项目一　计算机视觉概述

1.1　任务 1-1　计算机视觉概述

任务描述：什么是计算机视觉

任务要求：

- 掌握计算机视觉的概念；
- 了解计算机视觉发展历史；
- 掌握计算机视觉的主要研究内容；
- 了解计算机视觉的主要应用场景。

1.1.1　计算机视觉的概念

(1)人类视觉

提到计算机视觉，我们一定能想到人类的视觉，人类的视觉有什么作用呢？

人类获取外界信息的主要方式有眼睛(看)、耳朵(听)、鼻子(闻)、口(尝)、身体(感)。眼睛获取的信息比其他方式获取的信息都要多，是人类获取外界信息的主要方式。

我们用眼睛获取信息主要做了哪些事情呢？下面看几个典型的场景：

- 走路的时候，用眼睛观察周围的信息，例如路面信息，周围的人和物，然后根据看到的这些信息来指导下一步的行动，停止、前进、转弯、低头、高抬腿等；
- 读书的时候，获取书中的文字和图片信息之后，经过大脑的处理，记忆到大脑，从而形成知识，会影响日后生活、学习等；
- 吃饭的时候，看到饭菜，马上就知道这是什么菜，从而能够想象到味道，自己是否喜欢等各个方面。

通过这些应用场景，我们能够知道人类视觉大概能够完成如下功能：

- 使用眼睛获取外界信息，可能是一个独立的画面，可能是一个连续的画面；
- 对信息加以理解，画面中包含哪些信息？这些信息都是什么？它们之间的关系是什么？正在发生什么事情？

(2)计算机视觉

计算机视觉是模拟人类的视觉，从外界获取信息，并理解这些信息，然后会根据这些信息做后续的处理。

● 从外界获取信息，最典型的设备就是各种摄像头了，电脑上的摄像头，手机上的摄像头，各种监控摄像头，这些摄像头可以不间断地获取信息，这些信息被传送到各种存储设备上。

● 理解这些信息，从这些信息中提取有价值的信息，例如教室中安装的监控摄像头能够记录哪些人进入了教室，进入教室之后做了哪些事情。再例如，用于违章监控的摄像头能够记录哪些车辆违章了。

(3)计算机视觉的定义

计算机视觉对应的英文是 Computer Vision，简称 CV。上面是直观上对计算机视觉的理解，下面看看计算机视觉的定义。

计算机视觉是使用计算机及相关设备对生物视觉的一种模拟。它的主要任务就是通过对采集的图片或视频进行处理以获得相应场景的三维信息，就像人类和许多其他类生物每天所做的那样。

计算机视觉是一门关于如何运用照相机和计算机来获取我们所需的，被拍摄对象的数据与信息的学问。形象地说，就是给计算机安装上眼睛(照相机)和大脑(算法)，让计算机能够感知环境。中国人的成语“眼见为实”和西方人常说的“One picture is worth ten thousand words”表达了视觉对人类的重要性。不难想象，具有视觉的机器的应用前景能有多么地宽广。

计算机视觉既是工程领域，也是科学领域中的一个富有挑战性重要研究方向。计算机视觉是一门综合性的学科，它已经吸引了来自各个学科的研究者参加到对它的研究之中。其中包括计算机科学和工程、信号处理、物理学、应用数学和统计学，神经生理学和认知科学等。

(4)机器视觉

有时候也称计算机视觉为机器视觉，但是它们之间有些细节的区别。计算机视觉主要是指对图像进行数据采集后提取图像的特征，一般处理的图像的数据量比较大，偏软件层。机器视觉处理的图像一般不大，采集图像数据后仅进行少量数据流的计算，偏硬件层，用于工业机器人、工业检测等。

1.1.2 了解计算机视觉发展历史

计算机视觉的发展经历了多个阶段。

(1)计算机视觉的萌芽阶段

1966 年，人工智能学家 Minsky 给自己的学生布置了一个作业，作业让学生编写一个程序，然后让计算机说明它通过摄像头看到了什么，这被认为是计算机视觉最早的任务描述。

20 世纪七八十年代，出现了现代电子计算机，计算机视觉这个技术也开始初步萌芽。人们开始试着让计算机回答出它看到的是什么，怎么实现呢？首先想到的是人类如何看东西，然后借鉴。

首先，人类能看到事物并理解，是因为人类的两只眼睛能够立体地观察事物。因此计算机要理解它所看到的图像，必须先将事物的二维的图像恢复出三维结构，这就是“三维重构”方法。

其次，人之所以能识别出一个苹果，是因为人们已经认识了苹果，有了苹果的先验知识，例如多数苹果是红色的、圆的、表面光滑的，当然了也有其他品种。如果给机器也建立一个这样的事物知识库，让机器将看到的图像与知识库里的知识匹配，这样也许就可以让机器识别乃至理解它所看到的事物，这是“先验知识库”方法。

这个阶段的典型应用主要是光学字符识别、显微/航空图片识别、工件识别等等。

(2)广泛应用于工业领域

20 世纪 90 年代，计算机视觉技术得到了更大的发展，开始应用于各种工业场景。一个原因是计算机 CPU、DSP 等图像处理相关的硬件技术有了飞速进步；另一个原因是人们也开始尝试不同的思路，例如使用统计方法，引入局部特征描述符等。

