

OpenCV

轻松入门

面向Python

李立宗 著

OpenCV

轻松入门

面向Python

李立宗 著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书基于面向 Python 的 OpenCV (OpenCV for Python), 介绍了图像处理的方方面面。本书以 OpenCV 官方文档的知识脉络为主线, 并对细节进行补充和说明。

书中不仅介绍了 OpenCV 函数的使用方法, 还介绍了函数实现的算法原理。在介绍 OpenCV 函数的使用方法时, 提供了大量的程序示例。而且在介绍函数对图像的处理前, 往往先展示函数对数值、数组的处理, 方便读者从数值的角度观察和理解函数的处理过程和结果。在介绍具体的算法原理时, 本书尽量使用通俗易懂的语言和贴近生活的示例来说明问题, 避免使用过多复杂抽象的公式。

本书适合计算机视觉领域的初学者阅读, 包括在校学生、教师、专业技术人员、图像处理爱好者。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目 (CIP) 数据

OpenCV轻松入门: 面向Python / 李立宗著. —北京: 电子工业出版社, 2019.5
ISBN 978-7-121-36290-3

I. ①O… II. ①李… III. ①图象处理软件—程序设计 IV. ①TP391.413

中国版本图书馆CIP数据核字 (2019) 第067158号

策划编辑: 符隆美

责任编辑: 王中英

印 刷: 三河市君旺印务有限公司

装 订: 三河市君旺印务有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 787×1092 1/16 印张: 30.5 字数: 800.3 千字

版 次: 2019 年 5 月第 1 版

印 次: 2019 年 5 月第 1 次印刷

定 价: 99.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: 010-51260888-819, faq@phei.com.cn。

前言

本书是在作者多年从事计算机图形学教学和科研工作的基础上，结合多年的教学经验和研究成果编写而成的。在编写过程中，参考了国内外许多优秀的教材和有关文献，特别是清华大学出版的《计算机图形学》一书，受益匪浅。本书可作为高等院校计算机专业及相关专业的教材，也可供从事计算机图形学工作的工程技术人员参考。

谨以此书献给我的奶奶马井平女士。

本书的主要内容 and 特点

本书共分八章，主要内容包括：绪论、二维图形学、二维图形的表示方法、二维图形的变换、二维图形的消隐、二维图形的扫描转换、二维图形的裁剪、二维图形的反走样、二维图形的加速技术、二维图形的应用等。

本书力求做到概念清晰、重点突出、循序渐进、由浅入深、由易到难、由点到面、由理论到实践、由基础到应用。本书可作为高等院校计算机专业及相关专业的教材，也可供从事计算机图形学工作的工程技术人员参考。

本书可作为高等院校计算机专业及相关专业的教材，也可供从事计算机图形学工作的工程技术人员参考。

本书可作为高等院校计算机专业及相关专业的教材，也可供从事计算机图形学工作的工程技术人员参考。

本书可作为高等院校计算机专业及相关专业的教材，也可供从事计算机图形学工作的工程技术人员参考。

本书可作为高等院校计算机专业及相关专业的教材，也可供从事计算机图形学工作的工程技术人员参考。

本书可作为高等院校计算机专业及相关专业的教材，也可供从事计算机图形学工作的工程技术人员参考。

前言

目前，计算机视觉技术的应用越来越广泛。伴随着硬件设备的不断升级，构造复杂的计算机视觉应用变得越来越容易了。有非常多的软件工具和库可以用来构造计算机视觉应用，而面向 Python 的 OpenCV (OpenCV for Python) 就是一个很好的选择，本书正是基于面向 Python 的 OpenCV 来讲解的。

本书的主要内容和特点

OpenCV 本身是一个“黑盒”，它为我们提供了接口（参数、返回值）。我们只需要掌握接口的正确使用方法，就可以在完全不了解其内部工作原理（算法）的情况下，方便地进行各种复杂的图像处理。在这一点上，它和 Photoshop 等工具是相似的，只要掌握了正确的使用方法，就能够得到正确的处理结果。它们都尝试让我们专注于图像处理本身，而不用去考虑算法实现的细节。

