Python & OpenCV

·AIPKU·

人工智能引论实践课 计算机视觉小班

主讲人：胡越予

一个例子

```python
import cv2
import numpy as np

img = cv2.imread('./example.jpg')
# Properties in image objects
print(img.shape)
print(img.dtype)
# Channels and slices
b = img[:,:,0]
cv2.imwrite('blue.png', b) # what is the result?

norm_img = img.astype(np.float64) / 255.
light_norm_img = np.power(norm_img, 0.5)
light_img = light_norm_img * 255.

cv2.imwrite('light.png', light_img)
```

安装 python 和 pip

Go to https://www.python.org/downloads/ .

Download Python installation executables.

Python 3.6.8 Recommended

Then go to https://pip.pypa.io/en/stable/installing/ .

Installing with get-pip.py

To install pip, securely download get-pip.py. [1]:

```
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
```

Then run the following:

```
python get-pip.py
```

Anaconda is also recommended.

安装 python 和 pip

```
Install numpy and OpenCV packages
> pip install numpy --user
> pip install opencv-python --user
> pip install scikit-image scikit-learn --user
# READ DOCS https://pip.pypa.io/en/stable/installing/
# Having Trouble ? Google IT!
```

基本数据类型

```
x = 3
print(x)        # Prints "3"
print(x + 1)    # Addition; prints "4"
print(x - 1)    # Subtraction; prints "2"
print(x * 2)    # 乘法; prints "6"
print(x ** 2)   # 乘方; prints "9"
x += 1
print(x)  # Prints "4"
x *= 2
print(x)  # Prints "8"
y = 2.5
print(y, y + 1, y * 2, y ** 2) # Prints "2.5 3.5 5.0 6.25"
```

Source and further reading:
http://cs231n.github.io/python-numpy-tutorial/#python-basic

基本数据类型

```
x = 3
print(x / 2)        # Prints "1.5"
print(x // 2)       # Addition; prints "1"
```

Source and further reading:
http://cs231n.github.io/python-numpy-tutorial/#python-basic

Bool 类型和字符串

```
t = True
f = False
print(t and f) # Logical AND; prints "False"
print(t or f)  # Logical OR; prints "True"
print(not t)   # Logical NOT; prints "False"
print(t != f)  # Logical XOR; prints "True"
```

```
h = 'hello'           # String literals can use single quotes
w = "world"           # or double quotes; it does not matter.
print(len(h))         # String length; prints "5"
hw = h + ' ' + w      # String concatenation "hello world"
hw12 = '%s %s %d' % (h, w, 12)  # string formatting
print(hw12)           # prints "hello world 12"
```

循环

```
animals = ['cat', 'dog', 'monkey']
for animal in animals:
    print(animal)
# Prints "cat", "dog", "monkey", each on its own line.
# 对于在animals里面的每个animal，输出它
# Good code speaks for itself.
```

```
animals = ['cat', 'dog', 'monkey']
for idx, animal in enumerate(animals):
    print('#%d: %s' % (idx + 1, animal))
# Prints “#1: cat”, “#2: dog”, "#3: monkey“
```

函数

```
def hello(name, loud=False):
    if loud:
        print('HELLO, %s!' % name.upper())
    else:
        print('Hello, %s' % name)

hello('Bob') # Prints "Hello, Bob"
hello('Fred', loud=True)  # Prints "HELLO, FRED!"
```

类和面向对象编程

```
class Greeter(object):
    # Constructor
    def __init__(self, name):
        self.name = name  # Create an instance variable
    # Instance method
    def greet(self, loud=False):
        if loud:
            print('HELLO, %s!' % self.name.upper())
        else:
            print('Hello, %s' % self.name)
g = Greeter(‘Fred’)  # 新建一个实例
g.greet()            # 调用成员函数 "Hello, Fred"
g.greet(loud=True)
```

numpy 数组

```
import numpy as np

a = np.array([1, 2, 3])   # Create a rank 1 array
print(type(a))            # Prints "<class 'numpy.ndarray'>"
print(a.shape)            # Prints "(3,)"
print(a[0], a[1], a[2])   # Prints "1 2 3"
a[0] = 5                  # Change an element of the array
print(a)                  # Prints "[5, 2, 3]"

b = np.array([[1,2,3],[4,5,6]])    # Create a rank 2 array
print(b.shape)                     # Prints "(2, 3)"
print(b[0, 0], b[0, 1], b[1, 0])   # Prints "1 2 4"
```

