# CS512 Assignment 2: Report

Kajol Tanesh Shah
Department of Computer Science
Illinois Institute of Technology
Feb 17, 2022

## Abstract

In this assignment, we have performed various transformations on images like corner detection, smoothing, etc. based on the special keys pressed on the keyboard. To do these operations, we have used the OpenCV library in python and also used Cython to make our program's execution faster.

## 1. Problem Statement

In this assignment, our aim was to perform various transformations on images like converting to grayscale, smoothing, upscaling, downscaling, finding x-derivative, image gradients, detecting corners, etc. These image transformations are helpful as we can many times detect some features in the images after transformation which we are not able to identify in the original image. So, we have performed these transformations on images of different sizes and based on the key entered by the user on the keyboard, the operations were performed. For parts in code where double loops are used, to make implementation faster, we have also used Cython.

## 2. Proposed solution

Using the OpenCV library and other supporting libraries like numpy, we have implemented the program to do transformations on the images. Below are some of the functions that we performed and their working.

A)Smoothing

We implemented smoothing using two methods:-
1)Using the OpenCV function filter2D() in which we provided the grayscale image as input and we got the smoothed image as output. Also, we used the trackbar to control the amount of smoothing
2)Without using OpenCV functions, we implemented our own convolution function with the help of loops. We first created a kernel array for smoothing with all ones in it. Then, we applied this filter to the image and also did zero padding and we got a smoothed image as a result. As there were multiple for loops used in the logic, we used Cython to make the implementation faster. Also, we used the trackbar to control the amount of smoothing

B) Downsampling
Here, we downsampled the image by a factor of 2 using the resize function and also did smoothing. So, our image became half of its original size

C) Upsampling
Here, we upsampled the image by a factor of 2 using the resize function and also did smoothing. So, our image became 2 times its original size

D)Computing x-derivative
Here, using the Sobel function from OpenCV, we found the x-derivative of the image and also normalized it in the range [0,255]

E)Magnitude of image gradient
Here, we computed the magnitude of the image gradient by computing the x-derivative and the y-derivative of the image using Sobel() and normalizing it in the range [0,255]

F)Plotting N pixels of image gradient vectors using line segments
First, we calculated the x and y derivatives using Sobel() and then calculated the angle and used arrowedLine() function to plot the line segments. We also used trackbar to control the number of pixels

G)Corner detection
We implemented this in two ways:-
1)Using OpenCV's cornerHarris function, we detected the corners by specifying the threshold and identified the corners using red dots. We also used trackbar to control the number of corners shown using k parameter
2)Without using corner detection, we implemented corner detection by using the goodFeaturestotrack() function and a for loop to find the corners. We also used trackbar to control the number of corners shown

H)Some other functions which were implemented were reloading the image, saving the image and displaying the description of the program and the keys. We reloaded the image by reading the image again using imread()

## 3. Implementation details

Some of the problems and design issues which were faced are as mentioned below:

- While trying to implement the convolution function using cython for smoothing, the variables were giving errors in Cython due to the Numpy issue. The issue got resolved later after adding np.ndarray
- By using the trackbar, first the smoothing function was not able to accept the slider input and was giving an error for the image variable used as it was not declared as global. Was able to resolve it later
- For computing the magnitude of the image gradient, was not able to understand the logic and so, the magnitude value which we got was singular. After going through the formula, was able to code it properly
- For corner detection using cornerHarris, was not able to understand which parameter of the function controls the number of corner points. After going through the syntax and description, was able to understand that the k parameter controls the corner points and used the trackbar the get input k value

- Got Numpy error while trying to use threshold for corner detection in harris. As the grayscale image was of a single color channel, we were not able to identify corner points with red color as red had three channels. After using the cv2.merge() function, the issue got resolved

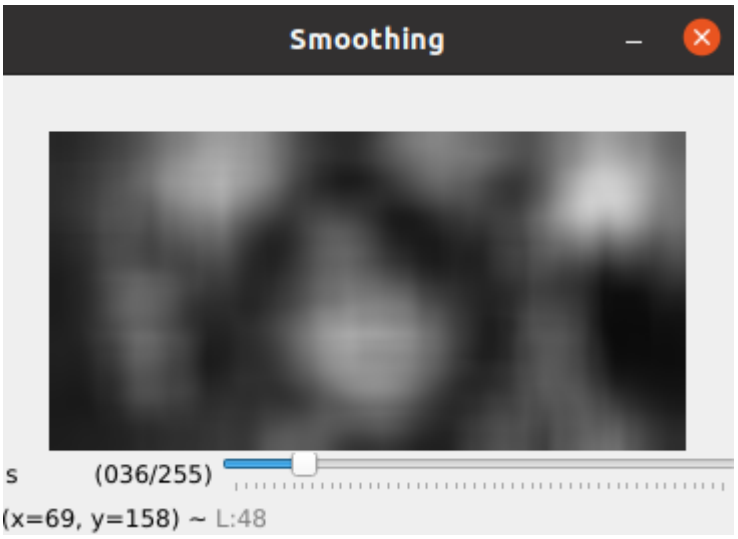
## 4. Results and Discussions

- Evaluation:

  Original image

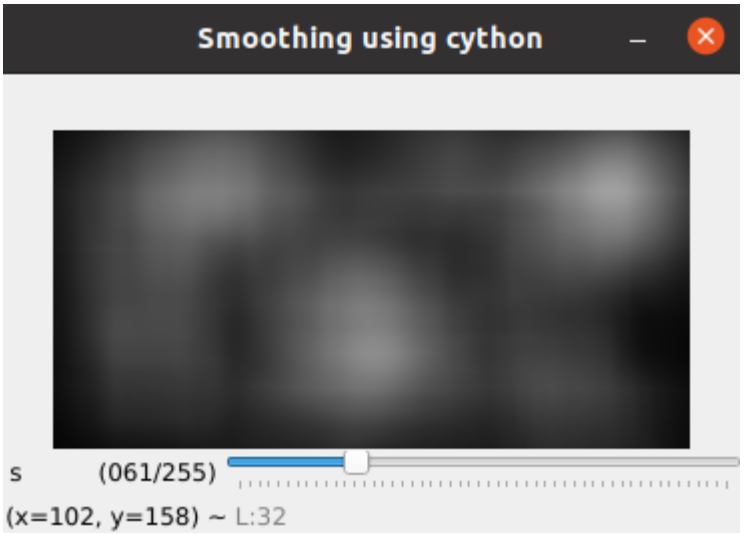
  After pressing s, smoothing operation using trackbar


  Here, we can see that the image is smoothed to some extent




Here, smoothing is very high and the features in the image are not visible

After pressing S, smoothing function(own implementation using Cython)

Implemented the above smoothing function using Cython

After pressing D, downsampling of the image after smoothing


```
Python 3.9.7 (default, Sep 16 2021, 13:09:58)
Type "copyright", "credits" or "license" for more information.

IPython 7.29.0 -- An enhanced Interactive Python.

In [1]: runfile('/home/kajol/Desktop/AS2.py', wdir='/home/kajol/Desktop')
(159, 318)
Press 'h' to see program description and keys, 'e' for exit

D
Size of image before downsampling is : (159, 318)
Size of image after downsampling is : (79, 159)
```

The size of the image got reduced

After pressing U, upsampling of image before smoothing

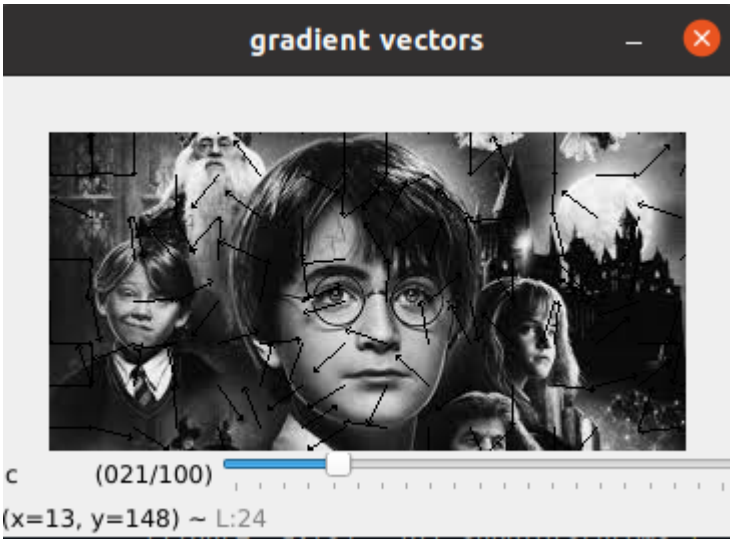
```
Python 3.9.7 (default, Sep 16 2021, 13:09:58)
Type "copyright", "credits" or "license" for more information.

IPython 7.29.0 -- An enhanced Interactive Python.

In [1]: runfile('/home/kajol/Desktop/AS2.py', wdir='/home/kajol/Desktop')
(159, 318)
Press 'h' to see program description and keys, 'e' for exit

U
Size of image before upsampling is : (159, 318)
Size of image after upsampling is : (318, 636)
```

