
Practical Content-based Image Retrieval Handbook

基于内容的图像检索 实用手册



一份以示例为驱动介绍图像处理和计算机视觉指南
书

整理：袁勇

整理时间：September 28, 2017

Email: yongyuanstu@gmail.com

Version: 2.00

目 录



1	简介	1
2	Python 及需要的安装包	3
2.1	NumPy 和 SciPy	3
2.1.1	Windows	3
2.1.2	OSX	4
2.1.3	Linux	4
2.2	Matplotlib	4
2.2.1	全平台	4
2.3	OpenCV	4
2.3.1	Windows 和 linux	5
2.3.2	OSX	5
2.4	Mahotas	5
2.4.1	全平台	5
2.4.2	跳过安装	5
3	载入、显示及保存	7
4	图像基础	10
5	特征编码	11
5.0.1	Fisher Vector	11
5.0.2	VLAD	11
	参考文献	12

第 1 章 简介



计算机视觉的最终目标是要理解图像中的所展现的内容。对于人类而言，理解图像中的内容是极其容易的，但是对计算机而言，该任务却困难重重。

那为什么还要不辞劳苦地去学习计算机视觉呢？

因为图像无处不在！

无论是在你的智能手机上的个人照片，还是 Facebook 上的公共照片，亦或是 YouTube 上的视频，相比于过去我们有了更多的图像——并且我们需要方法来对这些图像的内容进行分析、分类，量化。

比如，你最近是否在 Facebook 上对你自己的或你朋友的照片加过标记？Facebook 是怎样知道一幅图像中人脸的位置在哪里的？

Facebook 已经在他们的网站上实现了人脸识别算法，那意味着他们不仅可以在一幅图像中找到人脸，而且他们还可以辨别出这张人脸是谁的。人脸识别技术是计算机视觉在真实世界里的一个应用。

在计算机视觉中还有没有其他类型的有用的应用呢？

当然有，我们可以利用公共图像资源库如 Flickr 创建我们的三维世界表示。我们可以下载由市民用他们的智能手机和照相机拍摄的成千上万的曼哈顿照片，然后对它们进行分析，并对它们进行组织来构建曼哈顿城市的三维表示，我们还可以通过我们的计算机进行该城市的可视化导航。这听起来是不是很酷？

计算机视觉另一个很火的应用是监控。

虽然监控往往有各种各样的负面含义，但是监控类型还是各有差异的。其中一类监控涉及到视频安全分析，比如抢劫后寻找可能的犯罪嫌疑人。

不过还有另一类类型的在零售商店的监控。百货公司可以使用校准后的照相机来跟踪你怎么走过他们的商店，并且在哪些货架子你停下了。

你上一次去最喜欢的服装店时，你有没有在一件春天的最新潮流的牛仔裤前驻足？你在看那条牛仔裤时看了多久？你在看这些牛仔裤时的面部表情是怎样的？你有没有挑出来一条在试衣间里试穿它？所有这些类型的问题计算机视觉监控系统是能够回答的。

计算机视觉还可以应用于医学领域。一年前，我咨询了美国国家癌症研究所，该研究所研发出来了为发现癌症危险因素而能对乳腺癌组织图像进行自动分析。通常，类似这样的任务需要一个多年训练有素的病理学家——并且这将是非常耗时的！

我们的研究证明了计算机视觉可以应用于这些图像，并且能够自动的分析、量化这些细胞结构——没有人工交互！现在，我们分析乳腺组织学图像为癌症的危险因素比以往要快得多了。

当然，计算机视觉还可以用于其他的医学领域，可以用计算机视觉算法分析 X-射线，MRI 扫描图像以及细胞结构。

你可能听说过的最成功的计算机视觉的成功故事是 X-Box 360 Kinect。Kinect 用立体相机来理解图像的深度，使得它能够分辨出人类的姿势，当然，这必须得借助一些机器学习的东西。

上面列举的计算机视觉应用并不止于此。

现在计算机视觉在你生活的许多领域盛行，不管你有没有意识到它。我们用计算机视觉算法来分析电影，足球赛，手势识别（手语），车牌（比如你开车开得很快），医学，手术，军事，零售。

我们甚至可以将计算机视觉用于航天。



第 2 章 Python 及需要的安装包



要探索计算机视觉世界，首先我们需要安装一些安装包。作为探索计算机视觉之旅的起始，要安装这些安装包特别是 OpenCV 是非常冗余乏味的，它依赖于你所使用的操作系统。我已经试着将这些安装说明简化成了一份简短的使用指南，不过，正如你所了解的，锁着项目、网站的改变，这些安装说明是会变化的！如果你在安装中遇到了问题，确保查看安装包的网站以获得最新安装说明。