numpy 数据类型

```
import numpy as np

x = np.array([1, 2])   # Let numpy choose the datatype
print(x.dtype)         # Prints "int64"

x = np.array([1.0, 2.0])   # Let numpy choose the datatype
print(x.dtype)             # Prints "float64"

x = np.array([1, 2], dtype=np.int64)   # Force a particular
datatype
print(x.dtype)                         # Prints "int64"
```

常用数组

```
import numpy as np

a = np.zeros((2,2))   # Create an array of all zeros
print(a)              # Prints "[[ 0.  0.]
                      #          [ 0.  0.]]“

b = np.ones((1,2))    # Create an array of all ones
print(b)              # Prints "[[ 1.  1.]]“

c = np.full((2,2), 7) # Create a constant array
print(c)              # Prints "[[ 7.  7.]
                      #          [ 7.  7.]]“

d = np.eye(2)         # Create a 2x2 identity matrix
print(d)              # Prints "[[ 1.  0.]
                      #          [ 0.  1.]]"
```

数组下标操作

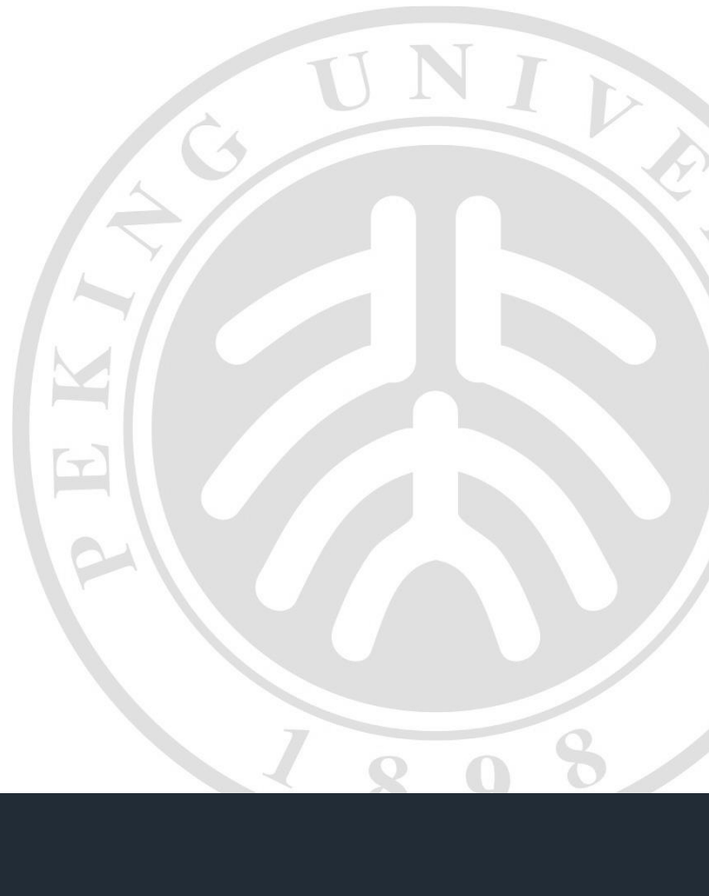
```
import numpy as np

# [[ 1  2  3  4]
#  [ 5  6  7  8]
#  [ 9 10 11 12]]
a = np.array([[1,2,3,4], [5,6,7,8], [9,10,11,12]])

# [[2 3]
#  [6 7]]
b = a[:2, 1:3]

print(a[[0, 1, 2], [0, 1, 0]])  # Prints "[1 6 9]"
```

数组拷贝

```
print(a[0, 1])   # Prints "2"

b[0, 0] = 77     # b[0, 0] is the same piece of data as a[0, 1]
print(a[0, 1])   # Prints "77“

d = a.copy()
d[0, 1] = 9999
print(a[0, 1])   # Prints "77"
```

