

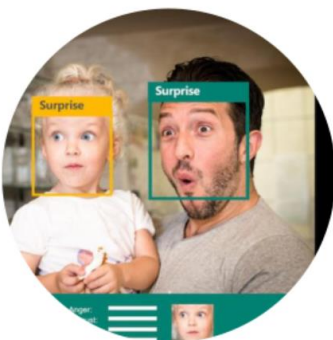


# 图像数据分析及挖掘

# 计算机视觉

## 什么是计算机视觉

- 从**图像**和视频中提取数值或符号信息的计算系统
- 让计算机能够和人类一样“看到并理解”图像
- 主要应用



### 图像理解—高级视觉

识别  
鉴别  
监测



### 图像分析—中级视觉

运动  
分割  
跟踪  
多视图几何



### 图像处理—低级视觉

线性滤波  
边缘检测  
纹理



### 图像获取—成像

相机模型  
相机标定  
辐射测定  
颜色

# 计算机视觉

## CV发展历史



# 计算机视觉库OpenCV

---

## OpenCV

- 开源的跨平台计算机视觉库
- 可运行在Linux、Windows、Android和MacOS操作系统上
- 轻量且高效，提供了Python、Ruby、Matlab等语言的接口
- 实现了图像处理和计算机视觉方面的很多通用算法



## OpenCV-Python

- 面向Python的OpenCV接口/库
- 支持NumPy数据结构

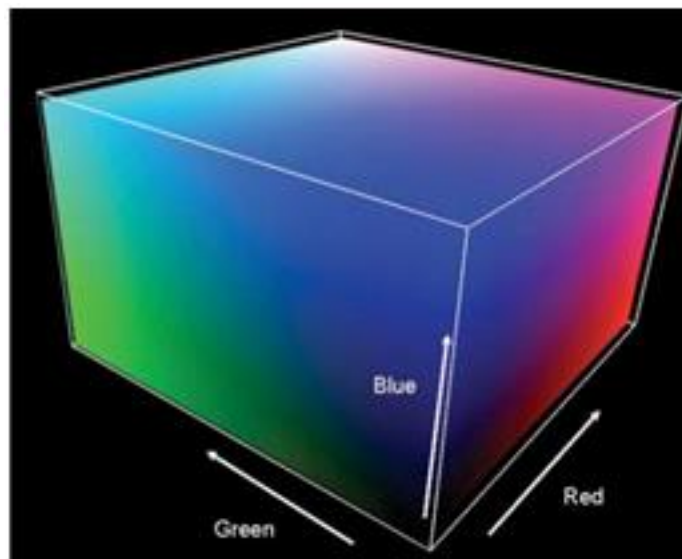
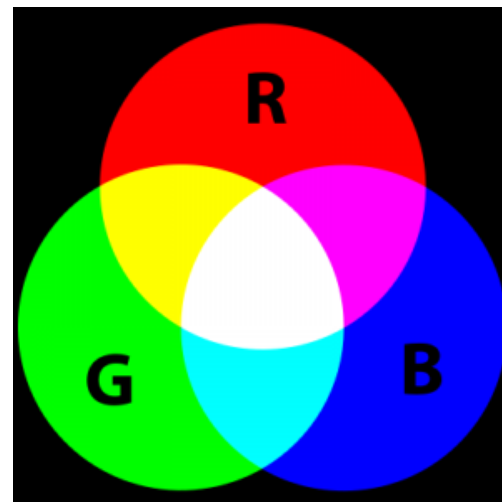
## 安装

- `pip install opencv-contrib-python`
-

# 图像数据基本概念

## RGB颜色空间

- 加法混色，彩色显示器
- 3通道
  - Red通道
  - Green通道
  - Blue通道
- 一个像素颜色值
  - $(b, g, r)$
- 取值范围
  - $[0, 255]$
  - $[0.0, 1.0]$