在学习 Photoshop 时，我们学习的是如何使用它的功能，而不需要系统地学习每个功能所采用的算法原理。但是很明显，我们在使用 OpenCV 进行图像处理时，是不能完全忽略算法实现的，否则是不可能用好 OpenCV 的，更不能设计出好的计算机视觉应用系统。

从上述角度讲，我们可以从两个角度学习 OpenCV：

- 将 OpenCV 作为“白盒”学习：深入学习 OpenCV 每个函数所使用算法的基本原理、每个函数的具体实现细节，进一步加深对图像处理的深入理解。
- 将 OpenCV 作为“黑盒”学习：仅仅将 OpenCV 作为一个工具来使用，学习的是每个函数内参数的含义和使用方式，学习的目的是更好地使用 OpenCV 函数。

本书尽量帮助读者在“黑盒”学习和“白盒”学习之间取得平衡。在介绍具体的算法原理时，尽量使用通俗易懂的语言和贴近生活的示例来说明问题，避免使用过多复杂抽象的公式。希望这样的安排能够帮助读者更好地掌握计算机视觉的相关知识，更透彻地理解计算机视觉的相关算法。在介绍 OpenCV 函数的使用方法时，我们为读者提供了大量的程序示例。而且在介绍函数对图像的处理前，往往先展示函数对数值、数组的处理，方便读者从数值的角度观察和理解函数的处理过程和结果。希望这些例题能够帮助读者更好地理解 OpenCV 处理图像的方式，快速地掌握 OpenCV 的使用方法，更好地使用 OpenCV 进行图像处理。需要说明的一点是，本书为黑白印刷，无法很好地呈现某些程序的运行效果，请读者自行运行程序并观察结果。

在内容的设置上，本书以 OpenCV 官方文档的知识脉络为主线，在此基础上对细节进行补充和说明。

为了方便读者学习，本书力求将每一个知识点作为一个独立的点来介绍和说明。在介绍知识点时，尽量采用从零开始的方式，以避免读者在学习过程中需要不断地离开当前知识点，去查阅相关背景资料。但是由于篇幅有限，如果某一函数已经在前面介绍过，在后面用到该函数时，就没有对其进行重复介绍，而是给出介绍该函数语法的章节位置，方便读者参考阅读。

本书适合计算机视觉领域的初学者阅读，包括在校学生、教师、专业技术人员、图像处理爱好者。

感谢

首先，我要感谢我的老师高铁杠教授，感谢高老师带我走进了计算机视觉这一领域，让我对计算机视觉产生了浓厚的兴趣，更要感谢高老师一直以来对我的关心和帮助。

感谢本书的策划编辑符隆美老师，符老师的专业精神给我留下了非常深刻的印象。感谢本书的责任编辑王中英和许艳老师，她们对本书内容做出了细致修改，不仅修改了很多不通顺的语句和错别字，还对书中存在的技术问题进行了确认和修正。还要感谢为本书出版而付出辛苦工作的电子工业出版社的其他老师们。

感谢 OpenCV 开源库的所有贡献者。

感谢合作单位天津拨云咨询服务有限公司为本书提供的支持。

感谢我的家人，感谢你们一直以来对我的理解、支持和付出。

互动方式

限于本人水平，书中肯定存在很多不足之处，欢迎大家提出问题和建议，也非常欢迎大家和我交流关于 OpenCV 的各种问题，我的邮箱是 lilizong@gmail.com。

李立宗

2019年3月

读者服务

轻松注册成为博文视点社区用户 (www.broadview.com.cn)，扫码直达本书页面。

- **提交勘误：**您对书中内容的修改意见可在 [提交勘误](#) 处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **交流互动：**在页面下方 [读者评论](#) 处留下您的疑问或观点，与我们和其他读者一同学习交流。