(3)机器学习方法阶段

进入 21 世纪，互联网以及移动互联网的兴起和数码相机的普及带来了海量数据，另外机器学习方法也得到了广泛应用，计算机视觉技术迅速发展。机器学习方法能够自动从海量数据中总结归纳对象的特征，然后进行判断和识别，所以以往那些基于规则和知识库的处理方式逐步被替代了。

这个阶段出现了很多典型应用，包括相机中使用的人脸检测，安防领域的人脸识别、车牌识别等等。这些应用中，对象特征相对比较明显。

(4)深度学习方法阶段

2010 年以后，深度学习在图像处理方面表现出了强大的优势，计算机视觉技术得到了爆发式的增长和产业化。通过深度神经网络，各种计算机视觉相关的识别精度都得到了大幅提升。

在计算机视觉竞赛(ILSVR)上，2010 年和 2011 年时，千类物体识别 Top5 错误率在分别为 28.2%和 25.8%，2012 年引入深度学习，后续 4 年分别降为 16.4%、11.7%、6.7%、3.7%，取得了显著突破。

1.1.3　计算机视觉的主要研究内容

图像理解是计算机视觉的终极目标。视频通常是由一系列的图像组成的，所以下面提到的图像既表示图像也表示包含图像的视频。

(1)图像的表示与存储

从各种视觉设备获取图像或者视频(图像序列)之后，需要使用特定的计算机符号来表示这些图片，并存储这些图片或者视频。图像的表示方式和图像存储方式与很多因素有关：

- 与获取图像的设备有关，不同设备的精度不同，存储和处理能力也不同，表示方式和存储方式就会有差别；
- 与传输方式相关，图像和视频可能比较大，要在网络上传输，可能需要用适合的方式表示图像，包括很多压缩方式；
- 与计算机视觉的任务相关，想用这些图像和视频做什么，也会对这些图像和视频的表示方式有不同的要求。

(2)图像处理

获取的原始图像质量可能存在问题，另外不同的计算机视觉任务需要使用不同类型的

图像特征以及表示方式，对图像有不同的要求，需要对图像进行各种加工处理。

(3)图像分割

图像中可能包含了多个对象，有些对象不是我们关心的，对象之间可能还存在着各种关系，使用图像分割就是根据图像的特征把图像中被关注的部分分割出来，为后续的视觉任务做准备。

(4)图像分类

在图像中得到各个组成部分之后，要对这些组成部分进行判断，判断对象属于哪个类别。

(5)目标检测

根据给定的某个目标特征，在图像或者视频中找到和它对应的目标，并标记出目标的位置。

(6)目标识别

识别图像中的目标是什么含义，例如图像中包含什么车牌号、联系方式和交通标志等。

(7)目标跟踪

根据最初给定的一个或者多个目标，利用这些目标的特征，能够在视频的后续各帧中继续找到这些目标，并跟踪这些目标，以及确定这些目标的位置。

1.1.4 计算机视觉的应用场景

近些年计算机视觉技术的应用场景也快速扩展，典型的一些应用场景如下。

(1)比较成熟的安防领域，家庭场景下、商场环境下、银行领域、交通领域等。大家看到的最多的可能就是在交通领域的应用，在十字路口监控车辆违章，例如监控闯红灯、不礼让行人、不按规定线路行驶、行人违章，还有监控道路流量和拥堵情况等。

(2)在金融领域的人脸识别身份验证。在银行办理各种业务的时候需要使用人脸识别，在手机上支付的时候可以使用人脸之别，在手机上办理各种金融相关业务的时候可以使用人脸识别，有时候还需要辅助做一些眼睛或者头部的动作。

(3)电商领域的商品拍照搜索。电商平台中包含大量的商品，根据名字和基本信息搜索比较困难的时候，如果用户有商品的照片，这种情况下就可以根据照片进行搜索。系统会根据用户提供的照片与系统中存储的商品的照片进行对比，从而找到用户感兴趣的商品。

(4)医疗领域的智能影像诊断。在医院中做各种检查会形成大量的图像，医生可以根据自己的专业知识和经验从这些图像中发现特定的疾病特征，从而做出判断，随着图像处理的技术进步，可以让计算机根据相关知识从这些图像中发现疾病特征。

(5)机器人/无人车/无人机上作为视觉输入系统等。现在有大量的企业投入无人驾驶汽车的研究中，百度在 2015 年下半年推出了无人驾驶汽车。百度无人驾驶汽车可自动识别交通指示牌和行车信息，具备雷达、相机、全球卫星导航等电子设施，并安装同步传感器。车主只要向导航系统输入目的地，汽车即可自动行驶，前往目的地。在行驶过程中，汽车会通过传感设备上传路况信息，在大量数据基础上进行实时定位分析，从而判断行驶方向和速度。