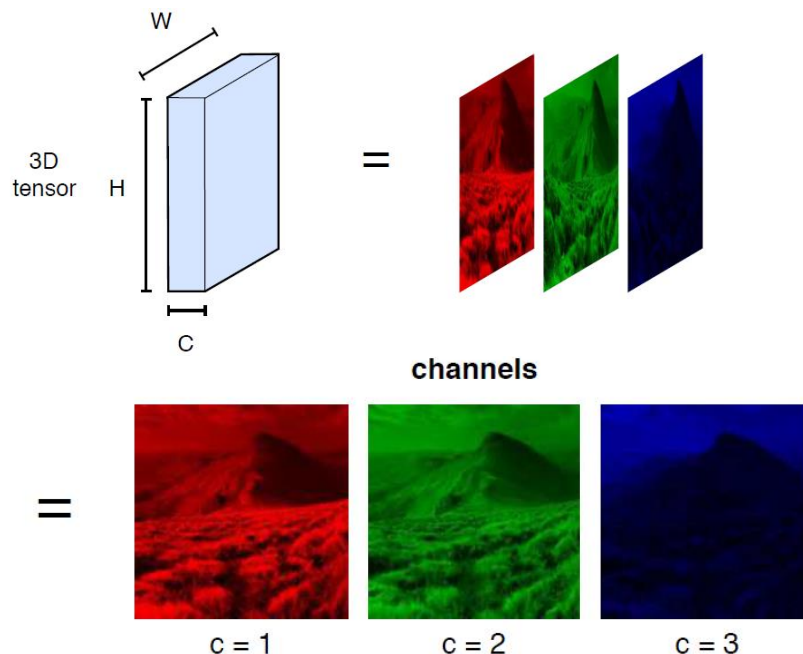


# 图像数据基本概念

## 主流颜色空间

- RGB三通道彩色图

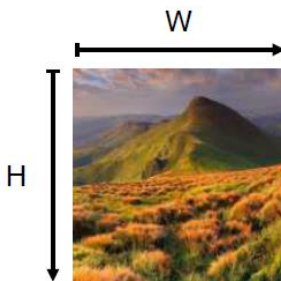
- 图片→3维矩阵  $([0,255])$



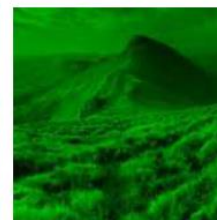
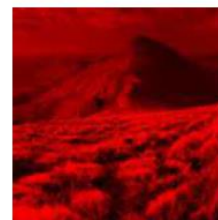
- 单通道灰度图

- 亮度信息  $([0,255])$

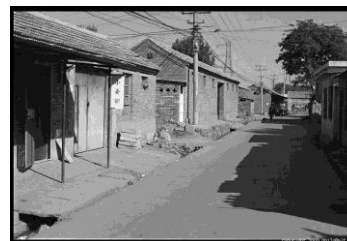
- $\text{Gray} = R \cdot 0.3 + G \cdot 0.59 + B \cdot 0.11$



=



灰度化



# 图像数据基本概念

---

## opencv的图像数据

- 图像数据是由NumPy的**多维数组 (ndarray)** 表示的
- 由opencv加载的图像数据可以调用其他常用的包进行处理和计算，如 matplotlib, scipy等

## 数据类型和像素值

- CV中图像的像素值通常有以下两种处理范围
    1. 0 - 255, 0: 黑色, 255: 白色
    2. 0 - 1, 0: 黑色, 1 : 白色
  - OpenCV读取的图像像素值范围是0-255
-

# 图像数据基本概念

---

## OpenCV图像IO

- 读取图像 `cv2.imread()`  
`cv2.IMREAD_COLOR` 读取彩色图像数据  
`cv2.IMREAD_GRAYSCALE` 读取灰度图像数据
  - 显示图像 `cv2.imshow()`, `cv2.waitKey(0)`, `cv2.destroyAllWindows()`
  - 保存图像 `cv2.imwrite()`
-