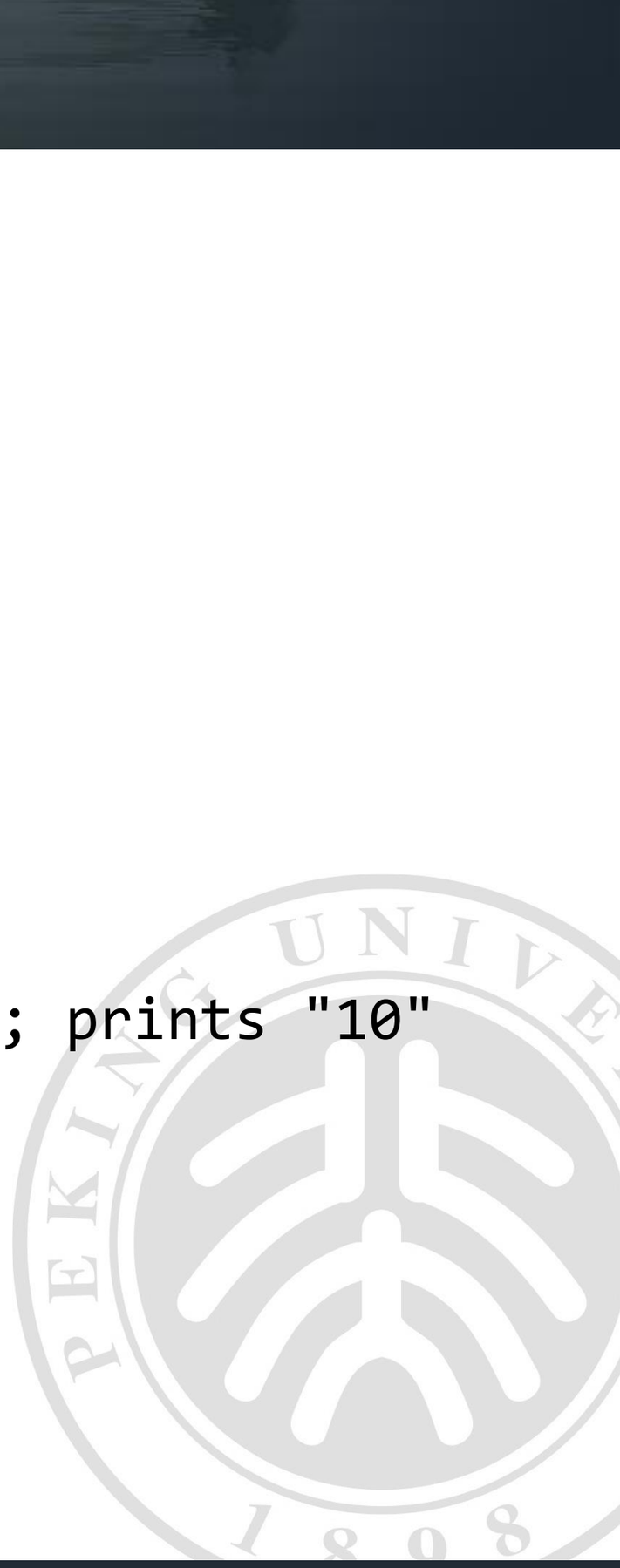
数组四则运算

```
import numpy as np

x = np.array([[1,2],[3,4]], dtype=np.float64)
y = np.array([[5,6],[7,8]], dtype=np.float64)

# [[ 6.0  8.0]
#  [10.0 12.0]]
print(x + y)

# [[-4.0 -4.0]
#  [-4.0 -4.0]]
print(x - y)

# [[ 5.0 12.0]
#  [21.0 32.0]]
print(x * y)
```



数组运算

```
# Elementwise division; both produce the array
# [[ 0.2         0.33333333]
#  [ 0.42857143  0.5       ]]
print(x / y)

# Elementwise square root; produces the array
# [[ 1.          1.41421356]
#  [ 1.73205081  2.        ]]
print(np.sqrt(x))

print(np.sum(x))  # Compute sum of all elements; prints "10"
print(np.sum(x, axis=0))  # prints "[4 6]"
print(np.sum(x, axis=1))  # prints "[3 7]"

print(np.mean(x)) # 2.5
print(np.mean(x, axis=0))  # prints "[2 3]"
print(np.mean(x, axis=1))  # prints "[1.5 3.5]"
```