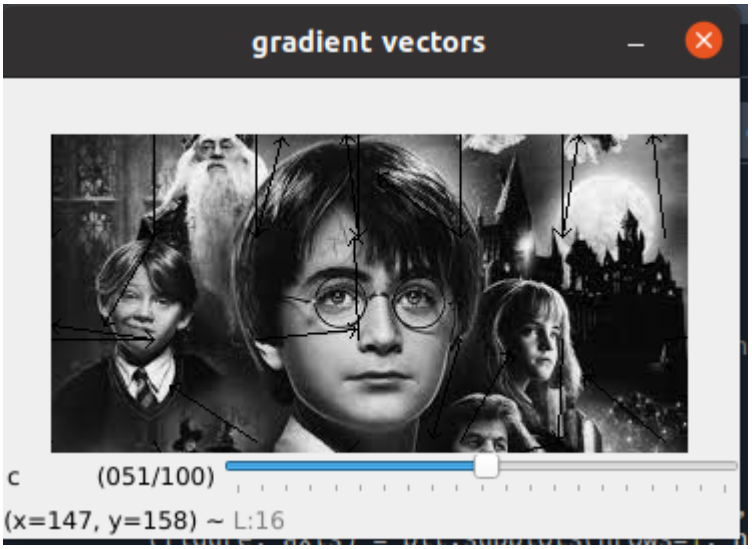
The size of the image  became 2 times the original

After pressing x, x-derivative of image

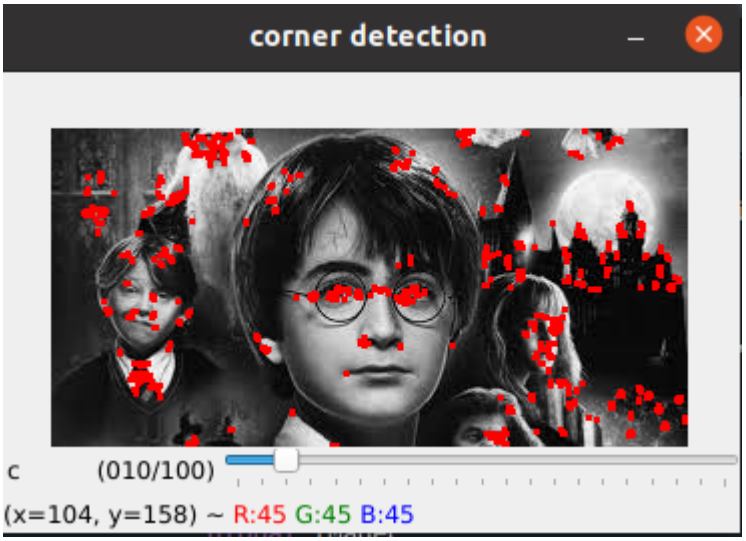

After pressing m, magnitude of the image gradient

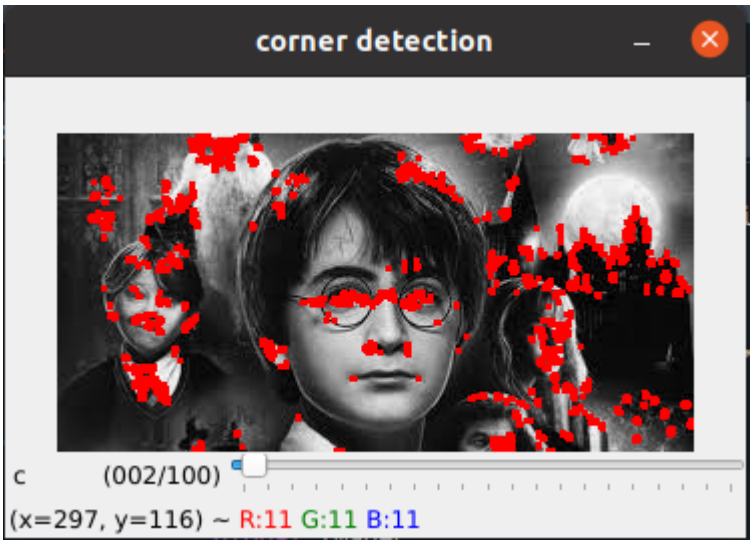
After pressing p, plotting of image gradient vectors



