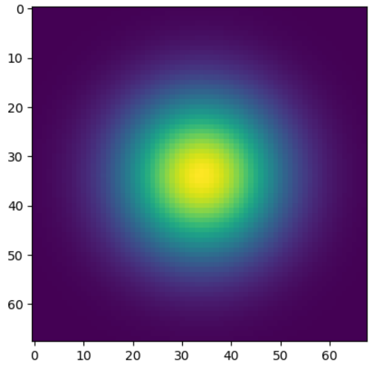
Gaussian Kernel 6



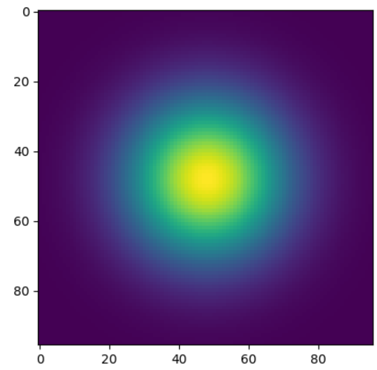
Gaussian Kernel 7



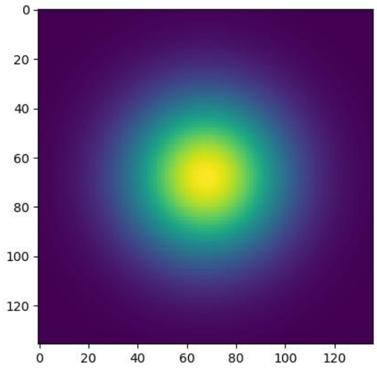
Gaussian Kernel 8



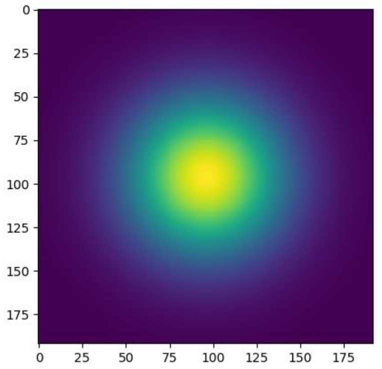
Gaussian Kernel 9



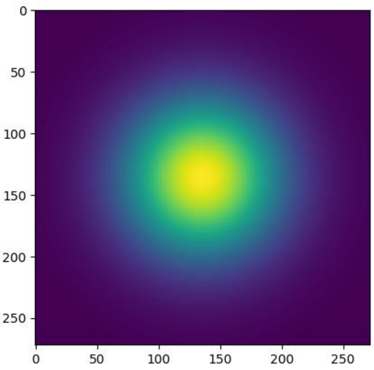
Gaussian Kernel 10



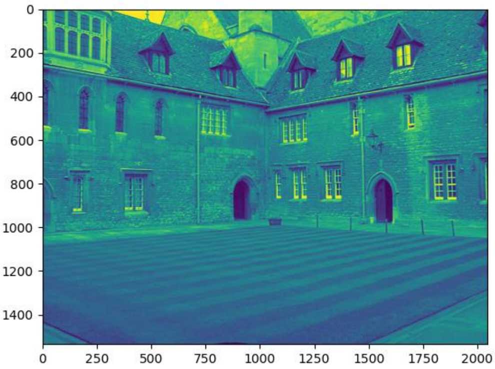
Gaussian Kernel 11



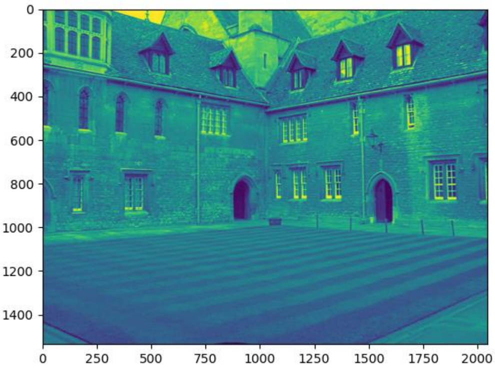
Gaussian Kernel 12



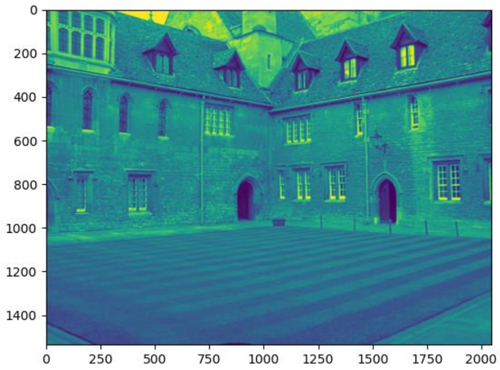
Gaussian Image 1



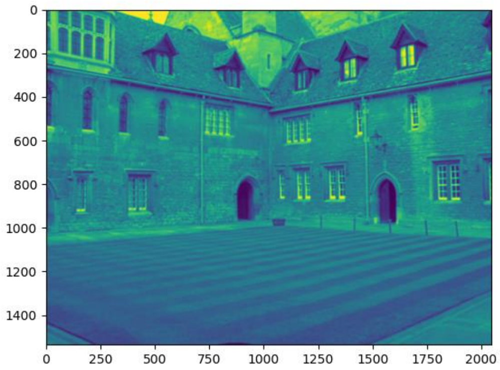