Broadcasting

```
# Add a vector to each row of a matrix
x = np.array([[1,2,3], [4,5,6]])
v = np.array([1,2,3])  # v has shape (3,)

# Broadcasting, x and v have different shapes
# [[2 4 6]
#  [5 7 9]]
print(x + v)
```

矩阵运算

```
x = np.array([[1,2],[3,4]])
y = np.array([[5,6],[7,8]])

v = np.array([9,10])
w = np.array([11, 12])

# Inner product of vectors; both produce 219
print(v.dot(w))
print(np.dot(v, w))

# Matrix / vector product; both produce the rank 1 array [29 67]
print(x.dot(v))

# Matrix / matrix product; both produce the rank 2 array
# [[19 22]
#  [43 50]]
print(x.dot(y))
```

Reshape & Transpose

```
x = np.array([[[1,2],[3,4]],[[5,6],[7,8]]])

# (2, 2, 2)
print(x.shape)

# [[1 2 3 4]
#  [5 6 7 8]]
print(x.reshape((2,4)))
print(np.reshape(x, (2,4)))

# [[1 2]
#  [3 4]
#  [5 6]
#  [7 8]]
print(x.reshape((4,2)))
```

Reshape & Transpose

```
x = np.array([[[1,2],[3,4]],[[5,6],[7,8]]])

# [[[1 3]
#   [2 4]]
#
#  [[5 7]
#   [6 8]]]
print(x.transpose([0,2,1]))
```

Further Reading

Python & numpy Tutorial:
http://cs231n.github.io/python-numpy-tutorial/#python-basic

Official Python Tutorial:
https://docs.python.org/3/tutorial/

Official numpy Tutorial:
https://docs.scipy.org/doc/numpy/user/quickstart.html

回到例子

```python
import cv2
import numpy as np

img = cv2.imread('./example.jpg')
# Properties in image objects
print(img.shape)
print(img.dtype)
# Channels and slices
b = img[:,:,0]
cv2.imwrite('blue.png', b) # what is the result?

norm_img = img.astype(np.float64) / 255.
light_norm_img = np.power(norm_img, 0.5)
light_img = light_norm_img * 255.

cv2.imwrite('light.png', light_img)
```

example.jpg

blue.png

example.jpg

light.png

OpenCV 基础图像处理

> OpenCV 图像文件I/O

```
img = cv2.imread('example.jpg')
img = cv2.imread('example.png', -1)

cv2.imwrite('result.png', img)
cv2.imwrite('result.jpg', img,
    params=[cv2.IMWRITE_JPEG_QUALITY, 50])
```

OpenCV 基础图像处理

> OpenCV 色彩空间变换

```
cv2.cvtColor(src, code) → dst

# Example:
hsv_img = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
```