# 图像数据基本概念

---

## 图像数据

- 图像数据是多维数组，前两维表示了图像的高、宽第三维表示图像的通道个数，比如RGB，第三个维度为3，因为有三个通道；灰度图像没有第三个维度
- 分割和索引
  - 像操作ndarray一样操作即可

## 色彩空间, RGB, HSV, Gray...

- RGB转Gray, `cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)`

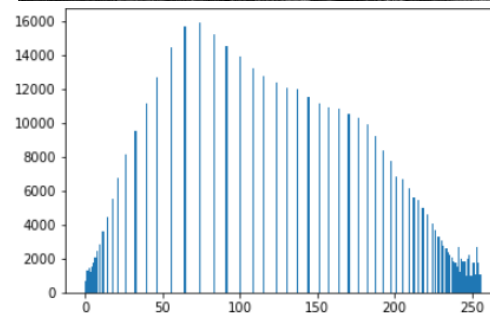
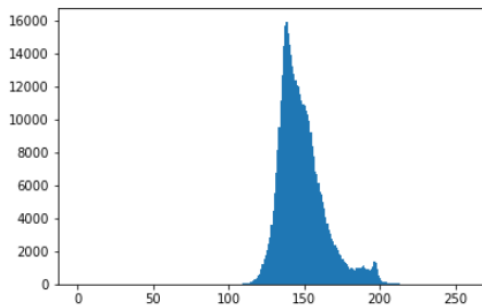
## 颜色直方图

- 直方图是一种能快速描述图像整体像素值分布的统计信息，  
`skimage.exposure.histogram`
    - 如：可以根据直方图选定阈值用于调节图像对比度
-

# 图像数据基本操作

## 直方图均衡化

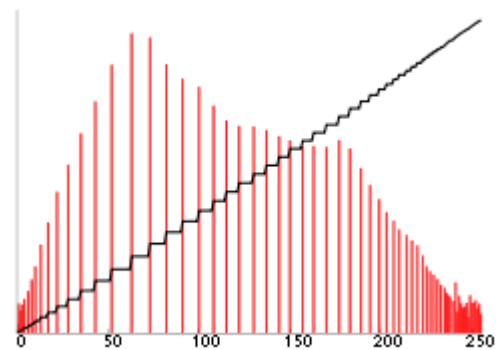
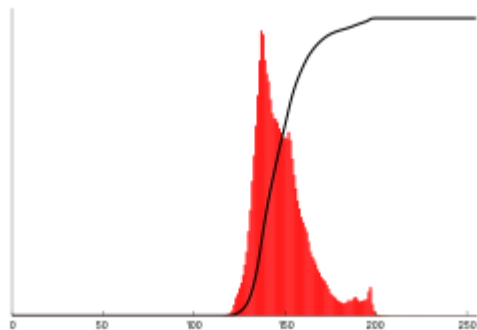
- 增强图像数据的对比度有利于特征的提取，不论是从肉眼还是算法来看都有帮助
- 自动调整图像的对比度 `cv2.equalizeHist(image)`