Gaussian Image 2



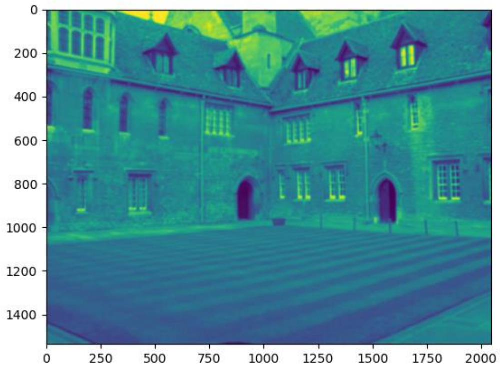
Gaussian Image 3



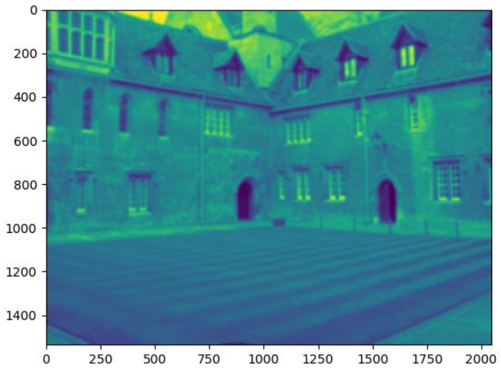
Gaussian Image 4



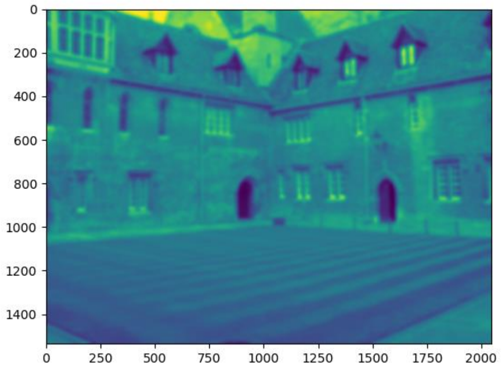
Gaussian Image 5



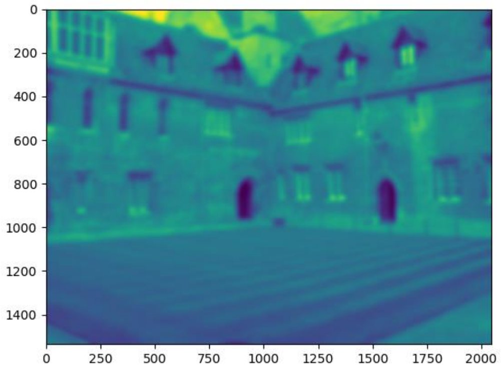
Gaussian Image 6



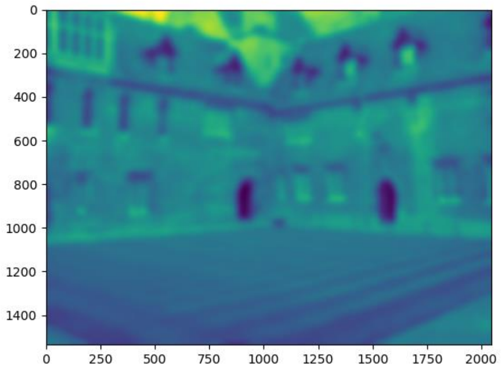
Gaussian Image 7



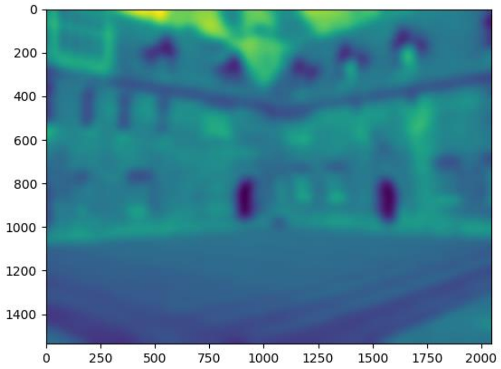
Gaussian Image 8



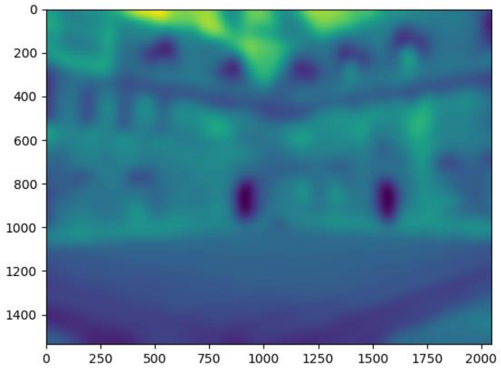
Gaussian Image 9



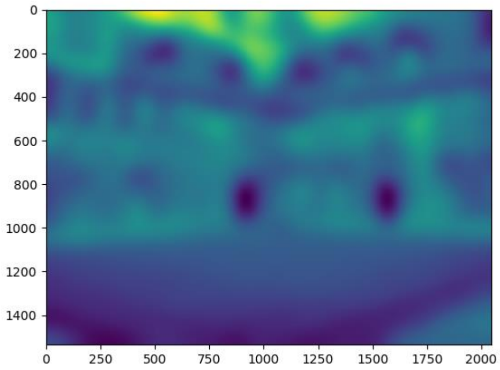
Gaussian Image 10