(6)照片自动分类。主要使用图像识别和图像分类技术，例如在华为手机中你选择查看

某张照片的时候，它会把同一个人的照片全部找出来。如图 1-1 所示。

(7)图像描述生成。根据图像内容识别出图像中的对象以及对象之间的关系，然后对识别的结果加以理解，然后给出关于图像的描述。如图 1-2 所示。

图 1-1　图片自动分类

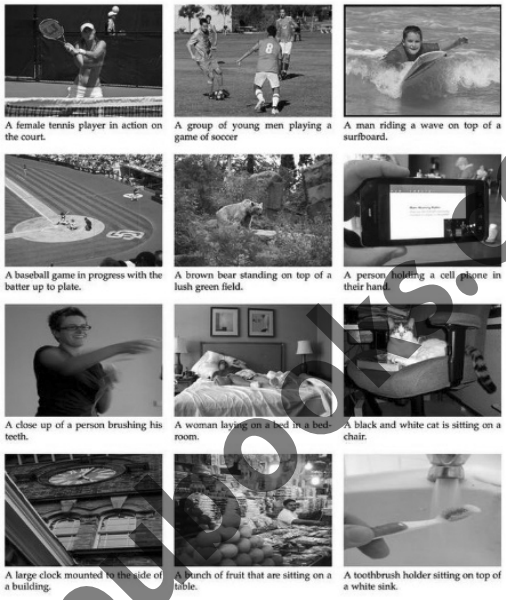

图 1-2　图像描述生成

1.2　任务 1-2　在 Windows 上安装 Python

任务描述：本教材介绍计算机视觉相关技术的时候使用 Python 介绍，所以需要安装 Python，本节的任务就是安装 Python 环境。

任务要求：

- 了解计算机视觉的主要工具；
- Python 与计算机视觉的关系；
- Anaconda 的安装。

1.2.1　计算机视觉的主要工具

计算机视觉相关的工具很多，下面介绍几个常用的。

(1)OpenCV

OpenCV，计算机视觉领域应用最出名最广泛的开源工具包。它基于 C/C++语言，支持 Linux/Windows/MacOS/Android/iOS 等大部分操作系统，并提供了 Python、MATLAB 和 Java 等语言的接口。因为其接口丰富，性能优秀和商业使用许可友好，在学术界和业界都非常受欢迎。

(2)MATLAB Computer Vision System Toolbox

MATLAB 一直是学术界所钟爱的计算工具,MATLAB 也提供了计算机视觉领域的支持。它的视觉工具包也沿袭了上手简单和可视化方便的风格,许多计算机视觉研究人员和工程师都选择使用。

(3)SimpleCV

SimpleCV 是基于 Python 的一个视觉库,提供简单易用的接口,底层基于 OpenCV、PIL 等其他的计算机视觉和图像处理库。

(4)深度学习工具

TensorFlow 是一个基于数据流编程的符号数学系统,广泛应用于各类机器学习算法的实现。TensorFlow 2.0 为图片、语音识别、对象检测、推荐和强化学习提供了预先准备的模型。

Keras 是一个深度学习的 Python 库,它综合了不同深度学习库的元素,例如 TensorFlow、Theano 和 CNTK。Keras 在 TensorFlow 之上运行,优于 Scikit-learn 和 PyTorch 等。

百度的 PaddlePaddle 也是一个深度学习框架,在国内也得到了广泛应用。

(5)其他

还有一些其他的计算机视觉工具。例如 CCV,一个基于 C 语言实现的带缓存的计算机视觉库,使用简洁。VLFeat 是一个老牌的计算机视觉库,基于 C 语言实现,并提供 MATLAB 的接口。VXL 是一个基于 C++语言实现的计算机视觉库。

1.2.2 Python 与计算机视觉

Python 中提供了大量的计算机视觉相关的开发工具,其中的部分工具介绍如下。

(1)NumPy

NumPy 是 Python 中的核心库之一,并为数组提供支持。图像本质上是包含像素值的标准 NumPy 数组。因此,可以通过 NumPy 数组操作修改图像。可以使用 Skimage 加载图像,也可以使用 Matplotlib 显示图像。

(2)PIL 库/Pillow 库

PIL(Python Imaging Library)是 Python 中最常用的图像处理库。Image 类是 PIL 库中一个主要类,通过这个类来创建图像实例,可以直接读取图像文件,通过抓取方法得到图像。得到图像之后,可以对图像进行裁剪、粘贴和各种加工操作,包括各种几何变换,例如对图像做色彩空间的变换,能对图像进行过滤、增强等。

Pillow 是 Python3 最常用的图像处理库,在 Python2 使用 PIL 库,两者方法类似,只是类的引用不同。

(3)OpenCV-Python

OpenCV-Python 是 OpenCV 的 Python API。后台由用 C/C++编写的代码组成,OpenCV-Python 不仅速度快,也很容易编程和部署。

OpenCV 提供了很多基本图像操作,包括图像读取、图像变换操作、基本的图形处理、图像特征的提取等,另外还提供了大量优秀算法,本教材中大部分计算机视觉相关的方法都来

自 OpenCV。

1.2.3　Anaconda 的安装

Anaconda 是一个 Python 的发行版，包括了 Python 和很多常见的软件库，和一个包管理器 conda，安装了 Anaconda 就不用单独再安装 Python 了。

(1)下载地址

官网的下载地址：https://www.anaconda.com/products/individual，如图 1-3 所示。下载速度慢。

也可以使用其他的国内镜像下载，例如清华的镜像：

https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/。

(2)安装

步骤 1：点击选择现下载的安装程序，进行安装，如图 1-4 所示欢迎界面。

图 1-3　下载界面

图 1-4　安装欢迎界面

步骤 2：点击 Next 进入下一步，如图 1-5 所示的是否同意协议界面。

步骤 3：在询问协议是否同意的界面，点击 I Agree 同意协议，然后进入图 1-6 所示界面，在这个界面选择安装对当前用户有效，还是对所有用户有效。在这个界面需要注意选择 All Users，也就是所有用户可用。