> OpenCV 重采样

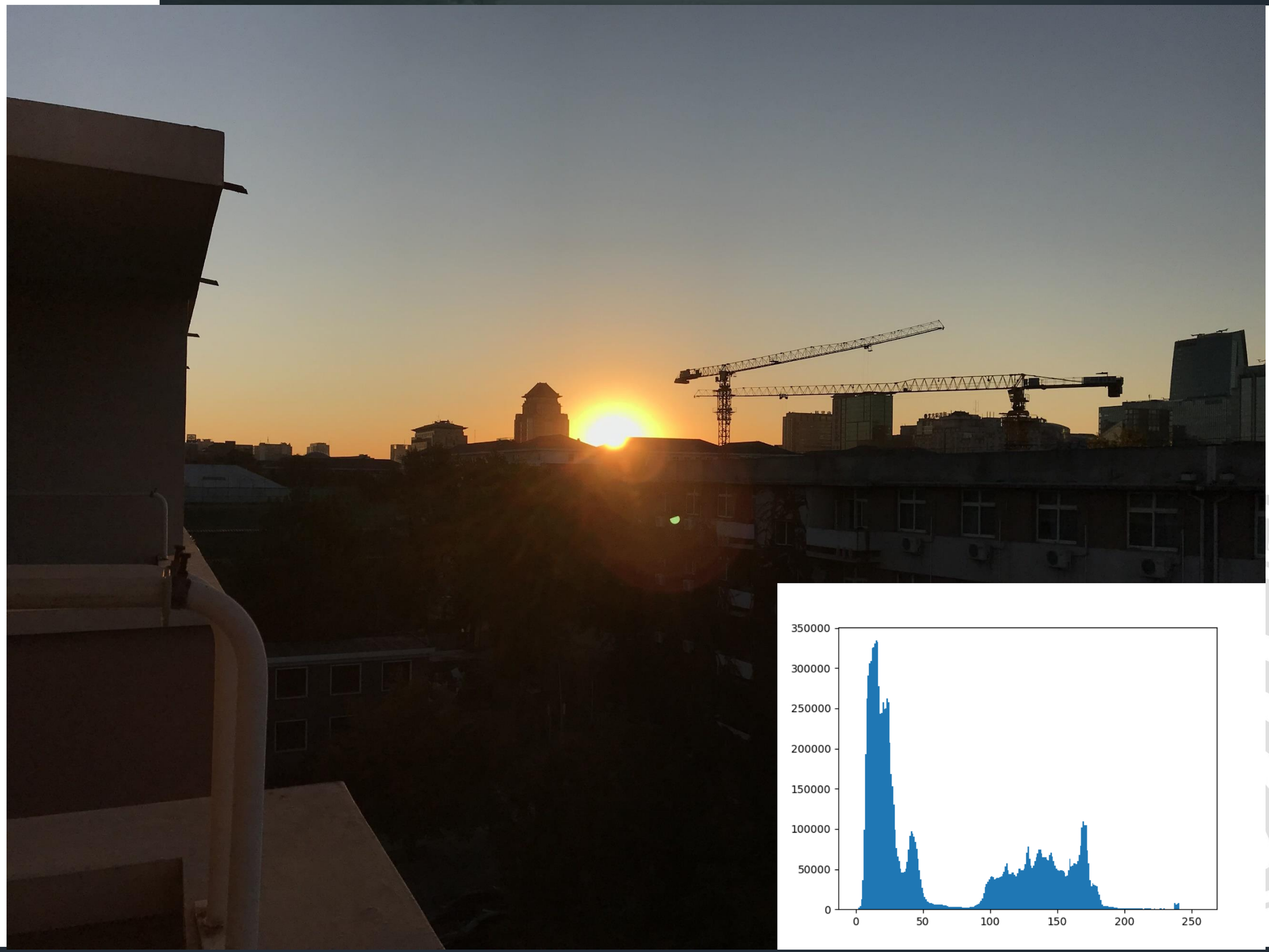
```
cv2.resize(src, dsize[, dst[, fx[, fy[, interpolation]]]])

# Example:
smaller_img = cv2.resize(img, (img.shape[1]*2, img.shape[0]*2),
   interpolation=cv2.INTER_CUBIC)

# Common interpolation methods: INTER_NEAREST, INTER_LINEAR,
INTER_AREA, INTER_CUBIC, INTER_LANCZOS4
```