Gaussian Image 11

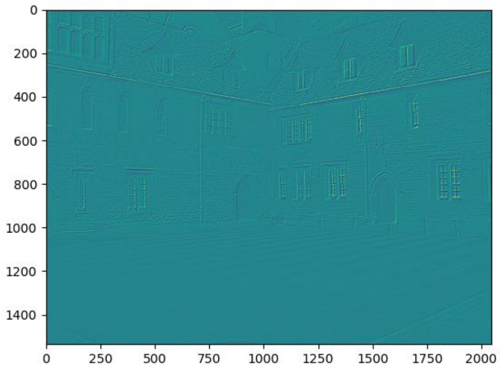


Gaussian Image 12

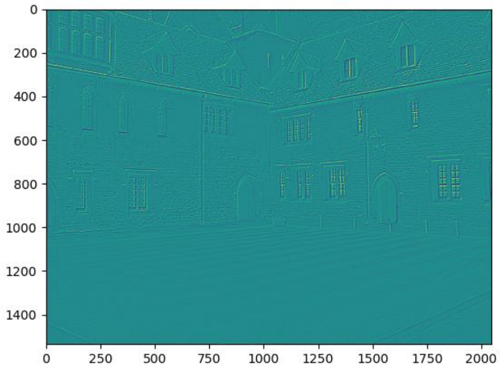


Subtask C – Output Images

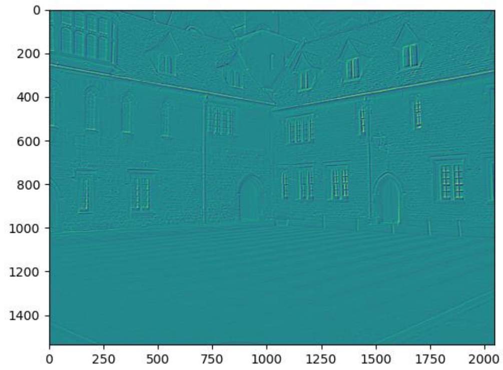
DOG Image 1



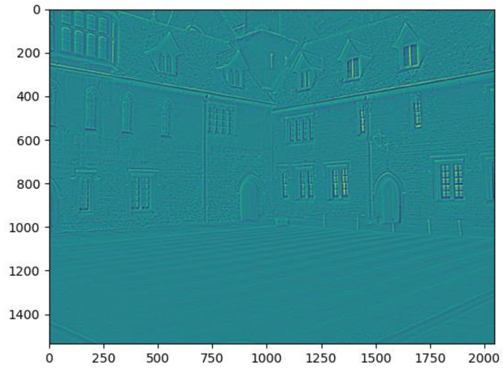
DOG image 2



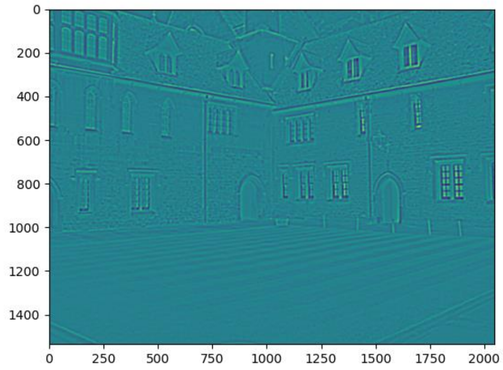
DOG Image 3



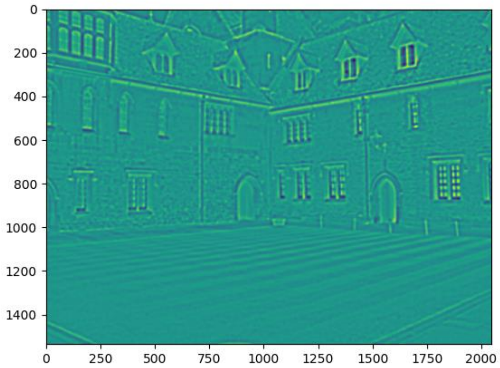
DOG Image 4



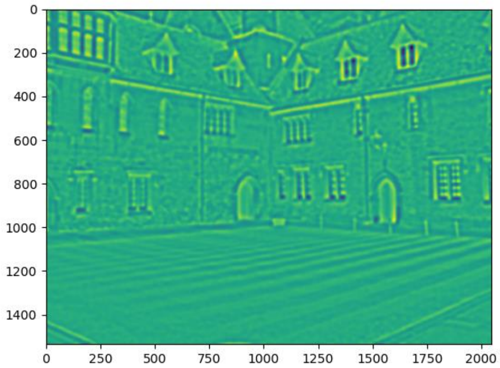
DOG Image 5



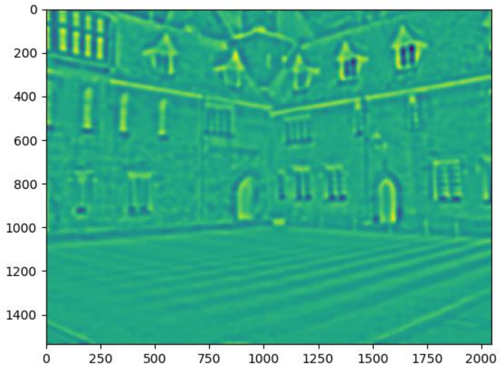
DOG Image 6



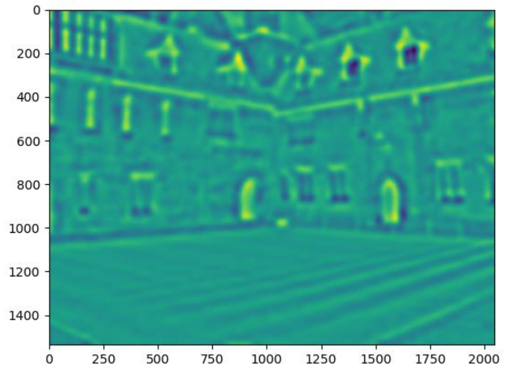