步骤 4：点击进入下一个界面，如图 1-7 所示。在该界面选择安装路，注意路径中不要使用空格和中文，否则可能出错。

步骤 5：选择路径之后点击 Next 进入如图 1-8 所示的下一个界面。在该界面选择“是否把 Anaconda 加入系统的路径”，这里不选择，另外需要选择把 Aanconda 作为系统 Python 环境这个复选框，都采用默认就可以了。

步骤 6：之后点击 Install 就进入安装画面了。

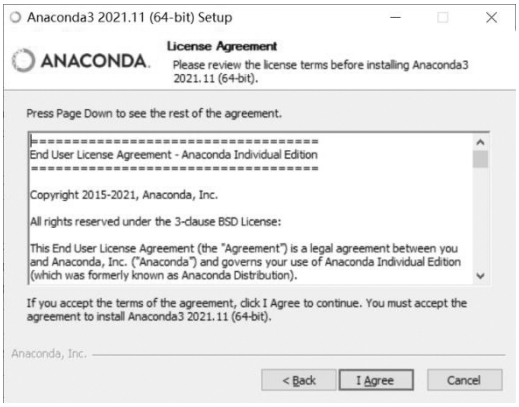

图 1-5　权力声明

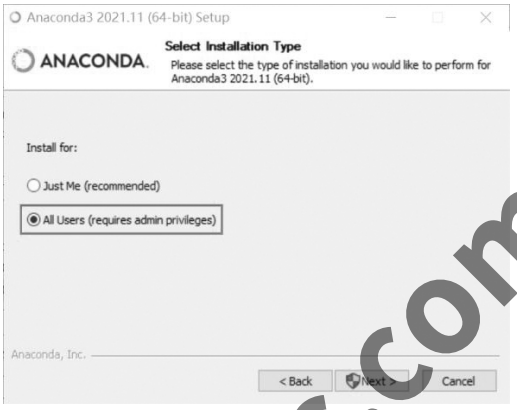

图 1-6　选择用户

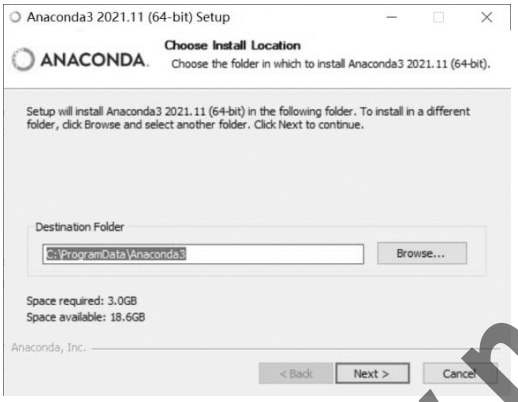

图 1-7　安装路径选择界面

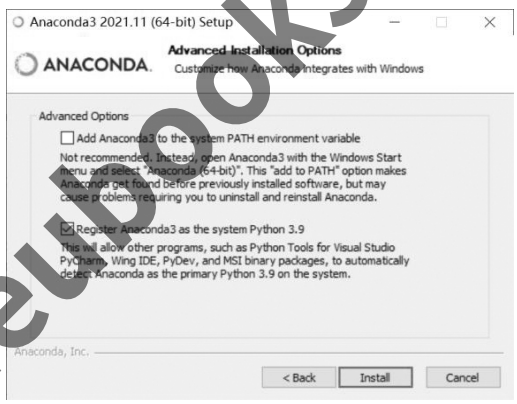

图 1-8　设置界面

步骤 7:安装完之后会提示是否安装 VSCode,选择 Skip,不安装。

步骤 8:然后进入安装结束的界面,两个复选框都不选,点击 Finish 完成安装。

(3)配置环境变量

此电脑右键点击属性,选择“高级”系统设置,选择“环境变量”,选择“path”,选择“编辑”,选择“新建”,在 path 中添加如下环境变量:

E:\Anaconda(Python 需要)

E:\Anaconda\Scripts(conda 自带脚本)

E:\Anaconda\Library\mingw-w64\bin(Python 中使用 C 的时候)

E:\Anaconda\Library\usr\bin

E:\Anaconda\Library\bin(jupyter notebook 动态库)