页面入口：<http://www.broadview.com.cn/36290>



目录

第 1 章 OpenCV 入门	1
1.1 如何使用	1
1.2 图像处理基本操作	3
1.2.1 读取图像	3
1.2.2 显示图像	5
1.2.3 保存图像	9
1.3 OpenCV 贡献库	10
第 2 章 图像处理基础	11
2.1 图像的基本表示方法	11
2.2 像素处理	15
2.3 使用 <code>numpy.array</code> 访问像素	23
2.4 感兴趣区域 (ROI)	29
2.5 通道操作	32
2.5.1 通道拆分	32
2.5.2 通道合并	34
2.6 获取图像属性	36
第 3 章 图像运算	37
3.1 图像加法运算	37
3.1.1 加号运算符	37
3.1.2 <code>cv2.add()</code> 函数	38
3.2 图像加权和	40
3.3 按位逻辑运算	43
3.3.1 按位与运算	43
3.3.2 按位或运算	46
3.3.3 按位非运算	47

3.3.4 按位异或运算	48
3.4 掩模	49
3.5 图像与数值的运算	52
3.6 位平面分解	53
3.7 图像加密和解密	59
3.8 数字水印	63
3.8.1 原理	64
3.8.2 实现方法	66
3.8.3 例题	73
3.9 脸部打码及解码	74
第 4 章 色彩空间类型转换	77
4.1 色彩空间基础	77
4.1.1 GRAY 色彩空间	77
4.1.2 XYZ 色彩空间	78
4.1.3 YCrCb 色彩空间	78
4.1.4 HSV 色彩空间	79
4.1.5 HLS 色彩空间	80
4.1.6 CIEL*a*b*色彩空间	80
4.1.7 CIEL*u*v*色彩空间	81
4.1.8 Bayer 色彩空间	82
4.2 类型转换函数	82
4.3 类型转换实例	88
4.3.1 通过数组观察转换效果	88
4.3.2 图像处理实例	92
4.4 HSV 色彩空间讨论	93
4.4.1 基础知识	93
4.4.2 获取指定颜色	95
4.4.3 标记指定颜色	96
4.4.4 标记肤色	100
4.4.5 实现艺术效果	101
4.5 alpha 通道	102
第 5 章 几何变换	106
5.1 缩放	106
5.2 翻转	110

5.3	仿射	111
5.3.1	平移	112
5.3.2	旋转	113
5.3.3	更复杂的仿射变换	114
5.4	透视	115
5.5	重映射	117
5.5.1	映射参数的理解	117
5.5.2	复制	119
5.5.3	绕 x 轴翻转	121
5.5.4	绕 y 轴翻转	122
5.5.5	绕 x 轴、y 轴翻转	124
5.5.6	x 轴、y 轴互换	126
5.5.7	图像缩放	128
第 6 章	阈值处理	130
6.1	threshold 函数	130
6.1.1	二值化阈值处理 (cv2.THRESH_BINARY)	131
6.1.2	反二值化阈值处理 (cv2.THRESH_BINARY_INV)	133
6.1.3	截断阈值化处理 (cv2.THRESH_TRUNC)	135
6.1.4	超阈值零处理 (cv2.THRESH_TOZERO_INV)	136
6.1.5	低阈值零处理 (cv2.THRESH_TOZERO)	138
6.2	自适应阈值处理	139
6.3	Otsu 处理	141
第 7 章	图像平滑处理	144
7.1	均值滤波	146
7.1.1	基本原理	146
7.1.2	函数语法	150
7.1.3	程序示例	150
7.2	方框滤波	152
7.2.1	基本原理	152
7.2.2	函数语法	153
7.2.3	程序示例	154
7.3	高斯滤波	156
7.3.1	基本原理	156
7.3.2	函数语法	158