matplotlib 样例

> 获取图像灰度直方图

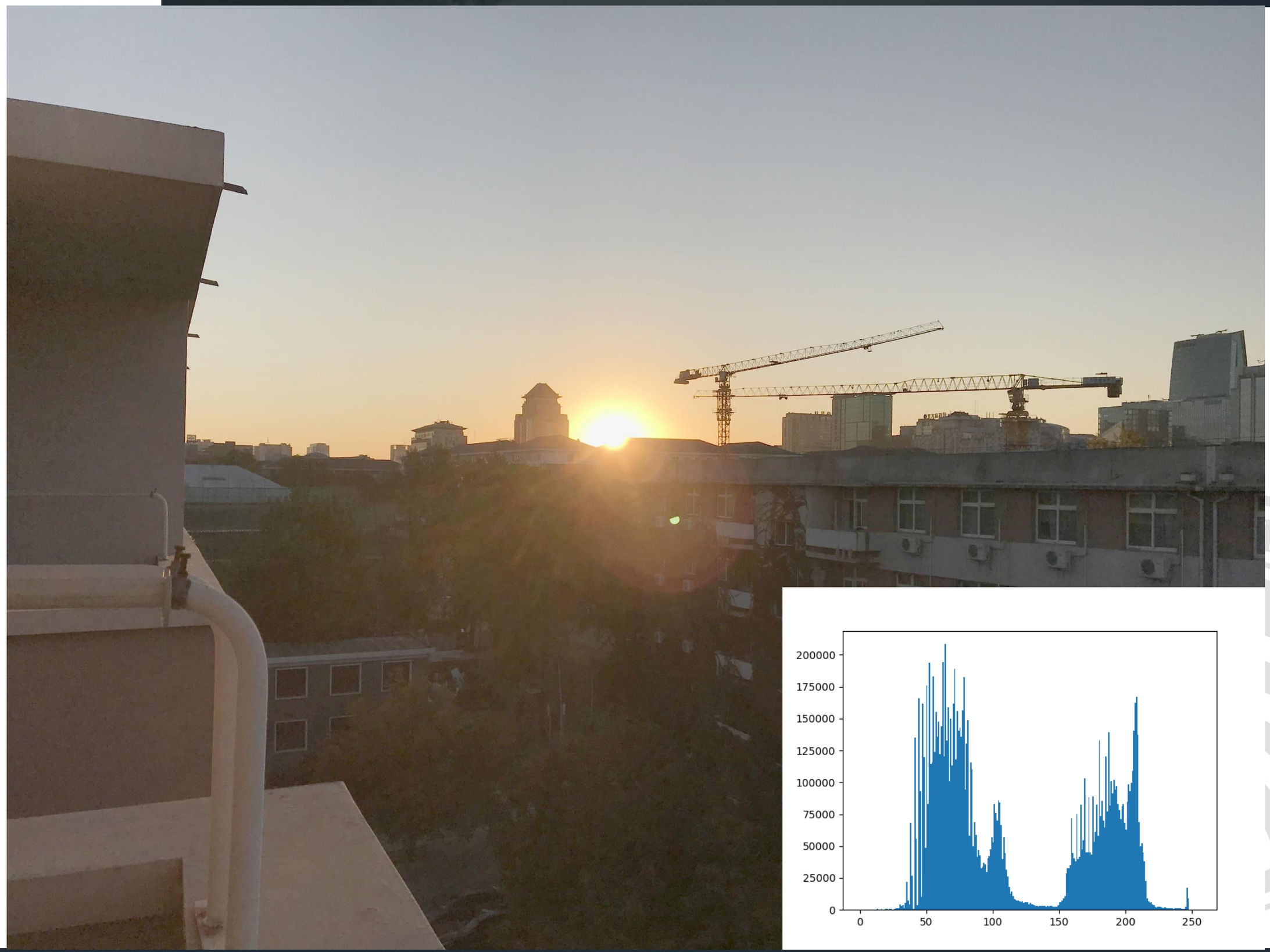
```
import matplotlib.pyplot as plt
import cv2import sys

img = cv2.imread(sys.argv[1])
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
flat_img = gray.reshape((-1))
plt.hist(flat_img, 256, (0,256))
plt.savefig('hist_' + sys.argv[1])
```

example.jpg



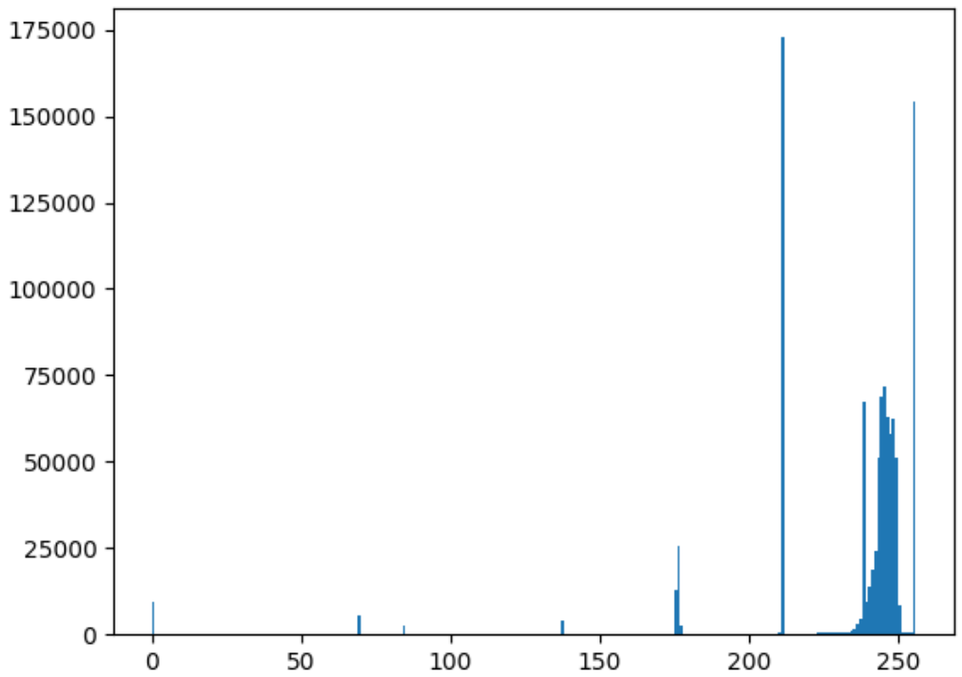
350000
300000
250000
200000
150000
100000
50000
0
0
50
100
150
200
250