(4)测试环境

测试 Python 环境,输入 Python,如果能进入 Python 的命令行模式,说明 Python 的环境变量起作用了,如图 1-9 所示。

```
(base) C:\Users\lixucheng>python
Python 3.7.4 (default, Aug  9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

图 1-9　测试 Python

测试 Anaconda 环境，输入 conda，如果能够提示 Anaconda 的用法提示信息，则说明 Anaconda 环境变量起作用了，如图 1-10 所示。

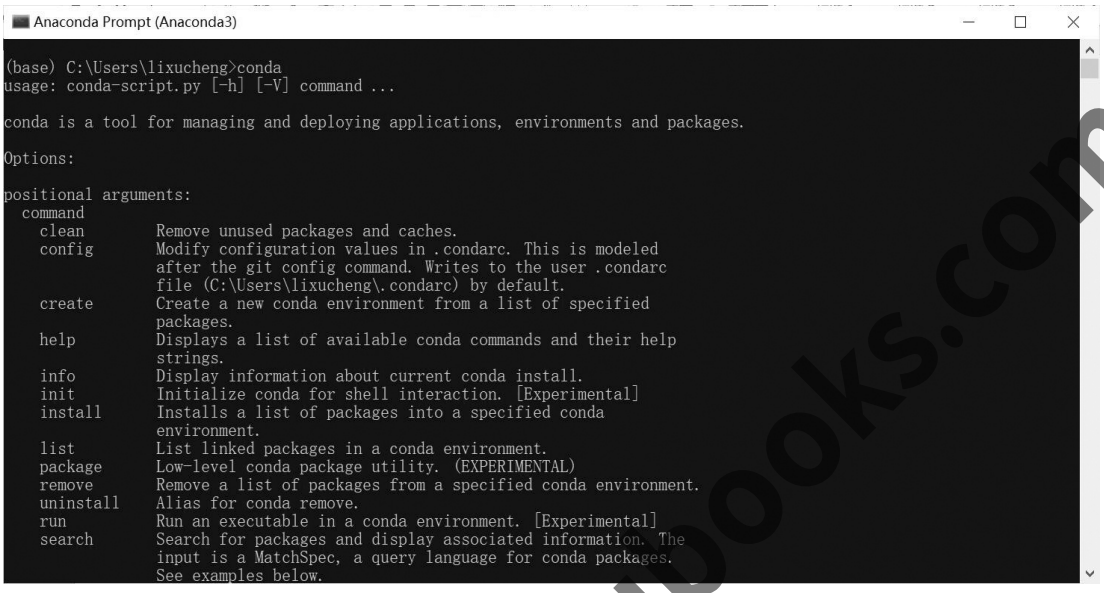

图 1-10　Anaconda 环境测试

1.3　任务 1-3　显示一张彩色照片

任务描述：从电脑中读取一张照片，然后把它显示在窗口中。

任务要求：

- 掌握如何读取图片；
- 掌握如何表示图片；
- 掌握如何显示图片；
- 掌握如何关闭窗口。

1.3.1　引入 cv2

后面介绍的大部分功能都会使用 OpenCV，cv2 就是 Python 中提供的 OpenCV 库，所以首选需要引入 cv2，下面是引入 cv2 的代码：

```
import cv2
```

1.3.2　读取文件

使用 cv2.imread(文件名，属性)读取图像文件。

第一个参数是文件名。

第二个参数表示打开方式。cv2.IMREAD_COLOR 表示读入彩色图像，是默认方式，OpenCV 读取的彩色图像为 BGR 模式，不是 RGB 模式。cv2.IMREAD_GRAYSCALE 表

示读灰度图像。返回 Image 对象，后续可以使用返回的对象操作图像。

例如读取 img 文件夹下的一个文件，img 文件夹与当前项目在相同文件夹：

```
img = cv.imread('img/Lenna.png')
```

1.3.3 图像的表示

通过 imread 方法返回的彩色图像实际上是一个三维的数组，使用 print(img.shape)打印类型，会输出类似下面的信息：(100,100,3)，表示图像的大小是 100×100，宽度和高度都是 100 个像素，整个图像是 10 000 个像素，每个像素包含 3 个值，分别表示蓝色(B)、绿色(G)和红色(R)，它们的取值范围都是 0～255。

可以使用下面的代码查看某个像素的 3 个值：

```
print(img[50][50])
```

也可以查看具体某个通道的值，下面的代码分别显示蓝色、绿色和红色部分的值：

```
print(img[50][50][0])    蓝色部分
print(img[50][50][1])    绿色部分
print(img[50][50][2])    红色部分
```

1.3.4 显示图像

cv2.imshow(窗口名，图像)，第一个参数表示显示图像的窗口的名字，第二个参数表示要显示的图像。

要显示上面的图像，可以使用：

```
title='lena'
cv2.imshow(title, img)
```

1.3.5 控制窗口显示的时间

可以使用 cv2.waitKey(parameter)来接收用户的键盘输入信息。

参数表示窗口显示的时间，单位是毫秒，如果要一直显示，可以设置 parameter = NONE 或者 0 来表示。

在窗口显示的过程中，如果用户点击了键盘，窗口就关闭了。

下面的代码显示 3 秒钟：

```
cv2.waitKey(3000)
```

下面的代码一直显示图像：

```
cv2.waitKey(0)
```

1.3.6 关闭窗口

调用 destoryAllWindows()方法关闭所有窗口。

```
cv2.destroyAllWindows()
```

1.3.7 完整代码

读取图片并显示图片的完整代码如下：

```
import cv2
img = cv2.imread('lena.jpg')
title='lena'
cv2.imshow(title,img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

1.4　任务 1-4　使用 Python 实现视频录制

任务描述：使用 Python 语言，调用电脑上自带的摄像头，录制视频信息，然后把视频保存到电脑上。

任务要求：

- 掌握视频与图像的关系；
- 掌握视频的常见格式；
- 掌握如何打开摄像头；
- 掌握如何读取摄像头图像；
- 掌握如何显示视频；
- 掌握如何释放摄像头；
- 掌握如何保存视频。