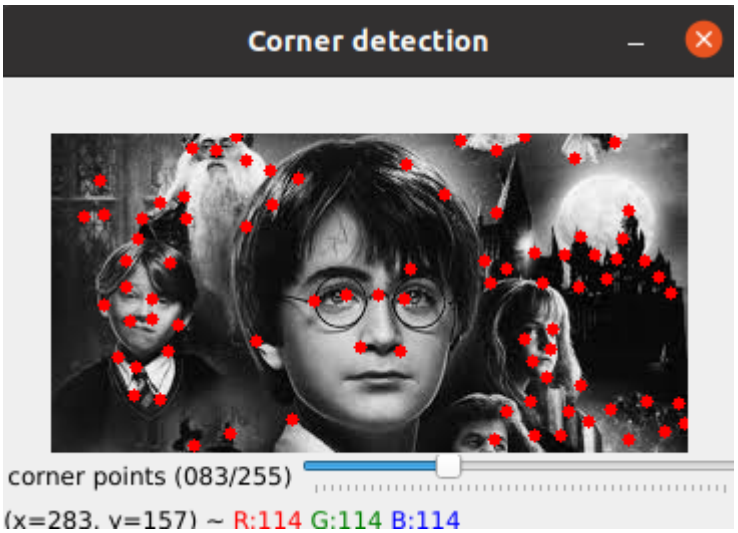

After pressing c, detecting corners using cornerHarris

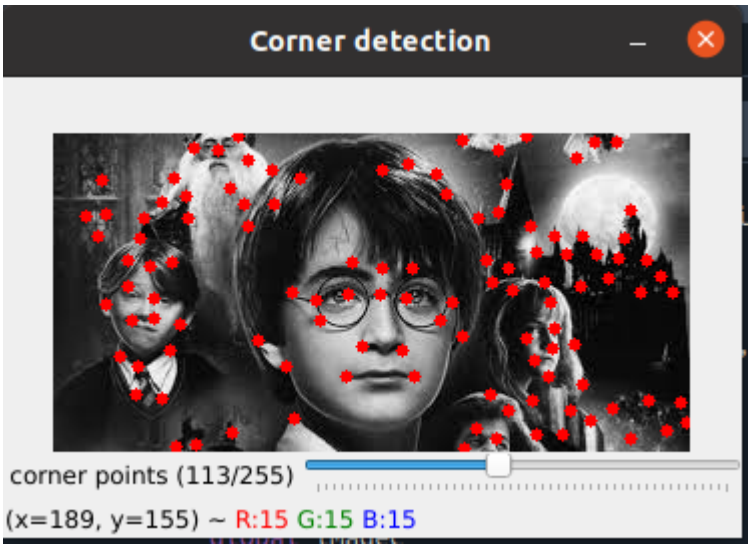



Here, the corners are highlighted in red and their count increases and decreases w.r.t input from the trackbar

After pressing C, execution of our own implemented corner detection function

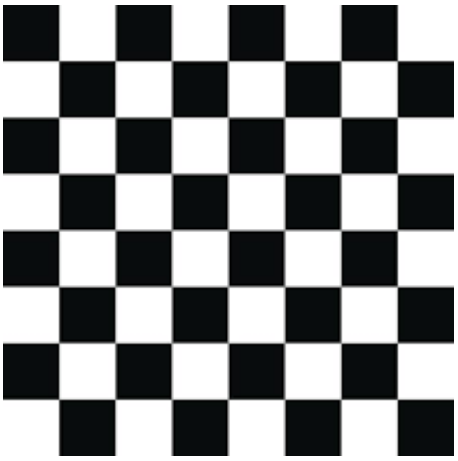


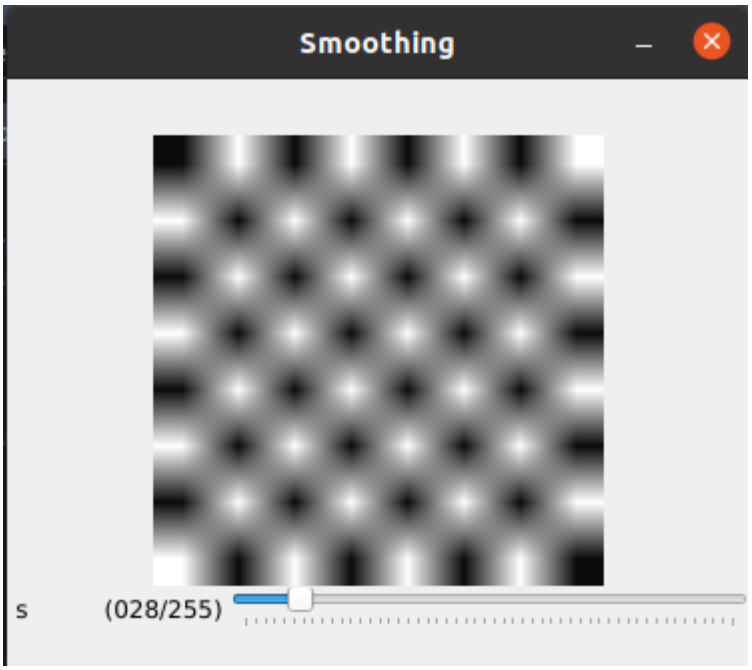

Here, the corners are highlighted in red and their count increases and descreases w.r.t input from the trackbar