7.3.3	程序示例	159
7.4	中值滤波	159
7.4.1	基本原理	160
7.4.2	函数语法	161
7.4.3	程序示例	161
7.5	双边滤波	162
7.5.1	基本原理	162
7.5.2	函数语法	164
7.5.3	程序示例	164
7.6	2D 卷积	166
第 8 章	形态学操作	168
8.1	腐蚀	168
8.2	膨胀	173
8.3	通用形态学函数	178
8.4	开运算	179
8.5	闭运算	180
8.6	形态学梯度运算	182
8.7	礼帽运算	183
8.8	黑帽运算	185
8.9	核函数	186
第 9 章	图像梯度	189
9.1	Sobel 理论基础	189
9.2	Sobel 算子及函数使用	191
9.2.1	参数 ddepth	192
9.2.2	方向	195
9.2.3	实例	196
9.3	Scharr 算子及函数使用	200
9.4	Sobel 算子和 Scharr 算子的比较	204
9.5	Laplacian 算子及函数使用	206
9.6	算子总结	208
第 10 章	Canny 边缘检测	209
10.1	Canny 边缘检测基础	209

10.2 Canny 函数及使用	213
第 11 章 图像金字塔	215
11.1 理论基础	215
11.2 pyrDown 函数及使用	217
11.3 pyrUp 函数及使用	219
11.4 采样可逆性的研究	220
11.5 拉普拉斯金字塔	223
11.5.1 定义	223
11.5.2 应用	225
第 12 章 图像轮廓	229
12.1 查找并绘制轮廓	229
12.1.1 查找图像轮廓: findContours 函数	229
12.1.2 绘制图像轮廓: drawContours 函数	237
12.1.3 轮廓实例	238
12.2 矩特征	240
12.2.1 矩的计算: moments 函数	241
12.2.2 计算轮廓的面积: contourArea 函数	243
12.2.3 计算轮廓的长度: arcLength 函数	246
12.3 Hu 矩	248
12.3.1 Hu 矩函数	248
12.3.2 形状匹配	252
12.4 轮廓拟合	254
12.4.1 矩形包围框	254
12.4.2 最小包围矩形框	257
12.4.3 最小包围圆形	259
12.4.4 最优拟合椭圆	260
12.4.5 最优拟合直线	261
12.4.6 最小外包三角形	262
12.4.7 逼近多边形	263
12.5 凸包	266
12.5.1 获取凸包	267
12.5.2 凸缺陷	268
12.5.3 几何学测试	270
12.6 利用形状场景算法比较轮廓	275

12.6.1	计算形状场景距离	275
12.6.2	计算 Hausdorff 距离	278
12.7	轮廓的特征值	280
12.7.1	宽高比	280
12.7.2	Extent	281
12.7.3	Solidity	282
12.7.4	等效直径 (Equivalent Diameter)	283
12.7.5	方向	284
12.7.6	掩模和像素点	286
12.7.7	最大值和最小值及它们的位置	291
12.7.8	平均颜色及平均灰度	293
12.7.9	极点	294
第 13 章	直方图处理	297
13.1	直方图的含义	297
13.2	绘制直方图	301
13.2.1	使用 Numpy 绘制直方图	301
13.2.2	使用 OpenCV 绘制直方图	302
13.2.3	使用掩模绘制直方图	307
13.3	直方图均衡化	312
13.3.1	直方图均衡化原理	313
13.3.2	直方图均衡化处理	317
13.4	matplotlib 模块介绍	319
13.4.1	subplot 函数	319
13.4.2	imshow 函数	320
第 14 章	傅里叶变换	324
14.1	理论基础	324
14.2	Numpy 实现傅里叶变换	328
14.2.1	实现傅里叶变换	329
14.2.2	实现逆傅里叶变换	330
14.2.3	高通滤波示例	331
14.3	OpenCV 实现傅里叶变换	333
14.3.1	实现傅里叶变换	333
14.3.2	实现逆傅里叶变换	335
14.3.3	低通滤波示例	336