1.4.1　视频与图像的关系

视频是连续的图像，包含多幅图像，并包含图像的运动信息。由于人眼识别的频率有限，所以在单位时间内看到的图像数量超过一定的数目，给人的感觉就是视频了。这个频率一般是每秒 25 张，通常称为 25 帧/秒。组成视频的每张图像也称为一帧。

1.4.2　视频的格式

(1)MPEG

MPEG 是运动图像专家组格式，是 Moving Picture Experts Group 的缩写。常见的 VCD、SVCD、DVD 就是这个格式。MPEG 是运动图像压缩算法的国际标准，是有损压缩，能够降低视频中冗余信息。MPEG 压缩保留了相邻的两幅图像中的绝大多数相同部分，而把冗余部分去除，从而进行压缩。MPEG 压缩标准主要有 MPEG-1、MPEG-2、MPEG-4、MPEG-7 与 MPEG-21。

(2)AVI

AVI 是音频视频交错，是 Audio Video Interleaved 的英文缩写，将视频和音频封装在一个文件里，且允许音频和视频同步播放，由微软 1992 年推出。优点是图像质量好，能跨多个平台；缺点是体积大，压缩标准不统一。

(3)ASF

ASF是微软发展出来的一种可以直接在网上观看的视频文件压缩格式,是Advanced Streaming Format的缩写。

(4)MOV

MOV是苹果公司的一种音频、视频文件格式,即QuickTime影片格式,用于存储常用的数字媒体类型,动画将保存为.mov文件。

(5)WMV格式

WMV是微软推出的一种流媒体格式,是Windows Media Video的缩写,是ASF格式的升级延伸。WMV格式体积非常小,适合在网上使用。

(6)3GP格式

3GP是"第三代合作伙伴项目"制定的一种多媒体标准。即一种3G网络流媒体的视频编码格式,是目前手机中最为常见的一种视频格式。

(7)RM格式与RMVB格式

RM是Real Networks公司的音频视频压缩规范,是Real Media的缩写。用户可以使用RealPlayer或RealOne Player对RM音频/视频资源进行实况转播,并且Real Media可以根据网络传输速率制定出对应的压缩比率,从而能够在低速率的网络上进行影像数据实时传送和播放。RMVB格式是RM视频格式的升级,先进之处在于打破了平均压缩采样的方式,在保证平均压缩比的基础上合理利用比特率资源。

(8)FLV/F4V

FLV是Flash Video的简称,是一种视频流媒体格式。它形成的文件小、加载速度快,使在网络中观看视频成为可能。F4V是Adobe公司推出的,支持H.264的高清流媒体格式,码率可达50 Mbps。F4V更小更清晰,更利于网络传播,并逐渐取代FLV,被大多数播放器所兼容。

1.4.3 打开摄像头

VideoCapture(int)是cv2中用于获取摄像头的函数,参数是整数,表示要打开的摄像头。下面的代码用于打开摄像头:

```
cap = cv2.VideoCapture(0)
```

该函数的参数为0,表示打开笔记本自带的摄像头。

1.4.4 读取摄像头图像

VideoCapture的read方法用于读取摄像头图像:

```
ret, frame = cap.read()
```

ret表示是否读取到图片,如果是true,表示读取到视频帧,如果是false,表示没有读取到视频帧,frame表示读取的视频帧数据。

有时候初始化摄像头可能出错了,也可以使用isOpened()方法来检查是否成功初始化了摄像头。返回值为true,表示初始化成功。如果没有成功初始化,可以使用open()方法打开摄像头。

1.4.5　显示视频

视频中的每一帧就是一幅图像，所以显示视频的过程是一帧一帧来显示图像，显示图像仍然使用 imshow 方法，第一个参数表示显示图像的窗口的标题，第二个参数表示要显示的图像，例如下面的代码：

```
cv2.imshow("cap2", frame)
```

要显示视频，就是不断地获取视频帧，并显现在界面上，可以使用循环来实现，下面的代码实现了视频的显示：

```
while 1:
    ret, frame = cap.read()
    cv2.imshow("cap2", frame)
    if cv2.waitKey(100) & 0xff == ord('q'):
        break
```

这里使用死循环来不停地展示图片，实际上就是视频了。如果用户在界面输入的信息，并且输入的信息是“q”的时候使用“break”退出循环。

1.4.6　释放摄像头

与使用其他资源一样，在使用之后需要关闭摄像头，使用 release 方法释放摄像头资源。

```
cap.release()
```

关闭摄像头之后，关闭窗口，使用 destroyAllWindows 方法关闭窗口：

```
cv2.destroyAllWindows()
```

1.4.7　完整的代码

下面的代码实现了获取摄像头的视频并显示在窗口中的功能，当用户在键盘输入“q”的时候，关闭摄像头并关闭窗口，完整的代码如下：

```
#coding=utf-8
import cv2
cap = cv2.VideoCapture(0)
while 1:
    ret, frame = cap.read()
    cv2.imshow("cap2", frame)
    if cv2.waitKey(100) & 0xff == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

1.4.8　保存视频

保存视频是在获取视频之后把视频保存到文件中。主要的过程如下：

(1)引入 cv2