# 图像数据基本操作

---

## 直方图均衡化



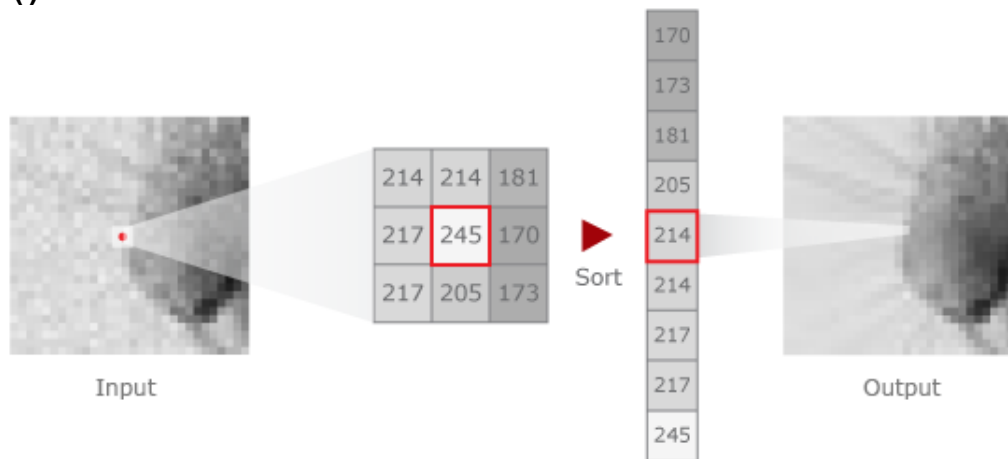
# 图像数据基本操作

## 图像滤波

- 滤波是处理图像数据的常用基础操作
- 滤波操作可以去除图像中的噪声点，由此增强图像的特征

## 中值滤波

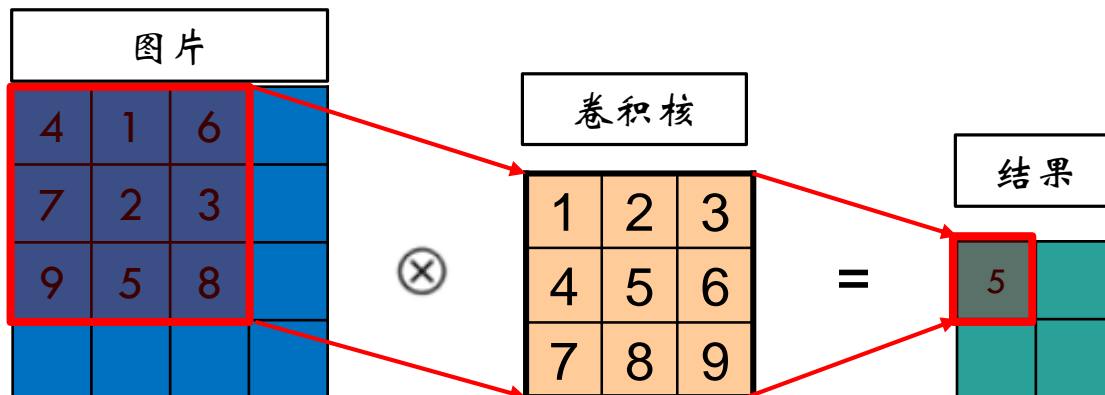
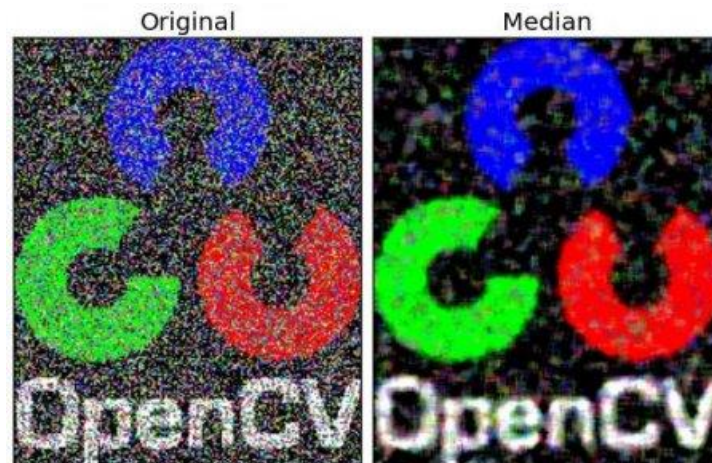
`cv2.medianBlur()`