DOG Image 7



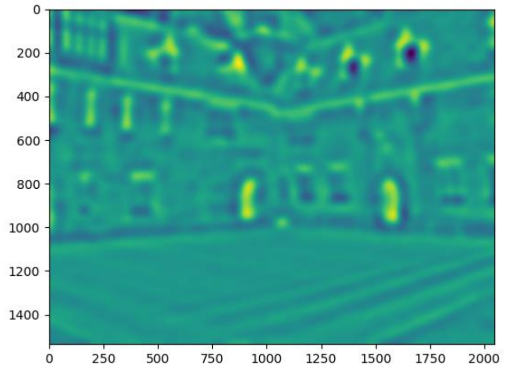
DOG Image 8



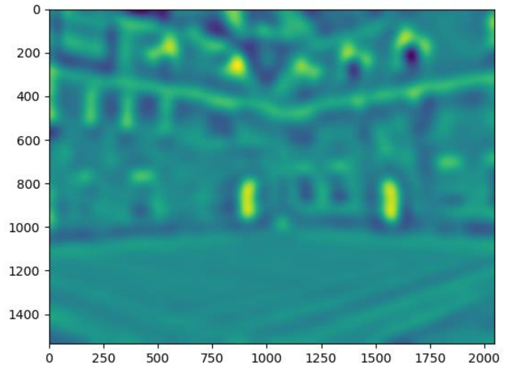
DOG Image 9



DOG Image 10



DOG Image 11

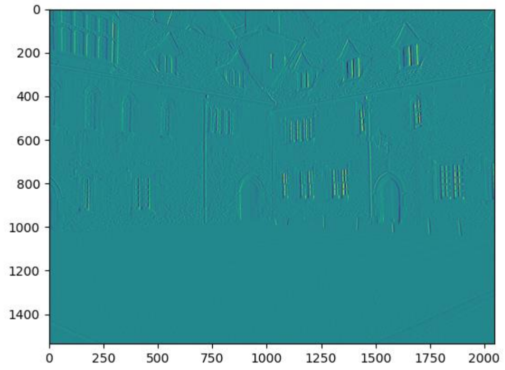


Subtask D

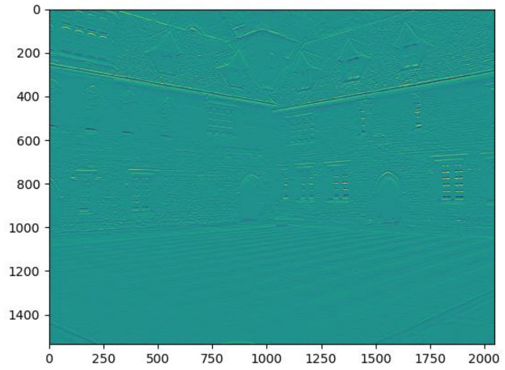
No. of Key-points generated: 4449

Subtask E Output Images

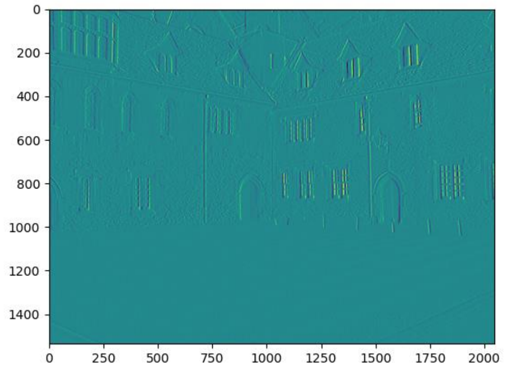
Derivative-X Image 1



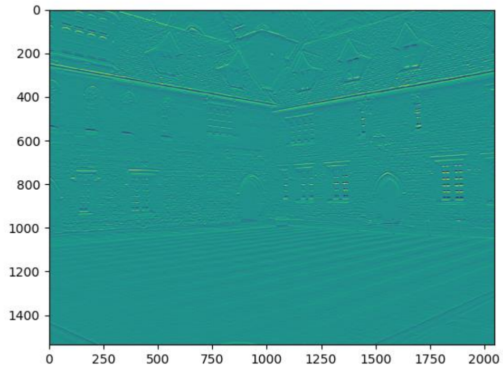
Derivative-Y Image 1



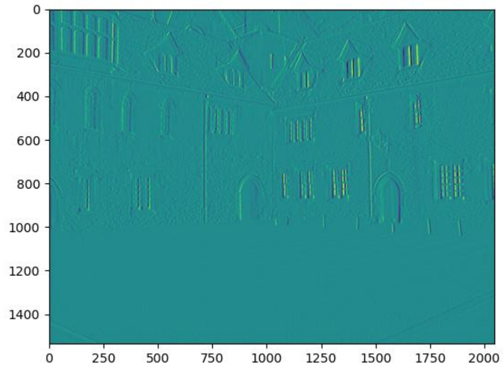
Derivative-X Image 2