我强力推荐你要么用 `easy_install` 或 `pip` 来管理你那些安装包。它会使你的生活变得更简单！

最后，如果你不想自己逐一安装这些安装包的话，我已经预先安装好后所有需要的安装包，并把它们放在一个 Ubuntu 的虚拟机里。使用该虚拟机可以使你跳过那些繁琐的安装包的安装，直接进入本书的实例中，而不必理会这些安装包的管理、安装说明以及编译错误。

要想获得更多这个预先配置好的虚拟机，可以前往<http://www.pyimagesearch.com/practical-python-opencv/>查看详细信息。

现在，让我们来安装这些安装包！

2.1 NumPy 和 SciPy

NumPy 是一个用于 Python 编程语言的库，它对大型多维数组提供支持。为什么 NumPy 很重要呢？原因是使用 NumPy，我们可以将图像表示为一个多维数组。将图像表示为 NumPy 数组，不仅易于计算和资源得以有效利用，而且一些其他的图像处理操作和机器学习库也是用 NumPy 数组来表示的。此外，使用 NumPy 内置的高水准数学函数，我们能在了一幅图像上做快速数值分析。

与 NumPy 形影不离的是 SciPy。SciPy 为科学和技术计算提供了更多的支持。

2.1.1 Windows

到目前为止，在你的 Windows 系统上安装 NumPy 和 SciPy 最简单的方式是<http://www.scipy.org/install.html>下载并安装二进制分发版。

2.1.2 OSX

如果你运行的是 OSX 10.7.0(Lion) 或更高版本，NumPy 和 SciPy 已经预先安装好了。

不过，我喜欢安装 ScipySuperpack。它包含了最新版的 NumPy、SciPy、Matplotlib 和其他非常有用的安装包，比如 ipython、pandas 和 scikit-learn。所有的这些安装包是值得安装的，并且如果你在www.PylImageSearch.com上阅读我的博客，你会发现这些安装包我用得非常的频繁。

2.1.3 Linux

很多 Linux 发行版，比如 Ubuntu，都预先安装有 NumPy 并且已经是配置好了的。

如果你想要最新版的 NumPy 和 SciPy，你可以从源代码进行编译，不过最简单的方法还是用一个包管理器，比如在 Ubuntu 上用 apt-get。

2.2 Matplotlib

用一句简洁的话概括，matplotlib 是一个绘图库。如果你以前用过 MATLAB，在 matplotlib 环境里你很可能会觉得非常的舒适。在分析图像的时候，我们会用到它。不管是画图像直方图还是只是简单地查看图像本身，matplotlib 是一个非常有用的工具，你值得将它纳入到你的工具箱中。

2.2.1 全平台

Matplotlib 可以在<http://matplotlib.org/>上下载。如果你已经安装了 ScipySuperpack，那么 Matplotlib 就已经在你的计算机上安装了。你也可以通过 easy_install 或 pip 来安装它。

此外，在 matplotlib 官网上还提供了 Windows 的二进制安装文件。

2.3 OpenCV

如果说 NumPy 的主要目标是用于有效地表示大型多维数组，那么，OpenCV 的主要目标便是实时处理图像。该库从 1999 年发布以来，已随处可见，直到在 2009 年发布的第 2 版中我们才看到了它对 NumPy 的支持。OpenCV 这个库本身是用 C/C++ 写的，但在运行该安装包时它提供了 Python 的绑定。OpenCV 是我手下最喜欢的计算机视觉库，在本书中，我们会经常使用它。

OpenCV 的安装是常常会变化的。由于这个库是用 C/C++ 写的，所以在编译的时候需要特别的注意，并确保预先要安装的东西都已安装。由于 OpenCV 的最新



安装说明是经常会改变的，所以在安装 OpenCV 时要确保自己去查看 OpenCV 的网站<http://opencv.org/>。

2.3.1 Windows 和 linux

OpenCV 文档提供了在 Windows 和 Linux 用二进制版本安装 OpenCV 非常棒的教程，你可以在这里看到它的安装说明http://docs.opencv.org/doc/tutorials/introduction/table_of_content_introduction/table_of_content_introduction.html#table-of-content-introduction。

2.3.2 OSX

过去数年，在 OSX 上安装 OpenCV 是一件很痛苦的事，幸运的是现在它的安装已变得越来越简单。你需要安装 cmake 来构建并编译该 OpenCV 库，你可以从这里下载 cmake<http://www.cmake.org/cmake/help/install.html>。