```
import cv2
```

(2)打开摄像头

VideoCapture(int)是 cv2 中用于获取摄像头的函数：

```
cap = cv2.VideoCapture(0)
```

该函数的参数为 0 表示打开笔记本的摄像头。

(3)得到摄像头图像的大小

cap 的 get 方法用于获取各种参数，要想获取视频图像的宽度和高度可以使用 cv2.CAP_PROP_FRAME_WIDTH 和 cv2.CAP_PROP_FRAME_HEIGHT，下面的代码用于获取摄像头的大小：

```
sz = (int(cap.get(cv2.CAP_PROP_FRAME_WIDTH)),
      int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT)))
```

(4)设置视频的编解码器

使用 VideoWriter_fourcc 设置视频的编解码器，典型的参数有：

cv2.VideoWriter_fourcc('I','4','2','0')，表示 YUV 编码类型，文件名后缀为.avi；

cv2.VideoWriter_fourcc('P','I','M','I')，表示 MPEG-1 编码类型，文件名后缀为.avi；

cv2.VideoWriter_fourcc('X','V','I','D')，表示 MPEG-4 编码类型，文件名后缀为.avi；

cv2.VideoWriter_fourcc('T','H','E','O')，表示 Ogg Vorbis，文件名后缀为.ogv；

cv2.VideoWriter_fourcc('F','L','V','1')，表示 Flash 视频，文件名后缀为.flv；

cv2.VideoWriter_fourcc(*'mpeg')，表示 mpeg 格式，文件后缀名为.mp4。

(5)得到视频输出流

使用 cv2.VideoWriter()函数获取视频输出流：

```
vout = cv2.VideoWriter()
```

(6)设置帧率

```
fps = 20
```

(7)创建输出文件

```
vout.open('d:/output.mp4',fourcc,fps,sz,True)
```

第一个参数 d:/output.mp4 表示文件存储的位置，第二个参数 fourcc 表示视频的编解码器类型，第三个参数 fps 表示帧率，第四个参数 sz 表示视频图像的大小，最后一个参数 True 表示彩色图片。

(8)把图像写入视频文件

```
vout.write(frame)
```

(9)循环采集视频并输出

```
cnt = 0
whilecnt<2000:
cnt += 1
    print(cnt)
    _, frame = cap.read()
    cv2.putText(frame, str(cnt), (10, 20), cv2.FONT_HERSHEY_PLAIN, 1, (0,255,0), 1, cv2.LINE_AA)
    vout.write(frame)
```

(10)释放摄像头

使用 release 方法释放摄像头：

```
cap.release()
```

(11)关闭窗口

使用 destroyAllWindows 方法关闭窗口：

```
cv2.destroyAllWindows()
```

(12)完成代码

```
#vedioTest.py
#coding=utf-8

# 引入库
import cv2

# 打开摄像头
cap = cv2.VideoCapture(0)

#得到图像大小
sz = (int(cap.get(cv2.CAP_PROP_FRAME_WIDTH)),
      int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT)))

#帧率
fps = 20

# 设置编码方式
fourcc = cv2.VideoWriter_fourcc('m', 'p', '4', 'v')
#fourcc = cv2.VideoWriter_fourcc('m', 'p', 'e', 'g')
#fourcc = cv2.VideoWriter_fourcc(*'mpeg')

# 得到视频输出流
vout = cv2.VideoWriter()
# 创建文件
vout.open('d:/output.mp4',fourcc,fps,sz,True)

#Cnt 用于计数
cnt = 0
# whilecnt<2000 表示循环 2000 次,也就是采集 2000 帧。
whilecnt<2000:
    cnt += 1
    # 输出计数器
    print(cnt)
    # cap.read()读取一帧数据,
```

```
        _, frame = cap.read()
        # cv2.putText在视频帧上输出文字,
        cv2.putText(frame, str(cnt), (10, 20), cv2.FONT_HERSHEY_PLAIN, 1, (0,255,0), 1, cv2.LINE_AA)
        #vout.write(frame)输出视频帧。
        vout.write(frame)

