

# 学习任务一 图像处理应用实践



## 学习目标

1. 根据工作任务书，填写工作联系单。
2. 能正确完成基本的计算机操作及系统操作。
3. 能根据相关技术要求及规范，完成 python 图像处理环境的搭建，填写步骤一览表。
4. 能根据任务活动的具体要求，在规定时间内完成 OpenCV 的图像简单处理、猫脸检测应用、程序员桌面边缘检测、找出漫展中的角色扮演者及交付验收等操作，通过项目检验确保图像处理模块的正常运作。
5. 能分析工作中存在问题的原因，并提出解决方案。
6. 能对学习与工作中出现的问题进行总结反思，并与他人开展良好合作，进行有效沟通。



## 建议课时

60 学时。



## 工作情景描述

学院校企合作单位 XX 公司准备投入人工智能产线，需要对图像处理模块进行建设和配置，同时还需要实施多个测试项目，确保图像处理模块的正常运作，要求我院人工智能技术应用专业教师组织同学共同完成。教师接受该测试订单，带领同学共同完成该任务，该任务要求 3 周内完成。

技术员(学生)接到主管(教师)下达的调试任务书后，认真阅读任务书，了解客户需求；收集相关后收集相关资料，熟悉 python 图像处理环境的搭建的相关步骤，制定 python 图像处理环境的搭建方案，并对方案进行可行性分析；而后按照方案，完成 OpenCV 的图像简单处理、猫脸检测应用、程序员桌面边缘检测、找出漫展中的角色扮演者等任务，并进行测试，出具验收报告；按照任务要求向主管(教师)交付验收并进行总结。



## 工作流程与活动

1. 明确任务（6 学时）
2. python 图像处理环境搭建（10 学时）
3. OpenCV 的图像简单处理（10 学时）
4. 猫脸检测应用（10 学时）
5. 程序员桌面边缘检测（9 学时）
6. 找出漫展中的角色扮演者（9 学时）
7. 总结评价（6 学时）

## 学习活动 2 python 图像处理环境搭建



### 学习目标

1. 能口述并填写常见的 Python 常用图像处理库一览表。
2. 能口述并填写 opencv-python 库的安装步骤一览表。
3. 能按要求完成 opencv-python 库的安装并进行基本代码测试，填写测试单。
4. 能按要求完成 python 图像处理第三方库的安装并填写 python 库一览表。
5. 能查阅 python 第三方库安装的相关资料，对安装过程中可能出现的问题，进行讨论分析并解决，填写安装问题一览表。



### 建议课时

10 学时。



### 学习过程

#### 一、OpenCV 的安装

OpenCV 是一个强大的图像处理和计算机视觉库，实现了很多实用算法。可完成图像及视频的基础处理，如对比度调节、图像平滑、图像模糊等，是目前应用较为广泛的图像处理算法之一。OpenCV 与很多编程环境都具备兼容性，其中在 Python 中其对应的库为 opencv-python。通过安装 opencv-python 库，可在 python 中直接调用 OpenCV 的相关算法，完成相应的图像处理操作。

##### （一）常用的图像处理库

Python 具有大部分面向对象语言的特征，可完全进行面向对象编程，具有丰富而且强大的第三方模块/框架。图像处理是 python 应用中的一个比较主流的应用方向，请查阅相关资料，熟悉在使用 python 开发人工智能应用中较多使用到图像处理模块都有哪些，并填写表 1-2-1 常见的 Python 常用图像处理库一览表。

表 1-2-1 常见的 Python 常用图像处理库一览表

库名称	库简述与特点

库名称	库简述与特点

## (二) opencv-python 库的安装步骤

Python 库的安装与常规的程序方式有很多种,其中最为常见的就是采用在线安装的方式,通过连接 python 的库服务器,输入相应的安装指令来完成 python 库的安装。请查阅相关的资料,并在 python 编程环境中操作,完成表 1-2-2opencv-python 库的安装

步骤一览表的补充填写。

表 1-2-2 opencv-python 库的安装步骤一览表

步骤名称	操作截图

### (三) opencv-python 代码测试

完成 opencv-python 库的安装以后，为了保证安装无误，需要进行 OpenCV 的代码测试，确保图像处理环境能正常运行。请按照表格 1-2-3 图像处理代码测试单中要求的相应测试代码，在 python 终端中运行测试代码，将测试结果截图保存填写。

表 1-2-3 图像处理代码测试单

测试代码	运行结果截图	是否正常
<pre>#导入cv模块 import cv2 #显示cv2的版本号 cv2.__version__</pre>		
<pre>#导入cv模块 import cv2 as cv #读取图像，支持 bmp、jpg、png、tiff 等常用格式 img = cv.imread("D:\python\test.jpg") #创建窗口并显示图像 cv.namedWindow("Image") cv.imshow("Image",img) cv.waitKey(0) #释放窗口 cv2.destroyAllWindows()</pre>		

## 二、python 图像处理第三方库的安装

在 python 中，除了 opencv-python 库以外，需要搭配其他图像处理相关的第三方库使用，如 Pillow、SimpleCV、Mahotas、.pgmagick 等，同时还需要搭配其他库，如 numpy、Matplotlib 等，才能发挥 python 强大的图片处理能力。通过安装更多的图像处理第三方库，与 OpenCV 搭配使用，完成较为复杂的图片处理任务。

### (一) Python 图像处理第三方库的安装

根据前面学习的内容，查阅相关的资料，熟悉目前可用于 Python 图像处理的常用的第三方库及配套的其他库，并完成相应库的安装，完成后填写表 1-2-4 python 库一览

表。

表 1-2-4 python 库一览表

已安装库的名称列表	查看已安装库的操作截图

## (二) Python 第三方库安装问题总结

在 Python 第三方库安装的过程中，根据所遇到的实际问题，进行截图保存，并总结解决的方法，进行汇总后，完成表 1-2-5 python 库安装问题一览表的填写。

表 1-2-5 python 库一览表

问题截图（文字描述）	解决方法