安装 cmake 后，你就可以从 SourceForge 资源库检查源代码并进行编译。一般情况下，我发觉 Guilherme Defreitas 关于怎样在 OSX 上安装 OpenCV 的说明是非常棒的，并且为 OpenCV 的学习给出了一个很好的开始。

你可以在这里找到他的说明：<http://www.guidefreitas.com/installing-opencv-2-4-2-on-mac-osx-mountain-lion-with-python-support>。

2.4 Mahotas

如 OpenCV 一样，Mahotas 也依赖于 NumPy 数组。很多在 Mahotas 中实现的函数都可以在 OpenCV 中找到，不过在某些情况下，Mahotas 接口比 OpenCV 接口更容易使用，我们将用 Mahotas 作为 OpenCV 的补充。

2.4.1 全平台

在不同的平台上安装 Mahotas 是极其容易的。假设你已经安装了 NumPy 和 SciPy，那么你所有需要做的只是运行 `pip install mahotas` 或 `easy install mahotas`。

既然我们已经安装了所有我们需要的安装包，现在就让我们开始探索计算机视觉世界之旅吧！

2.4.2 跳过安装

正如我在前面提到过，安装这些安装包是费时且繁琐的。如果你想跳过上面那些安装过程而直接进入图像处理和计算机视觉世界，我已经打包好了一个预先配置好了的 Ubuntu 虚拟机，该虚拟机安装好了上面提到的所有安装包。



如果你对该虚拟机感兴趣并且想要下载它的话（它能够节省你很多时间并避免很多麻烦），你可以到这里下载<http://www.pyimagesearch.com/practicalpython-opencv/>。



第 3 章 载入、显示及保存



这本书的初衷是成为一本实用手册，用于指引怎样用 Python 和 OpenCV 来开始计算机视觉编程。本着这样一条信念，那就让我们不要浪费任何时间了。拿起你的键盘开始写些简单的代码，用于从磁盘载入一幅图像，在屏幕上显示它，并将它用不同的格式写入一个文件。在执行该代码的时候，我们的 Python 脚本应该可以在屏幕上显示我们载入的图像，正如图 3.1 所示。

首先，我们创建一个名为 `load_display_save.py` 来包含我们的代码，现在我们可以开始写代码了：

列表 3.1: `load_display_save.py`

```
1 import argparse
2 import cv2
3
4 ap = argparse.ArgumentParser()
5 ap.add_argument("-i", "--image", required = True,
6                 help = "Path to the image")
7 args = vars(ap.parse_args())
```

我们要做的第一件事是导入在这个例子中我们需要的包。我们用了 `argparse` 来解析我们的命令行参数。然后，导入 `cv2` - 这里 `cv2` 就是我们的 OpenCV 库，它包含了我们需要的图像处理函数。

上面代码第 4-7 行用于解析命令行参数。这里我们需要的唯一一个参数是 `-image`：磁盘上图像所在的路径。最后，我们解析这些参数，并在它们保存在一个字典中。

列表 3.2: `load_display_save.py`

```
1 image = cv2.imread(args["image"])
2 print "width: %d pixels" % (image.shape[1])
3 print "height: %d pixels" % (image.shape[0])
4 print "channels: %d" % (image.shape[2])
5
6 cv2.imshow("Image", image)
7 cv2.waitKey(0)
```

既然我们有了图像所在的路径，在第 8 行我们就可以用 `cv2.imread` 函数从磁盘载入该图像。`cv2.imread` 函数返回一个 NumPy 数组，用于表示一幅图像。