    # 关闭输出流
    vout.release()
    # 关闭摄像头
    cap.release()
```

小结

本章首先介绍了计算机视觉相关的概念,让读者知道计算机视觉的主要研究内容、应用场景。之后介绍了如何在 Windows 中安装计算机视觉的开发环境 Python,这里介绍了 Anaconda 环境的安装。最后分别通过两个任务介绍了图像的读取和显示以及视频的录制与保存。通过这些内容读者应该对计算机视觉有了初步的认识。

作业

1.列举 3 个日常生活中所见的计算机视觉的应用案例。

2.从网上查阅资料,看看有哪些成功的计算机视觉相关的公司,每个公司的主要业务有哪些?

参考公司如下:

商汤科技:安防监控、人脸识别;

旷视科技:人脸检测、人脸识别、人脸分析;

依图科技:车辆、人脸识别;

云从科技:人脸识别;

百度:图像检索、智能驾驶;

华为:智能驾驶;

海康威视:安防监控;

大华:安防监控。

3.从招聘相关的网站上查询与计算机视觉相关的就业岗位有哪些?对知识有什么要求?待遇如何?

4.在 Python 中使用 OpenCV 的时候,引入包的语句是(import cv2)。

5. 在 OpenCV 中读取图像的函数是(imread)。

6. 在 OpenCV 中显示图像的函数是(imshow)。

7. 在 OpenCV 中关闭图像窗口的函数是(destroyAllWindows)。

8. 在 OpenCV 中获取摄像头的函数是(VideoCapture)。

9. 在 OpenCV 中通过摄像头读取视频帧的函数是(read)。

项目二　计算机视觉基础知识

2.1　任务 2-1　图像的数字表示

任务描述：掌握图像是如何表示的。

任务要求：

- 掌握图像分辨率的概念；
- 掌握图像类型转换；
- 掌握图像的坐标系。

2.1.1　图像分辨率

图像的大小用分辨率来表示，如图 2-1 显示的是查看某个图像时候显示的信息，说明该图像的分辨率是 256×256，宽度是 256 像素，高度是 256 像素。

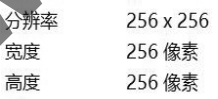

图 2-1　图像分辨率

另外，在使用电脑的时候也会设置屏幕的分辨率，分辨率的高低会影响显示内容的多少，图 2-2 所示画面是能够选择的屏幕分辨率。

购买的相机或者手机上的摄像头都有最大支持的分辨率，通常分辨率越高，采集的照片越清晰，当然通常价格也要更高。图 2-3 显示的是某款手机的两个摄像头的分辨率。

图 2-2　某显示器分辨率

图 2-3　某手机摄像头分辨率

图像的分辨率越高，需要的存储空间越大。

2.1.2　图像在内存中的表示

(1)显示图像

在项目一介绍了如何使用 OpenCV 代码显示图像，使用的主要代码如下：

```
import cv2
img = cv2.imread("lena.jpg")
cv2.imshow("lena",img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

图像的显示效果如图 2-4 所示。

图 2-4　图像显示效果

(2)查看像素值

接下来研究一下图像究竟是如何表示的。首先在代码中加入 print(img.shape)可以输出 img 对象的存储结果，输出结果为(256,256,3)。

该信息表示图像有 256×256 个像素组成，每个像素由 3 个值来表示。下面是左上角的 4 个像素的值：

[116 134 227]　[116 134 227]
[115 133 226]　[115 133 226]

(3)修改像素值

这些值是什么含义呢？把其中一个值调整为 255，把另外两个值调整为 0，来看效果，下面是修改的代码和显示效果。

首先修改第一个通道的值，使用下面的代码：

```
import cv2
img = cv2.imread("lena.jpg")

for i in range(100):
    for j in range ( 100):
        img[i][j][0]=255
        img[i][j][1]=0
        img[i][j][2]=0

cv2.imshow("lena",img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

代码中通过两层循环把前 100 行，前 100 列的第一个通道的值分别改成 255，其他两个值改为 0。

运行效果如图 2-5 所示。发现图像的左上角变成了蓝色，因为修改了前 100 行中前 100 列，这些数据对应图像的左上角。变成蓝色是因为把第一个通道的值设置为 255，另外两个通道的值设置为 0，3 个通道的值分别为蓝色分量、绿色分量和红色分量，采用的是 BGR 模

式，这里的颜色修改相当于设置颜色为(255,0,0)，表示蓝色。各个通道的最小值是 0，最大值是 255。

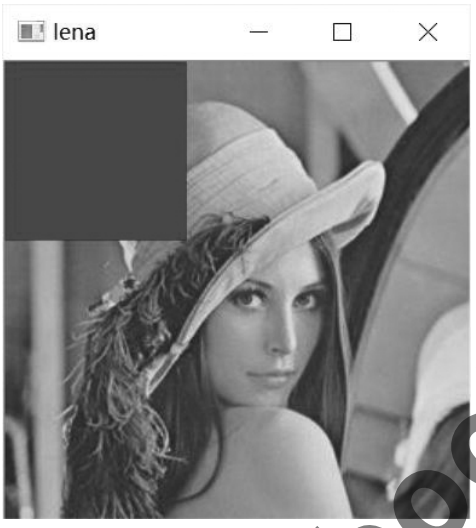

图 2-5　修改第一个颜色通道

同样使用类似的代码修改第二个通道的值为 255，其他通道值为 0，相当于修改为绿色，修改的代码如下：

```
for i in range(100):
    for j in range( 100):
        img[i][j][0]=0
        img[i][j][1]=255
        img[i][j][2]=0
```

运行效果如图 2-6 所示，可以看出把左上角的颜色改为绿色了。

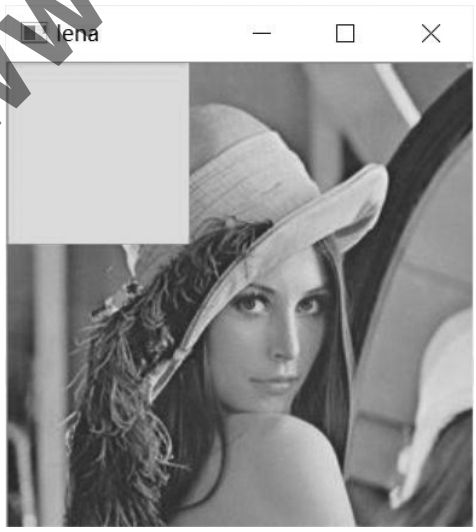

图 2-6　修改第二个颜色通道

同样使用类似的代码修改第三个通道的值为 255，其他通道值为 0，相当于修改为红色，修改的代码如下：