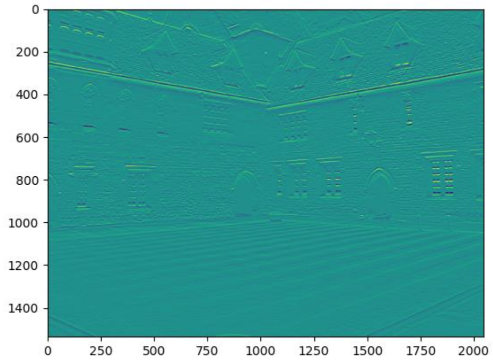
Derivative-Y Image 2



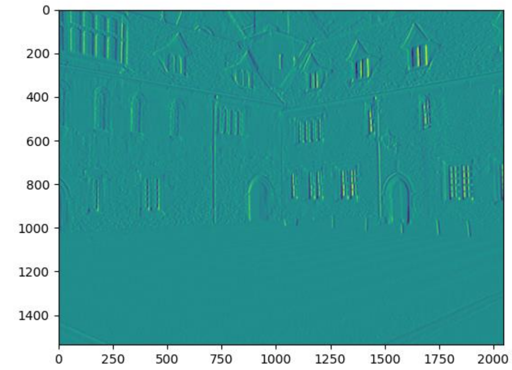
Derivative-X Image 3



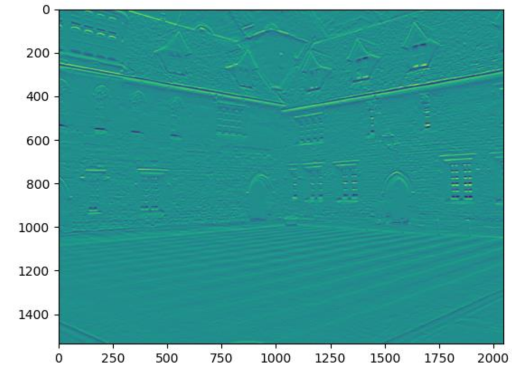
Derivative-Y Image 3



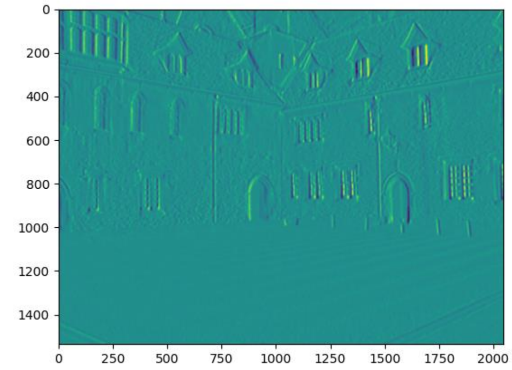
Derivative-X Image 4



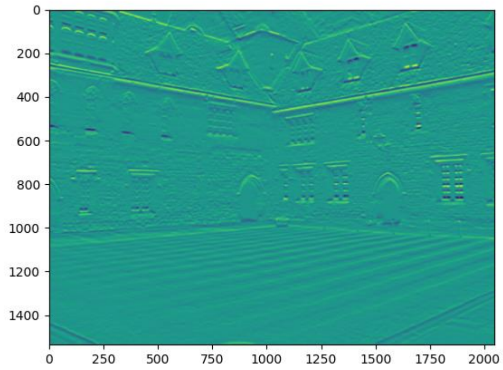
Derivative-Y Image 4



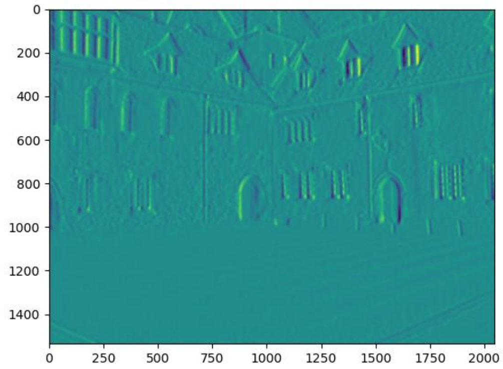
Derivative-X Image 5



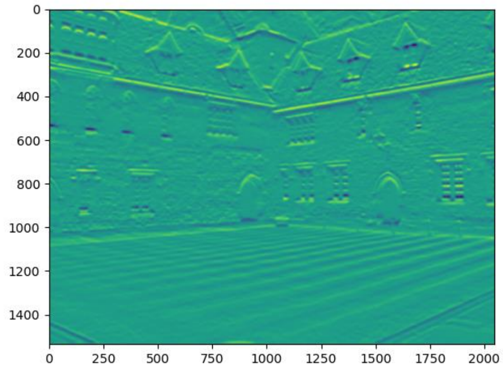
Derivative-Y Image 5



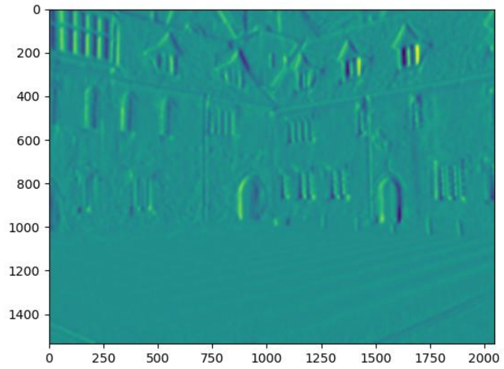
Derivative-X Image 6