第 9-11 行检查图像的维数。由于图像是用 NumPy 数组来表示的，我们可以用 `shape` 属性来检查该图像的宽、高，以及颜色通道数。

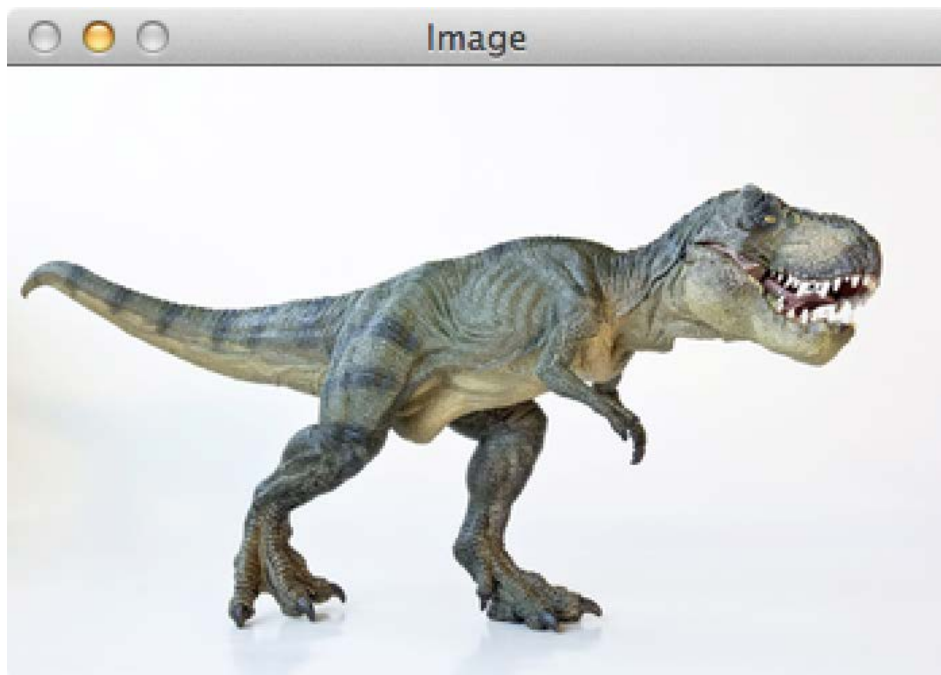


图 3.1: 载入并在屏幕上显示一幅霸王龙图像的例子

最后，第 13 行和第 14 行用于在屏幕上显示实际载入的图像。cv2.imshow 的第一个参数是一个字符串，是窗口的“名字”；第二个参数是我们在第 8 行从磁盘载入的图像。最后调用了 cv2.WaitKey 函数来暂停运行的脚本，直到我们在键盘上按下一个键。用参数“0”表示按下任意一个键就会释放暂停。

最后，我们将该幅图像以 JPG 的格式写入文件中：

列表 3.3: load_display_save.py

```
1 cv2.imwrite("newimage.jpg", image)
```

要运行上面的脚本并显示出图片，我们只需要打开终端窗口，并执行下面的命令：

列表 3.4: load_display_save.py

```
1 $ python load_display_save.py --image ../images/trex.png
```

如果不错什么问题的话，你应该可以在你的屏幕上看到图 3.1 中的霸王龙。要终止脚本的运行的话，只需要在图像窗口上单击，并按下任意键。

检查上面脚本的输出结果，你应该可以看到图片的一些基本信息。你会看到对于那张霸王龙图片，它的宽是 350 个像素，高是 228 个像素，以及 3 个颜色通道 (R、G、B 三个颜色通道)。用一个 NumPy 数组表示的话，该数组大小为 (350,228,3)。

当我们在写矩阵时，通常我们会将它们写成这种形式： $\# \text{行} \times \# \text{列}$ ——但对于 NumPy，便不是这样的了。NumPy 通常会先给出列数，然后才是行数。这点，你要牢牢记住。

最后，注意文件夹里的内容。你会看到有一个新文件在那儿：newimage.jpg。



OpenCV 会自动地为我们将 PNG 格式的图片转换为 JPG 格式图片！所以在图像格式间进行转换时我们并不需要费什么劲。

下一章，我们将探索怎样在一幅图像上获取像素值并对像素值进行操作。



第 4 章 图像基础



在这一章中，我们将审查图像的积木——像素。我们将详细的讨论什么是像素，以及像素是怎样形成一幅图像的，并且怎么在 OpenCV 里对这些像素进行操作。

每幅图像由像素集组成。像素是一幅图像的原材料，积木。它是图像的最小单元。

一般情况下，像素通常采用两种方式进行表示：灰度和彩色。在灰度图像中，每一个像素的像素值在 0 到 255 之间，“0”代表“黑色”，“1”表示“白色”。

第 5 章 特征编码



在这一章中，我们将审查图像的积木——像素。我们将详细的讨论什么是像素，以及像素是怎样形成一幅图像的，并且怎么在 OpenCV 里对这些像素进行操作。

5.0.1 Fisher Vector

每幅图像由像素集组成。像素是一幅图像的原材料，积木。它是图像的最小单元。

一般情况下，像素通常采用两种方式进行表示：灰度和彩色。在灰度图像中，每一个像素的像素值在 0 到 255 之间，“0”代表“黑色”，“1”表示“白色”。

5.0.2 VLAD

每幅图像由像素集组成。像素是一幅图像的原材料，积木。它是图像的最小单元。

一般情况下，像素通常采用两种方式进行表示：灰度和彩色。在灰度图像中，每一个像素的像素值在 0 到 255 之间，“0”代表“黑色”，“1”表示“白色”。

参考文献