第 15 章 模板匹配.....	339
15.1 模板匹配基础.....	339
15.2 多模板匹配.....	345
第 16 章 霍夫变换.....	351
16.1 霍夫直线变换.....	351
16.1.1 霍夫变换原理.....	351
16.1.2 HoughLines 函数.....	357
16.1.3 HoughLinesP 函数.....	359
16.2 霍夫圆环变换.....	361
第 17 章 图像分割与提取.....	364
17.1 用分水岭算法实现图像分割与提取.....	364
17.1.1 算法原理.....	364
17.1.2 相关函数介绍.....	366
17.1.3 分水岭算法图像分割实例.....	375
17.2 交互式前景提取.....	376
第 18 章 视频处理.....	383
18.1 VideoCapture 类.....	383
18.1.1 类函数介绍.....	383
18.1.2 捕获摄像头视频.....	387
18.1.3 播放视频文件.....	388
18.2 VideoWriter 类.....	389
18.2.1 类函数介绍.....	389
18.2.2 保存视频.....	391
18.3 视频操作基础.....	392
第 19 章 绘图及交互.....	393
19.1 绘画基础.....	393
19.1.1 绘制直线.....	394
19.1.2 绘制矩形.....	394
19.1.3 绘制圆形.....	395
19.1.4 绘制椭圆.....	397
19.1.5 绘制多边形.....	398
19.1.6 在图形上绘制文字.....	400
19.2 鼠标交互.....	402

19.2.1	简单示例	404
19.2.2	进阶示例	405
19.3	滚动条	407
19.3.1	用滚动条实现调色板	408
19.3.2	用滚动条控制阈值处理参数	409
19.3.3	用滚动条作为开关	410
第 20 章	K 近邻算法	412
20.1	理论基础	412
20.2	计算	415
20.2.1	归一化	415
20.2.2	距离计算	416
20.2	手写数字识别的原理	417
20.3	自定义函数手写数字识别	421
20.4	K 近邻模块的基本使用	427
20.5	K 近邻手写数字识别	429
第 21 章	支持向量机	431
21.1	理论基础	431
21.2	SVM 案例介绍	434
第 22 章	K 均值聚类	439
22.1	理论基础	439
22.1.1	分豆子	439
22.1.2	K 均值聚类的基本步骤	441
22.2	K 均值聚类模块	441
22.3	简单示例	442
第 23 章	人脸识别	448
23.1	人脸检测	448
23.1.1	基本原理	448
23.1.2	级联分类器的使用	451
23.1.3	函数介绍	452
23.1.4	案例介绍	453
23.2	LBPH 人脸识别	454
23.2.1	基本原理	454
23.2.2	函数介绍	456

23.2.3 案例介绍	457
23.3 EigenFaces 人脸识别	458
23.3.1 基本原理	458
23.3.2 函数介绍	459
23.3.3 案例介绍	460
23.4 Fisherfaces 人脸识别	461
23.4.1 基本原理	461
23.4.2 函数介绍	463
23.4.3 案例介绍	464
23.5 人脸数据库	465
参与文献	467
附录 A 范例	470

第 1 章

OpenCV 入门

OpenCV 是一个开源的计算机视觉库，1999 年由英特尔的 Gary Bradski 启动。Bradski 在访学过程中注意到，在很多优秀大学的实验室中，都有非常完备的内部公开的计算机视觉接口。这些接口从一届学生传到另一届学生，对于刚入门的新人来说，使用这些接口比重复造轮子方便多了。这些接口可以让他们在之前的基础上更有效地开展工作。OpenCV 正是基于为计算机视觉提供通用接口这一目标而被策划的。

由于要使用计算机视觉库，用户对处理器（CPU）的要求提升了，他们希望购买更快的处理器，这无疑会增加英特尔的产品销量和收入。这也许就解释了为什么 OpenCV 是由硬件厂商而非软件厂商开发的。当然，随着 OpenCV 项目的开源，目前其已经得到了基金会的支持，很大一部分研究主力也转移到了英特尔之外，越来越多的用户为 OpenCV 做出了贡献。