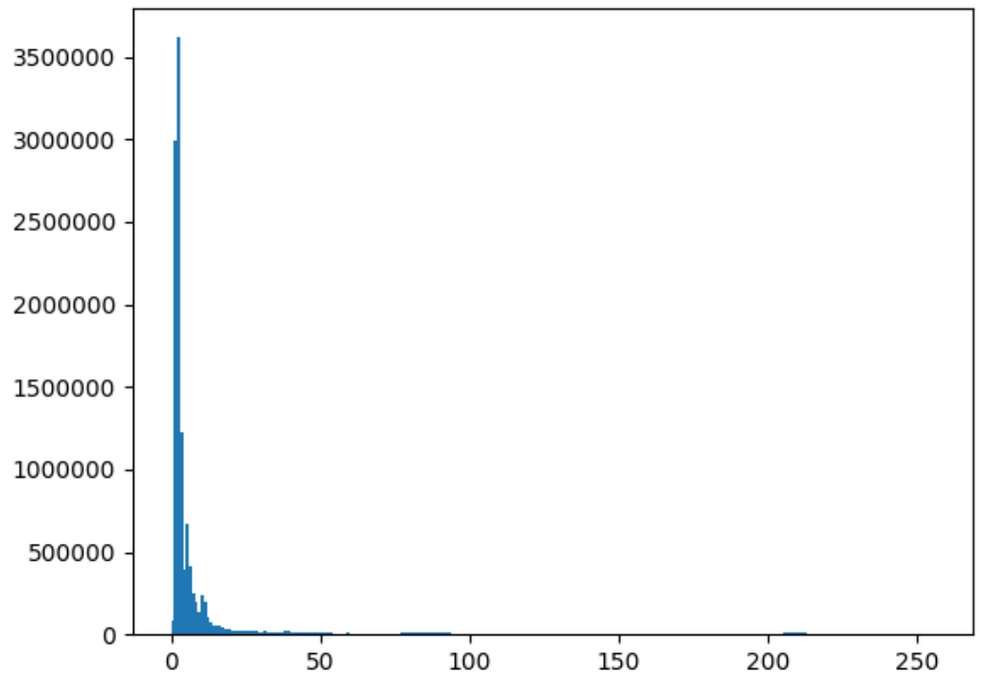
light.png


200000
175000
150000
125000
100000
75000
50000
25000
0
0
50
100
150
200
250

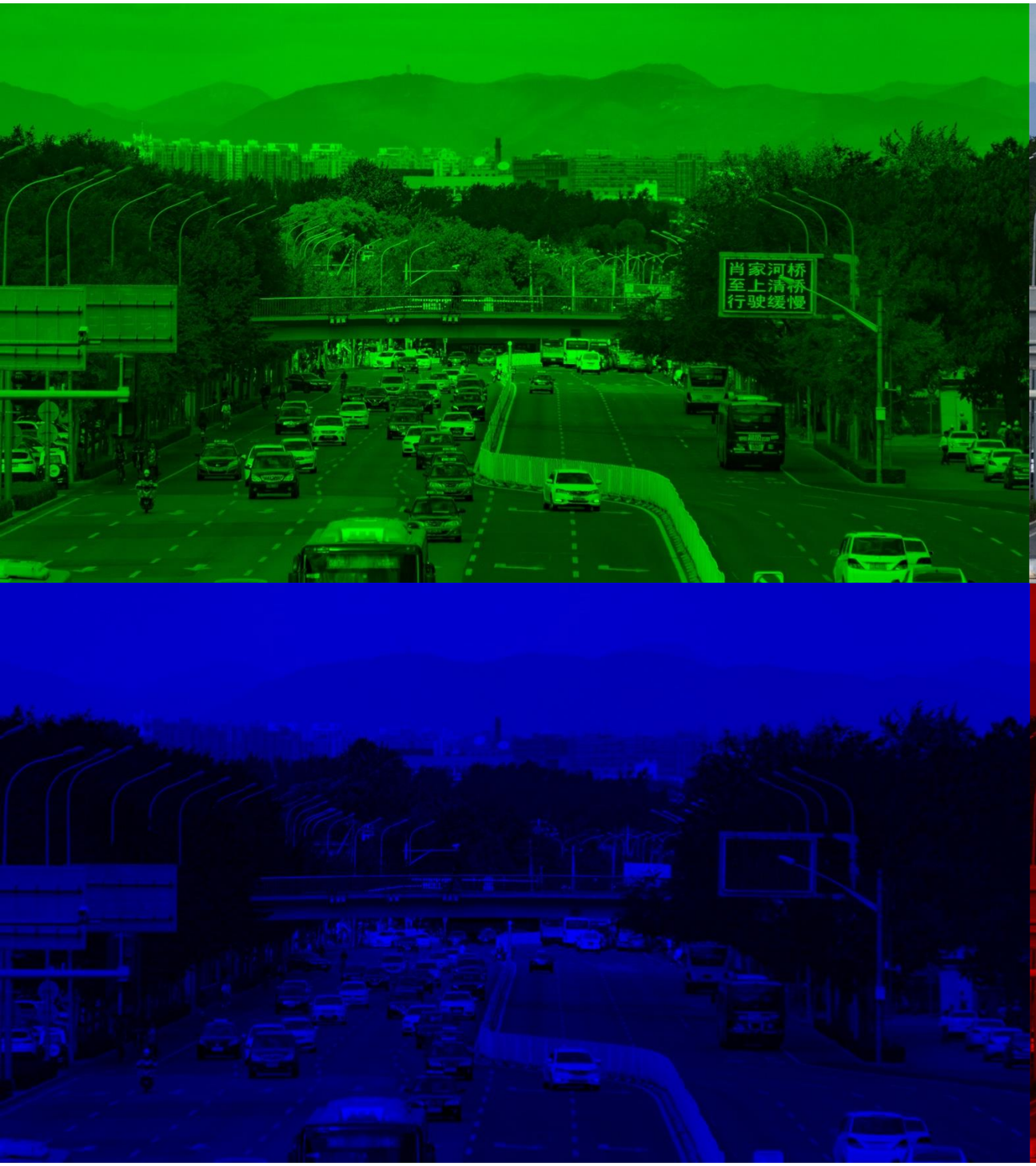
编程作业

1. 编写程序：程序读入目录下的 r.png，g.png，b.png 文件，分别对应三个通道的信息，将三个通道拼合成彩色图像，输出为 ans1.png
2. 以下给定了2个具有特点的直方图，请你通过各种方式获取得到直方图相似的图像，提交这些图像和对应的直方图。





Examples



一些可能有用的工具

系统平台

WSL, zsh (oh-my-zsh), msys2, tmux, conEmu

IDE/编辑器

Visual Studio Code, Sublime, PyCharm (edu邮箱有免费License)

Python环境

Anaconda, IPython, jupyter

遇到问题

遇到任何问题，请首先善用 Google。

Google后仍然不能解决，在询问前先说你做了哪些尝试。

鼓励互相讨论，也可以通过邮箱联系助教。

作业提交

编程作业：

1. 所有文件打包为 1.zip，在教学网对应位置提交
2. 本次作业截止日期为 3 月 17 日 周日 23：59：59

大班书面作业：

目前已经布置的作业下次小班课前交（3月14日晚上18:40）

之后的大班作业当周小班课前交

在教学网对应位置提交PDF文件