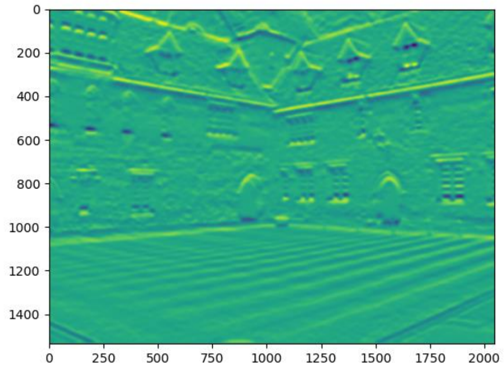
Derivative-Y Image 6



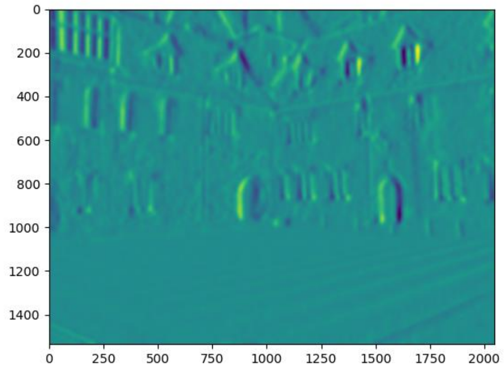
Derivative-X Image 7



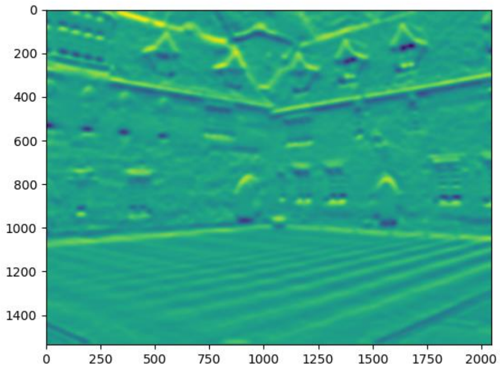
Derivative-Y Image 7



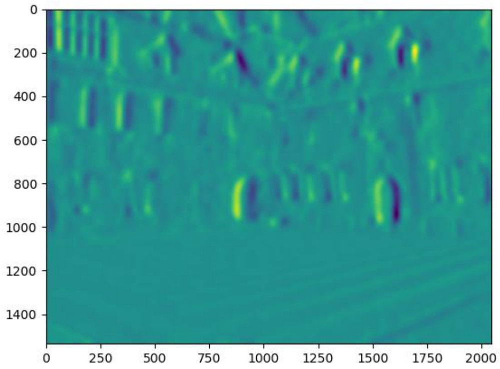
Derivative-X Image 8



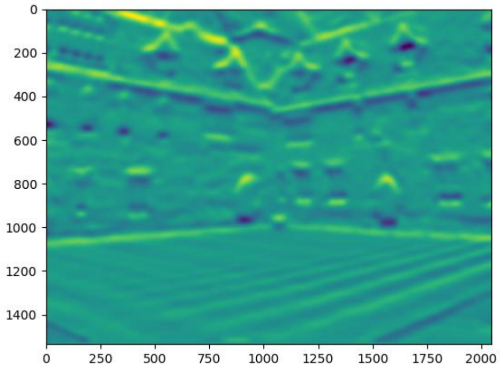
Derivative-Y Image 8



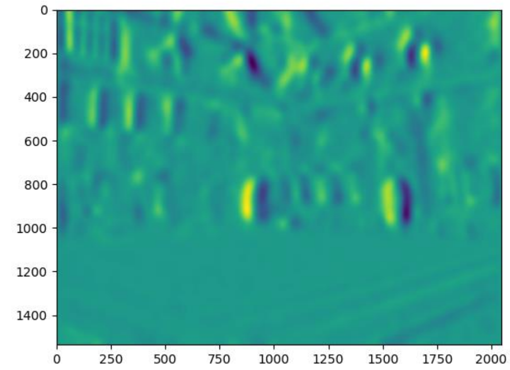
Derivative-X Image 9



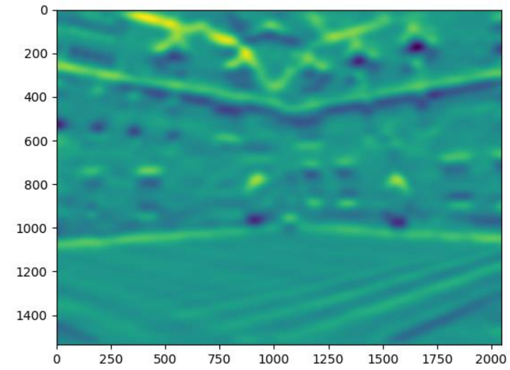
Derivative-Y Image 9



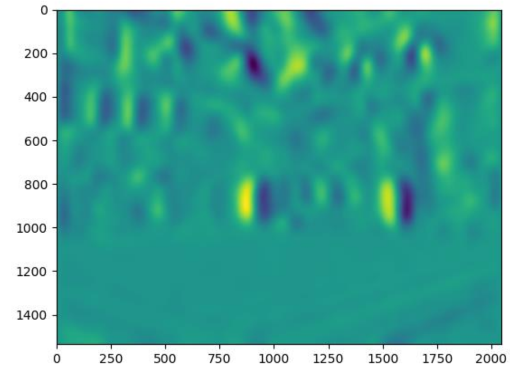
Derivative-X Image 10