OpenCV 库由 C 和 C++ 语言编写，涵盖计算机视觉各个领域内的 500 多个函数，可以在多种操作系统上运行。它旨在提供一个简洁而又高效的接口，从而帮助开发人员快速地构建视觉应用。

OpenCV 更像一个黑盒，让我们专注于视觉应用的开发，而不必过多关注基础图像处理的具体细节。就像 PhotoShop 一样，我们可以方便地使用它进行图像处理，我们只需要专注于图像处理本身，而不需要掌握复杂的图像处理算法的具体实现细节。

本章将介绍 OpenCV 的具体配置过程及基础使用方法。

1.1 如何使用

Python 的开发环境有很多种，在实际开发时我们可以根据需要进行选择一种适合自己的。在本书中，我们选择使用 Anaconda 作为开发环境。本节简单介绍如何配置环境，来实现在 Anaconda 下使用基于 Python 语言的 OpenCV 库。

1. Python 的配置

在本书中，我们使用的是 Python 3 版本。虽然 Python 2 和 Python 3 有很多相同之处，以至于 Python 2 的读者也可以使用本书，但还是要说明一下，本书直接面向的版本是 Python 3。

可以在 Python 的官网上 (<https://www.python.org/downloads/>) 下载 Python 3 的解释器。在该网页顶部已经指出了最新的版本，例如“Download Python 3.7.0”就表示当前（2018 年 10 月 9 日）的最新版本是 Python 3.7.0，单击“Download Python 3.7.0”，就会开始下载解释器软件。

如果想安装其他版本，可以向下拖动鼠标，你将看到一个滚动页面，其中显示了一个列表集，其中列出了不同的安装包，具体选择哪种安装包依赖于两个因素：

- 操作系统，例如 Windows、macOS、Linux 等。
- 处理器位数，例如 32 位或者 64 位。

根据自己的计算机配置，在列表集中选择对应的安装包下载。

下载完成后，按照提示步骤完成安装即可。

2. Anaconda 的配置

可以在 Anaconda 的官网上 (<https://www.anaconda.com/download/>) 下载 Anaconda。在下载页顶部指出了当前的最新版本。

如果想安装其他版本，可以在下载页内根据实际情况选择。具体选择哪种安装包依赖于三个因素：

- Python 的版本，例如 Python 2.7 或者 Python 3.7 等。
- 操作系统，例如 Windows、macOS、Linux 等。
- 处理器位数，例如 32 位或者 64 位。

根据自己的环境配置，在下载页中选择对应的安装包下载。

下载完成后，按照提示步骤完成安装即可。

3. OpenCV 的配置

可以从官网下载 OpenCV 的安装包，编译后使用；也可以直接使用第三方提供的预编译包安装。

在本书中，我们选择由 PyPI 提供的 OpenCV 安装包，可以在 <https://pypi.org/project/opencv-python/> 上面下载最新的基于 Python 的 OpenCV 库。同样，下载页顶部指出了当前的最新版本。

如果想安装其他版本，可以在 Download files 栏目内根据实际情况选择。具体选择哪种安装包依赖于三个因素：

- Python 的版本，例如 Python 2.7 或者 Python 3.7 等。
- 操作系统，例如 Windows、macOS、Linux 等。
- 处理器位数，例如 32 位或者 64 位。

完成下载后，在 Anaconda Prompt 内使用 `pip install` 完整路径文件名完成安装。例如，假设文件存储在 `D:\anaconda\Lib` 目录下面，则要使用的语句为：

```
>>>pip install D:\anaconda\Lib\opencv_python-3.4.3.18-cp37-cp37m-win_amd64.whl
```

安装完成后，可以在 Anaconda Prompt 内使用 `conda list` 语句查看安装是否成功。如果安装成功，就会显示安装成功的 OpenCV 库及对应的版本等信息。需要注意，不同安装包的名称及版本号可能略有差异。例如，图 1-1 中的包列表显示了系统内配置了 OpenCV 其他版本的情况，该图说明在当前系统内配置的是 OpenCV 3.4.2 版本。