# 图像数据基本操作

## 中值滤波

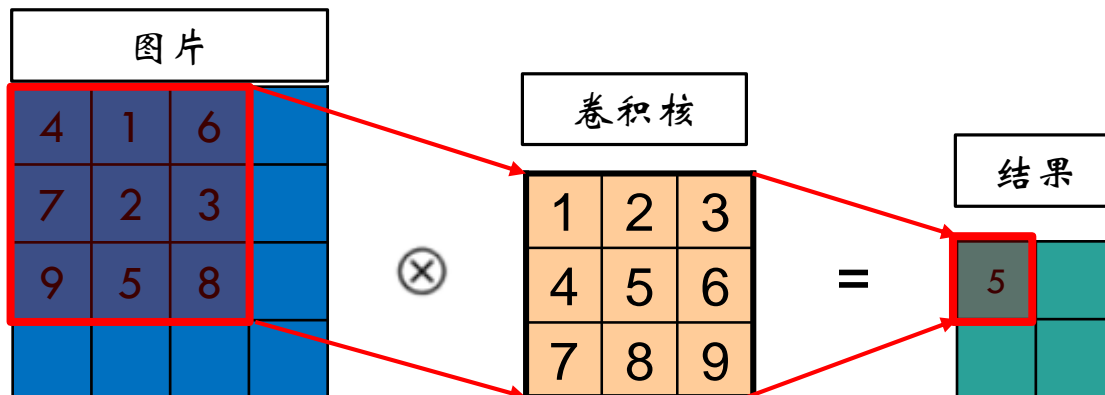
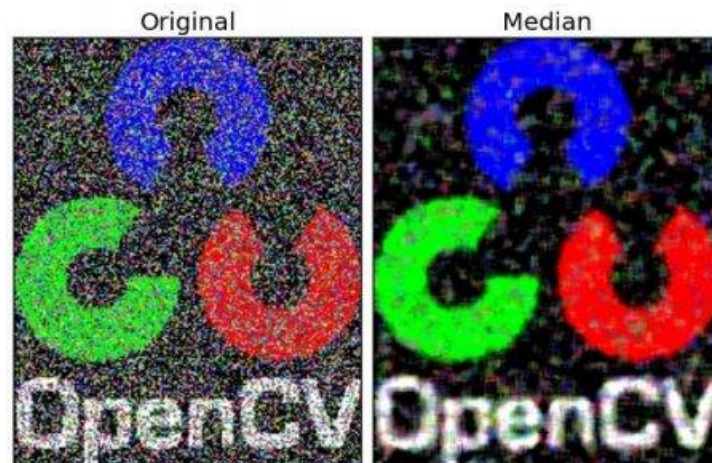
- 奇数尺寸
  - $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$ ,  $2n-1 \times 2n-1$
- 操作原理
  - 卷积域内的像素值从小到大排序
  - 取中间值作为卷积输出
  - 有效去除椒盐噪声



# 图像数据基本操作

## 中值滤波

- 奇数尺寸
  - $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$ ,  $2n-1 \times 2n-1$
- 操作原理
  - 卷积域内的像素值从小到大排序
  - 取中间值作为卷积输出
  - 有效去除椒盐噪声





# 图像数据基本操作

## 滤波/卷积

- 参数解释

- $x, y$  是像素在图片中的位置/坐标
- $k, l$  是卷积核中的位置/坐标
  - 中心点的坐标是  $(0,0)$
- $f[k, l]$  是卷积核中在  $(k, l)$  上的权重参数
- $I[x+k, y+l]$  是与  $f[k, l]$  相对应的图片像素值
- $h[x, y]$  是图片中  $(x, y)$  像素的滤波/卷积结果

滤波结果

滤波函数

像素邻域值

$$h[x, y] = \sum_{k, l} f[k, l] I[x + k, y + l]$$

$(-1,-1)$	$(-1,0)$	$(-1,1)$
$(0,-1)$	$(0,0)$	$(0,1)$
$(1,-1)$	$(1,0)$	$(1,1)$

