SciComp with Py

CVIP

Basic Image Processing with OpenCV

Part 05

Vladimir Kulyukin
Department of Computer Science
Utah State University

Review

3 Big M's

- Mean – is the average of a set of values
- Median – a numerical value **v** right in the middle of the sorted sequence of values so that exactly half of the values in the sequence are less than v and half are greater than **v**
- Mode – the most frequent value in a set of values

Plotting Mean & Median

source in mean_median_mode.py

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
## use normal distribution to generate 10,000 points
## centered on 30,000 with an STD = 15,000
incomes = np.random.normal(30000, 15000, 10000)
## if you want to see an outlier, add this
## billionaire's income to the list of incomes
#incomes = np.append(incomes, [1000000000])
mn = np.mean(incomes)
print('mean    = ' + str(mn))
md = np.median(incomes)
print('median = ' + str(md))
## if you want to see a plot of incomes
plt.hist(incomes, 50)
plt.show()
```

Computing Mode

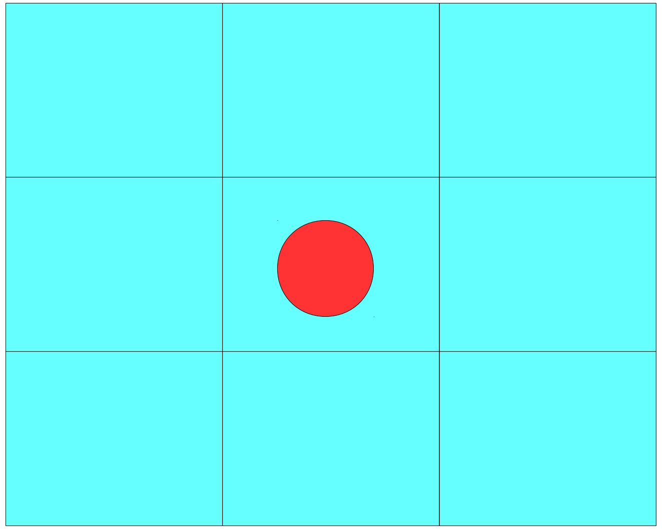
source in mean_median_mode.py

```
import numpy as np

import matplotlib.pyplot as plt

from scipy import stats

## create a random array of 500 ages from 10 up to 90.

ages = np.random.randint(10, high=90, size=500)

print(ages)

mo = stats.mode(ages)

## the result will print like (array([ 22.]), array([ 18.])),

## where the first number, 22, is the most frequent age

## and the second number, 18, is the number of times

##that age occurs in ages.

print(mo)
```

Pixel Masks

	0	1	2	…	N
0	128	255	10	201	203
1	120	35	50	25	137
2	34	89	190	197	108
…	180	178	215	37	45
M	24	25	91	225	225

3 x 3 Pixel Mask



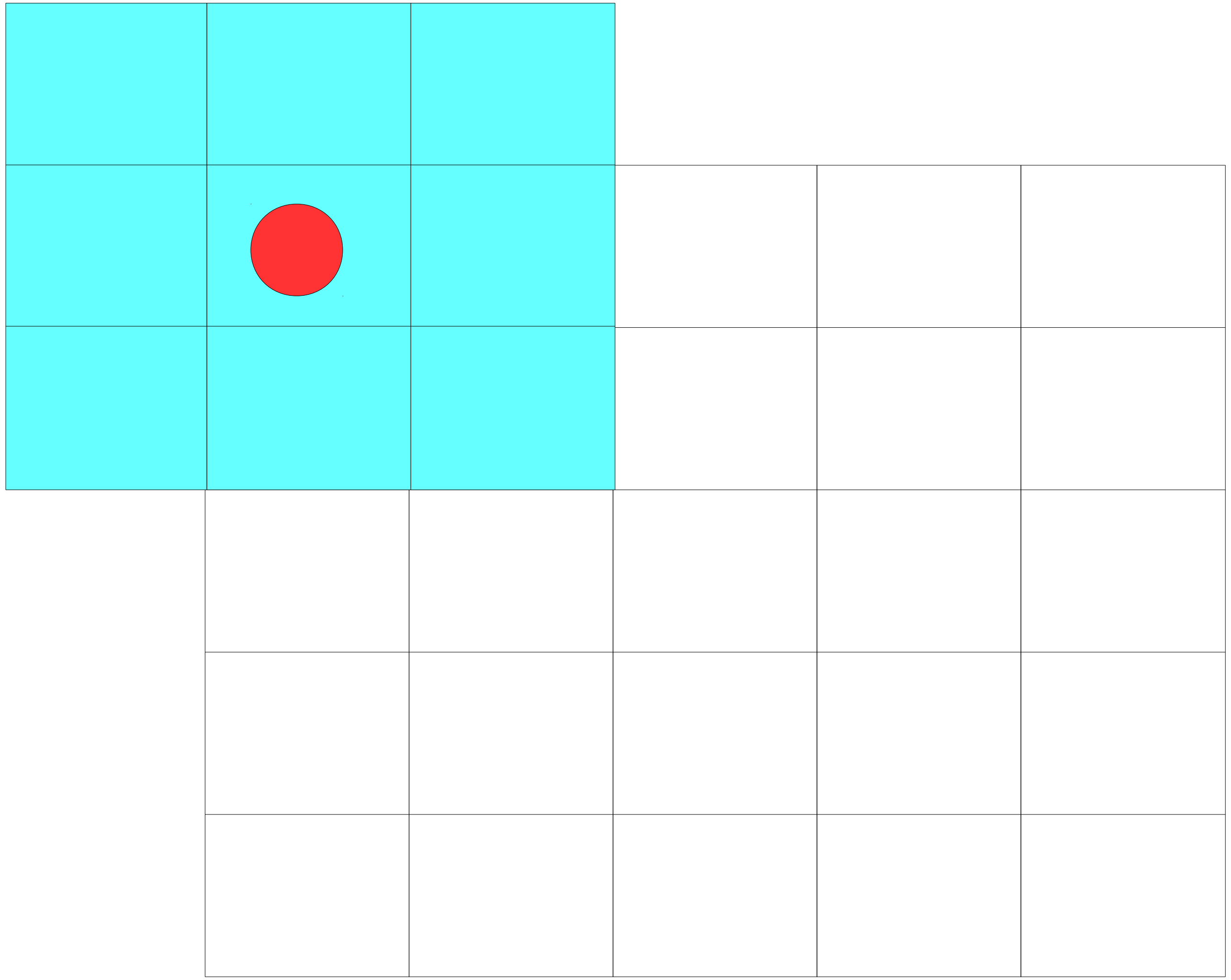
Pixel masks are used to compute various properties of the center pixel

Pixel Masks

	0	1	2	…	N
0	128	255	10	201	203
1	120	35	50	25	137
2	34	89	●	197	108
…	180	178	215	37	45
M	24	25	91	225	225

Pixel masks are superimposed on the image to compute various properties of the center pixel.

Border Pixel Problem



What happens when the mask is centered at a border pixel? Some pixels covered by the mask do not exist.

Two possible approaches:

1) Pad the image; this can be done virtually.

2) Do not center the mask at the border pixels.

Image Convolution: Applying Masks to Images

- Given an image **I** and a mask **M**, **M** is centered at each pixel (or a subset of pixels) and a value **v** is computed
- This value **v** is saved in an new image or the value of the current pixel on which **M** is centered is destructively modified with **v**
- This process is referred to as *image convolution*

Blurring

Blurring

- Blurring is a filtering operation
- A sharp image is an image where one can clearly see all objects
- Sharpness is a consequence of clear edges
- Why do we need to blur?

Why Blur?

- We may want to blur to make the image smoother (remove some small edges here and there) in the image to make subsequent processing more effective
- We may want to blur to create an artistic effect (e.g., motion blur)

Types of Blurring

- Mean filter
- Weighted average filter
- Median filter
- Gaussian filter
- Bilateral filter
- All these filters (and many more) are available in OpenCV

Problem

Write a program that takes a command line argument that specifies a path to an image, applies various blurring filters to the image and displays the results.

Sample run:

```
$ python blurring.py road01.png
```

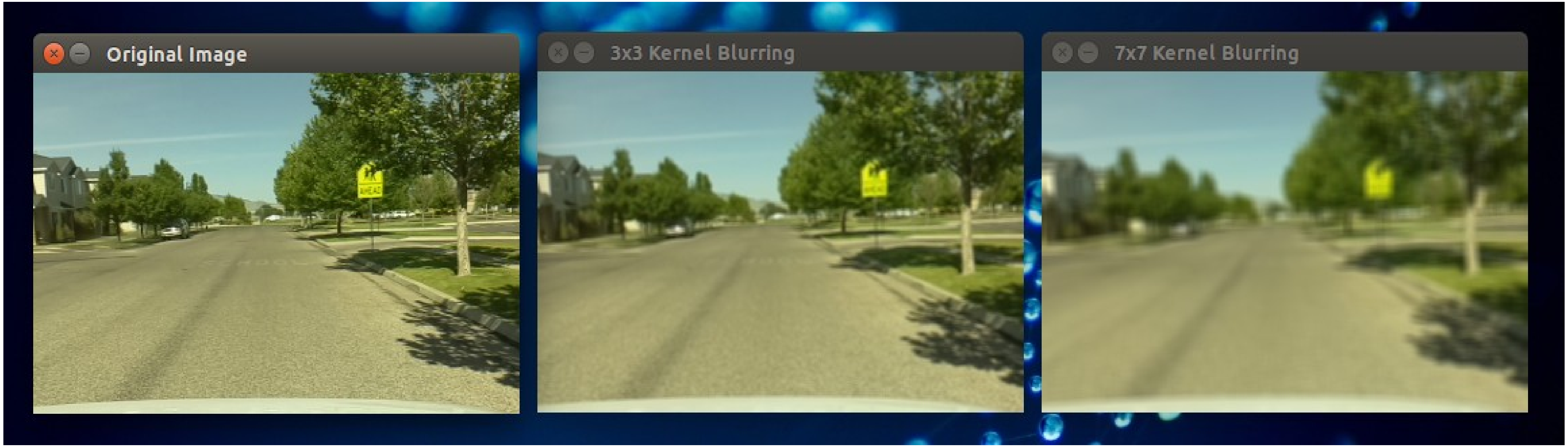
Mean Blurring

```
image = cv2.imread(sys.argv[1])
cv2.imshow('Original Image', image)


kernel_3x3 = np.ones((3, 3), np.float32) / 9


blurred = cv2.filter2D(image, -1, kernel_3x3)
cv2.imshow('3x3 Kernel Blurring', blurred)


kernel_7x7 = np.ones((7, 7), np.float32) / 49


blurred2 = cv2.filter2D(image, -1, kernel_7x7)
cv2.imshow('7x7 Kernel Blurring', blurred2))
```

What is -1? This means the depth (number of bits for each color in a single pixel) of the blurred image (blurred) will be the same as the depth of the original image (image)

source in blurring.py

Sample Run


Original Image
3x3 Kernel Blurring
7x7 Kernel Blurring
AHEAD

References

- https://en.wikipedia.org/wiki/Gaussian_blur
- www.opencv.org