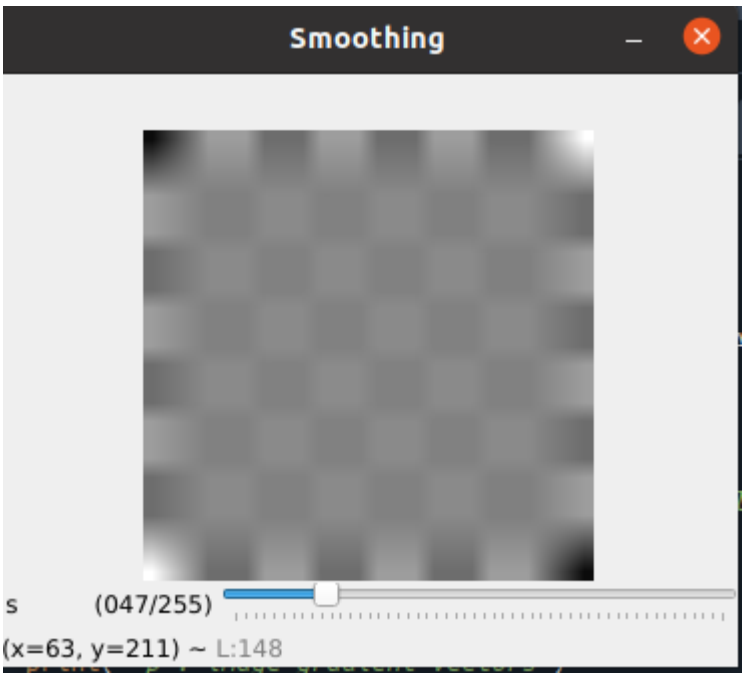
Case 2:

Original image

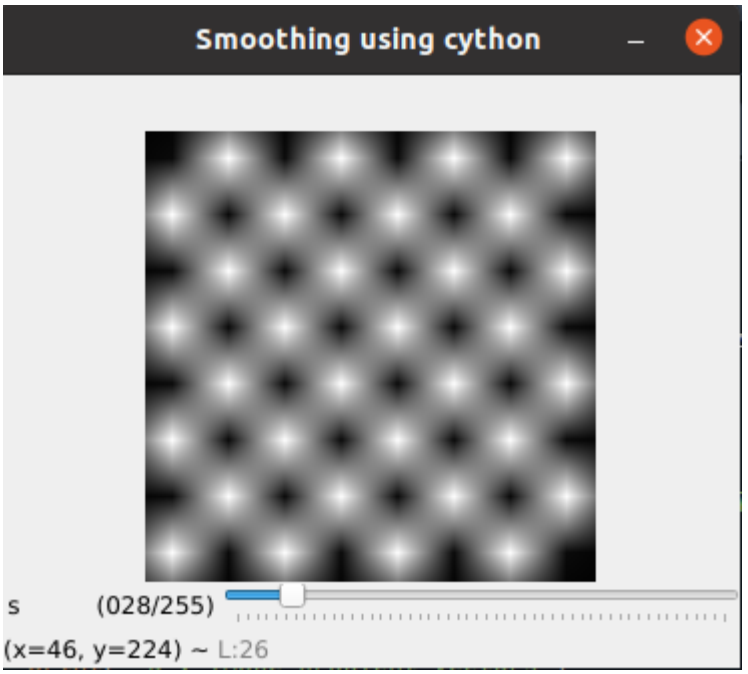


After pressing s

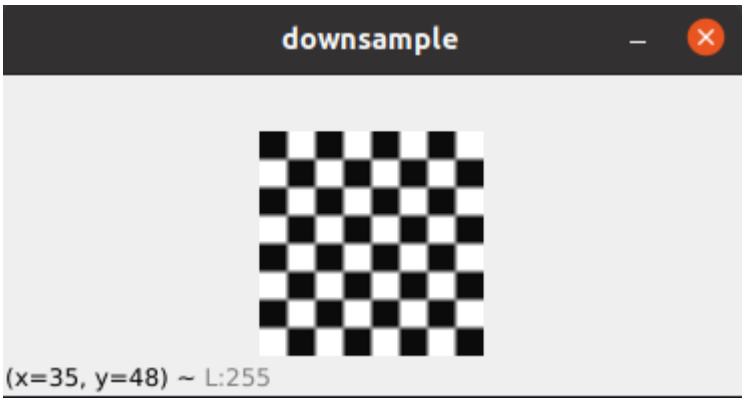




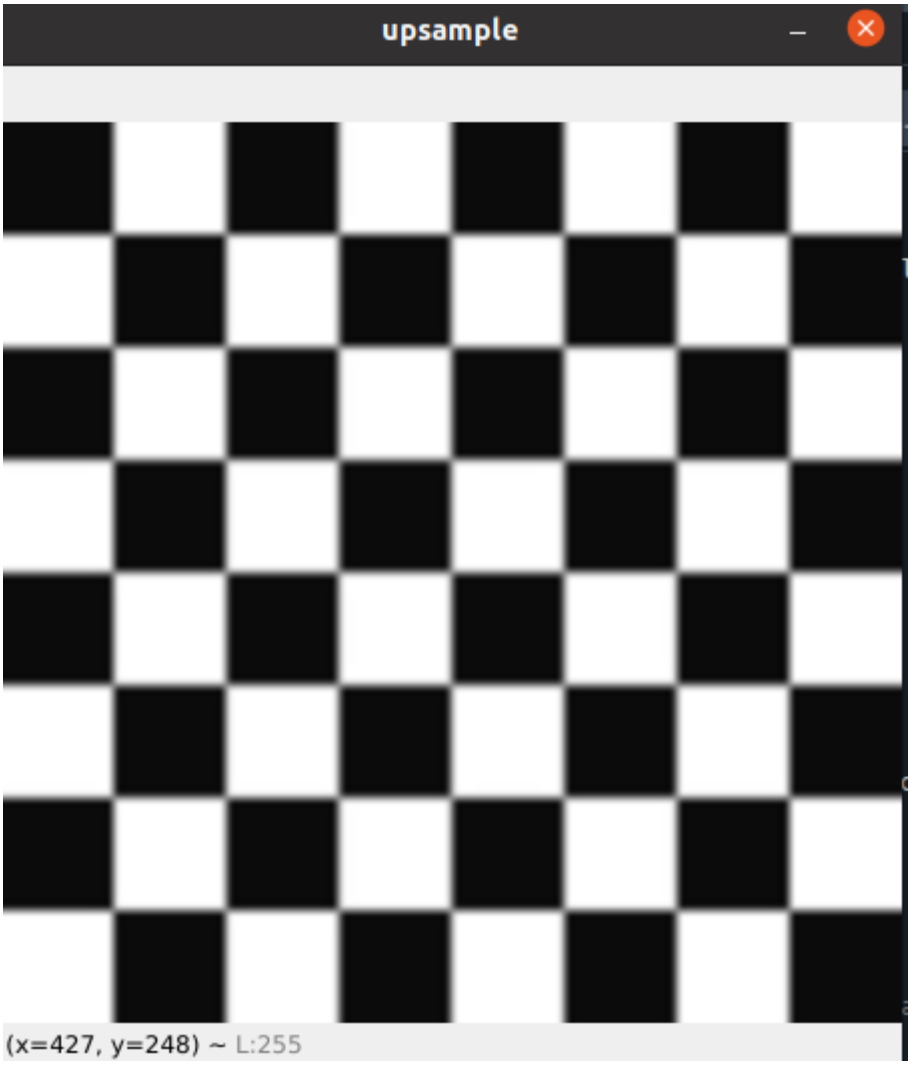

After pressing S



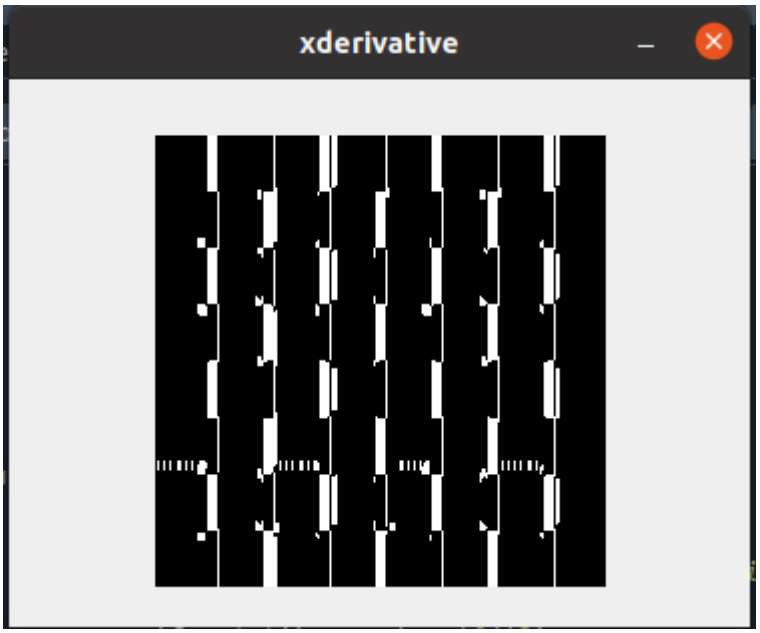
After pressing D



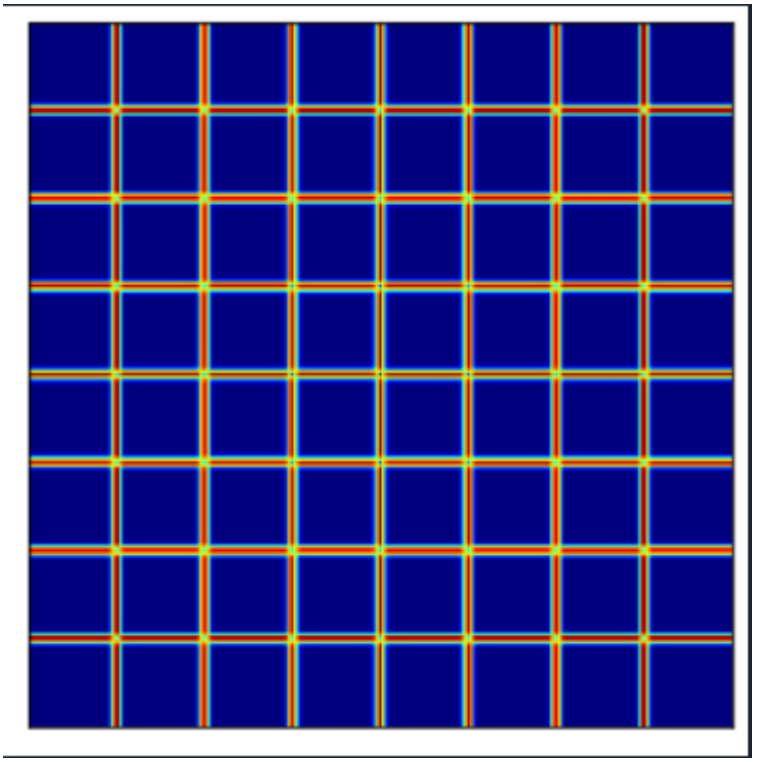

After pressing U



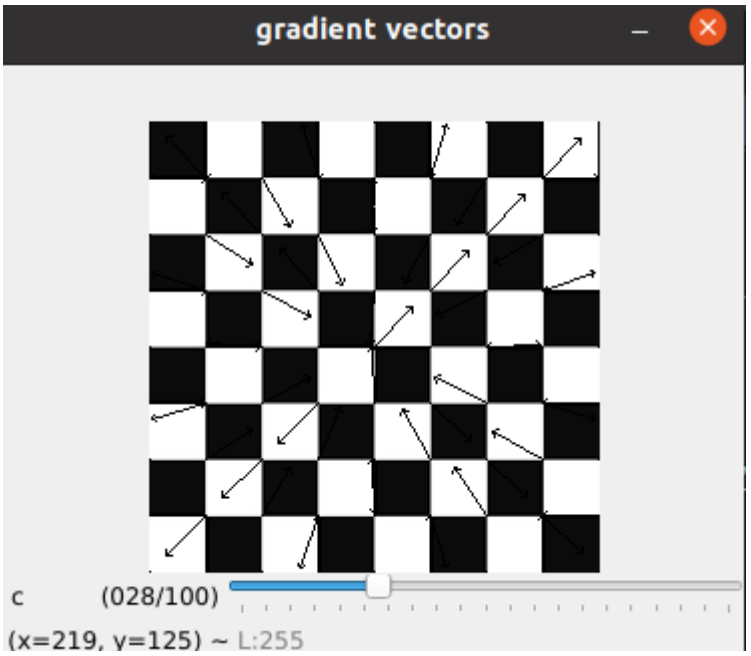
After pressing x



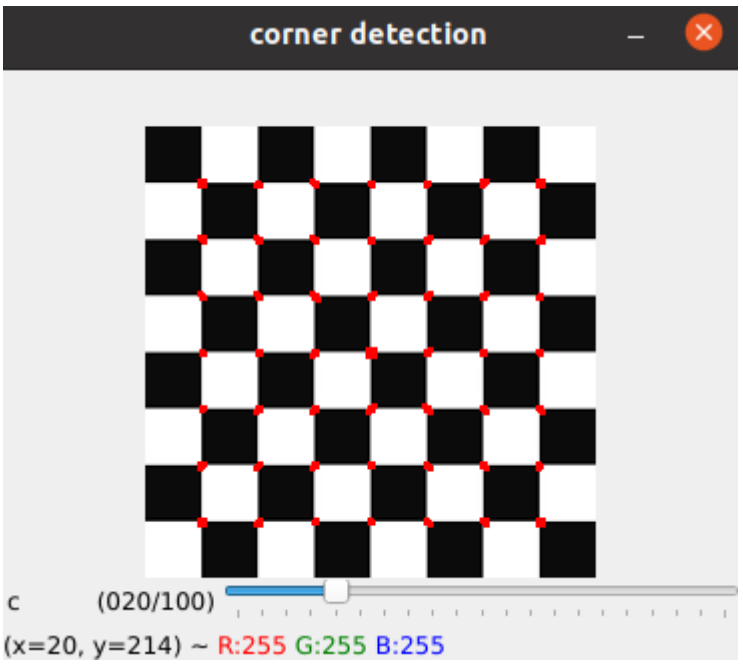

After pressing m