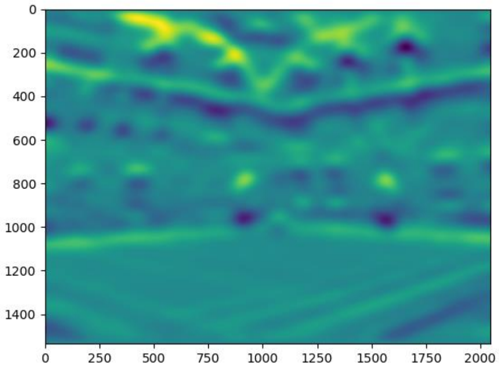
Derivative-Y Image 10



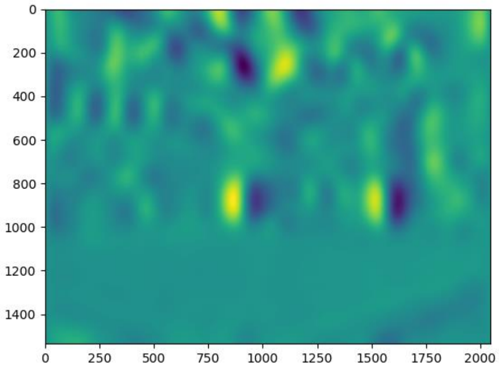
Derivative-X Image 11



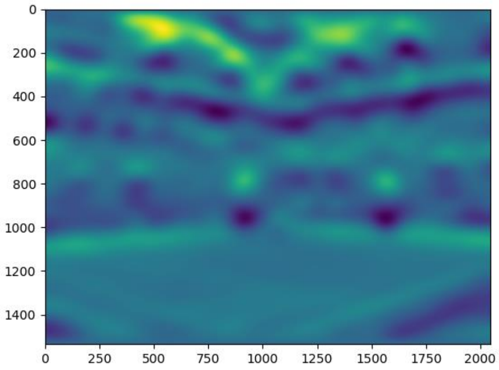
Derivative-Y Image 11



Derivative-X Image 12

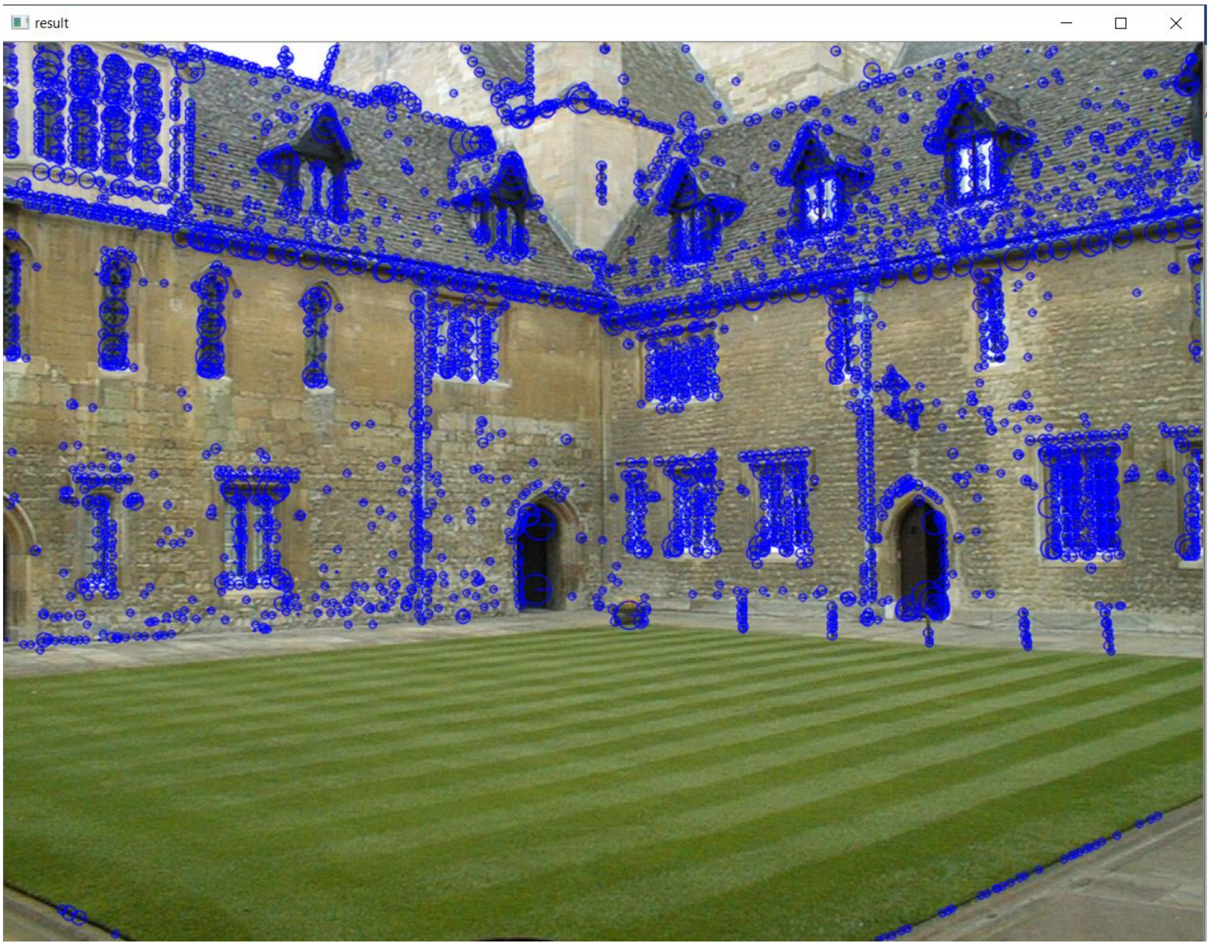


Derivative-Y Image 12



Subtask G

Final image with key-points

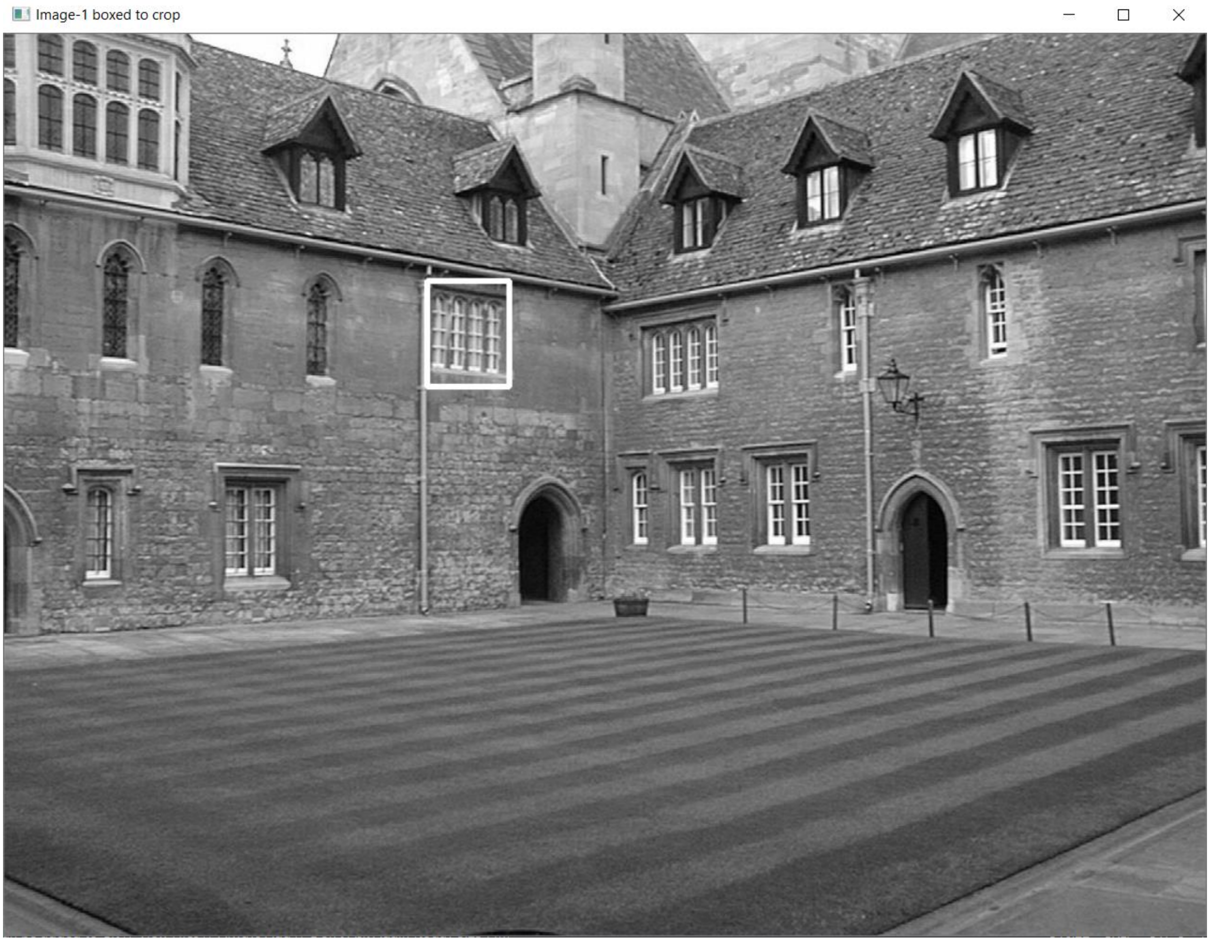


Task 2

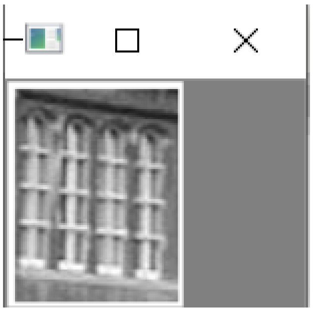
Implementation of a correlation-based area matching algorithm to find a patch extracted from one image in a second image