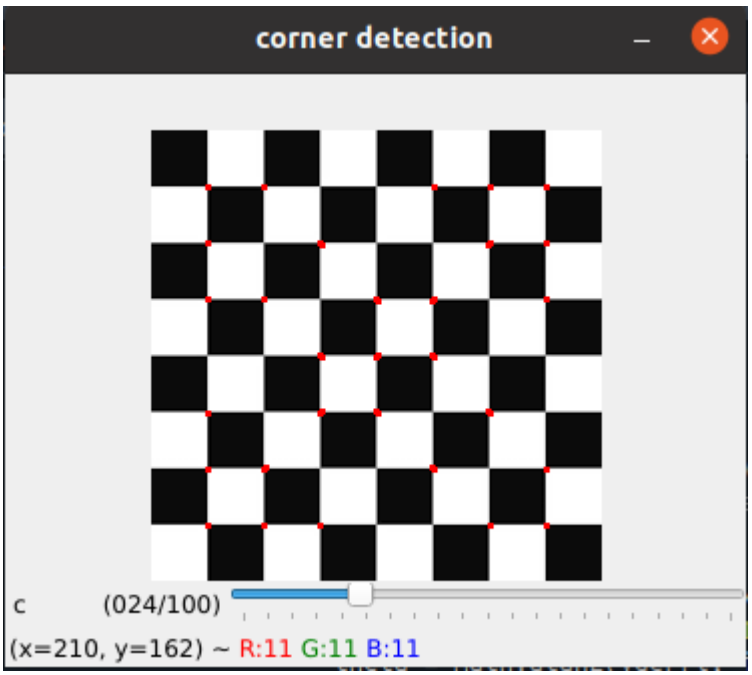


After pressing p

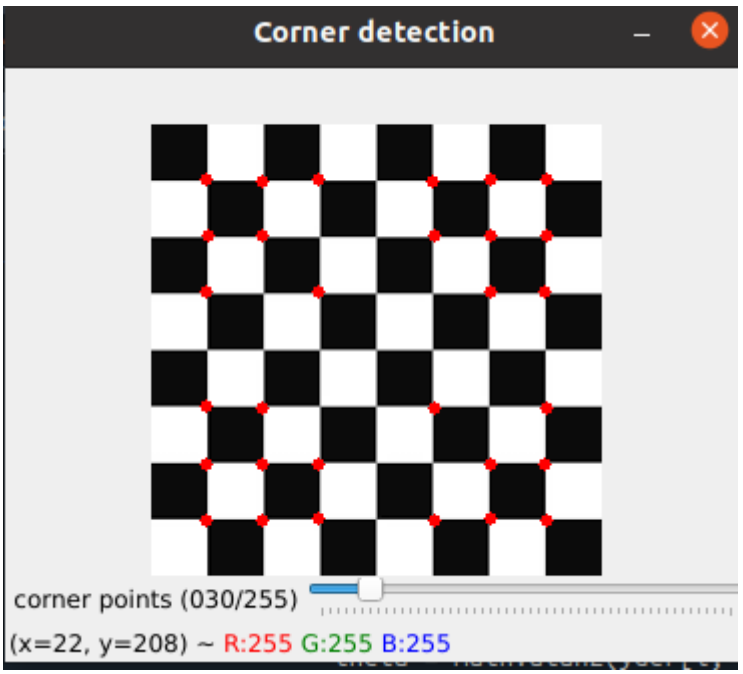


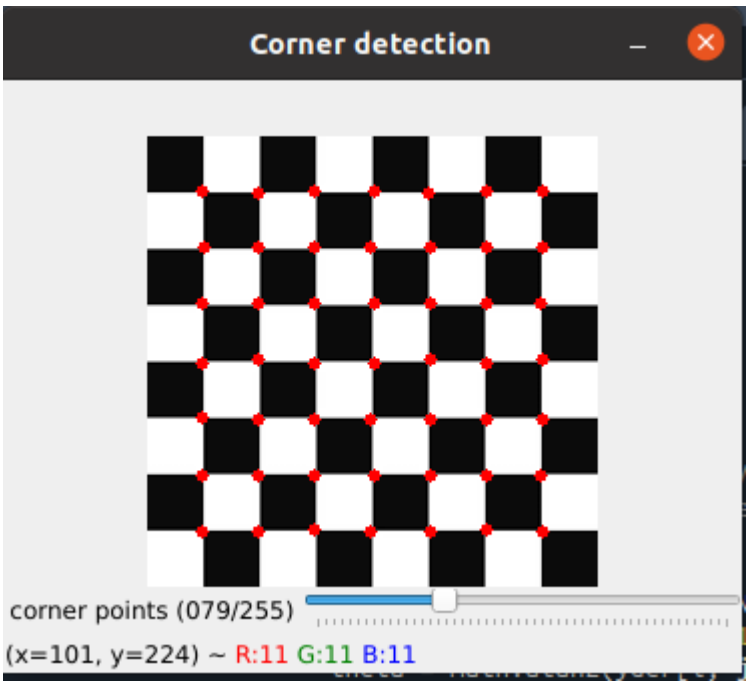

After pressing c






After pressing C






After pressing h

```
Python 3.9.7 (default, Sep 16 2021, 13:09:58)
Type "copyright", "credits" or "license" for more information.

IPython 7.29.0 -- An enhanced Interactive Python.

In [1]: runfile('/home/kajol/Desktop/AS2.py', wdir='/home/kajol/Desktop')
(225, 225)
Press 'h' to see program description and keys, 'e' for exit

h
's': smooth the image using opencv
'S': smooth the image using your own implementation in cython
'D': Downsample the image
'U': Upsample the image
'x': x-derivative of the image
'm': magnitude of the image
'p': image gradient vectors
'c': detection of corners using cornerharris
'C': detection of corners without using cornerharris
'h': display a short description of the program and the keys
```