Subtask B Output Images

Image-1 with a rectangle drawn around the area of interest

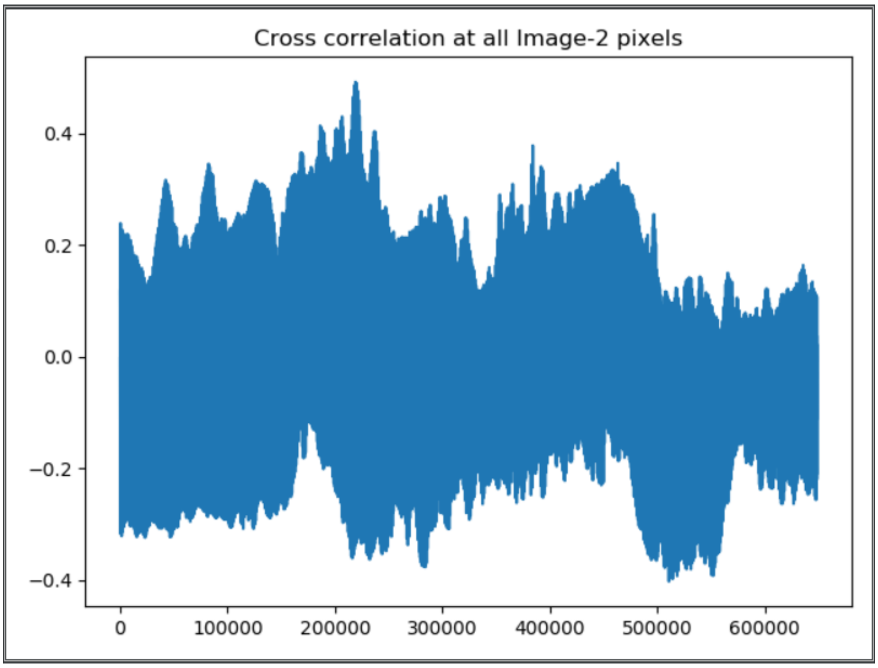


Cropped patch from Image-1



Subtask C Output Images

Cross correlation between all patches of Image-2 and the cropped patch from Image-1



Matched area in Image-2 indicated by a rectangle

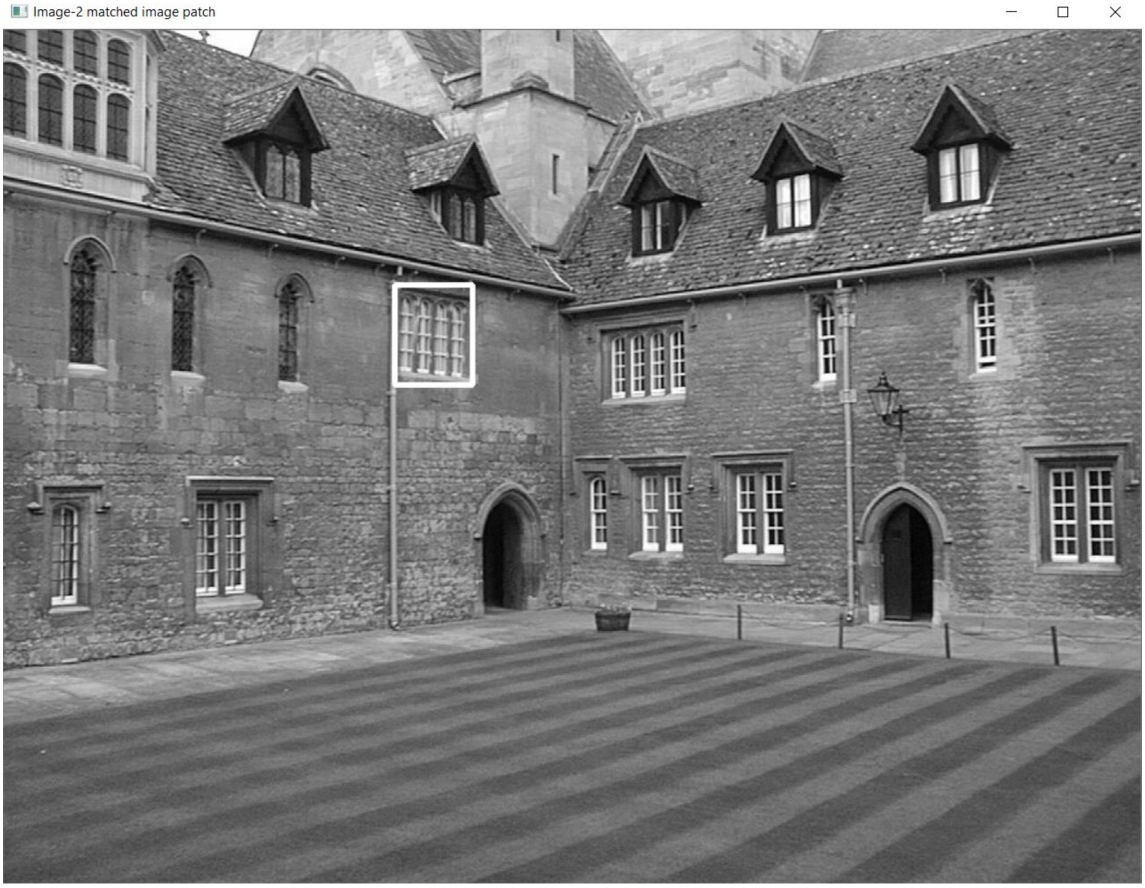


Image-2 matched image patch