

# 感知实验1说明



## 实验环境配置说明

- Anaconda、Spyder、OpenCV-Python安装

## 实验1内容：

- 阅读示例程序opencv\_example.py
- 实现水平翻转、高斯模糊、Canny边缘检测三个操作
- 水平拼接实验结果图，并在图的左上方标注组号

## 提交内容：

- 实验结果的图像及代码文件，打包上传至canvas

# 实验环境配置说明



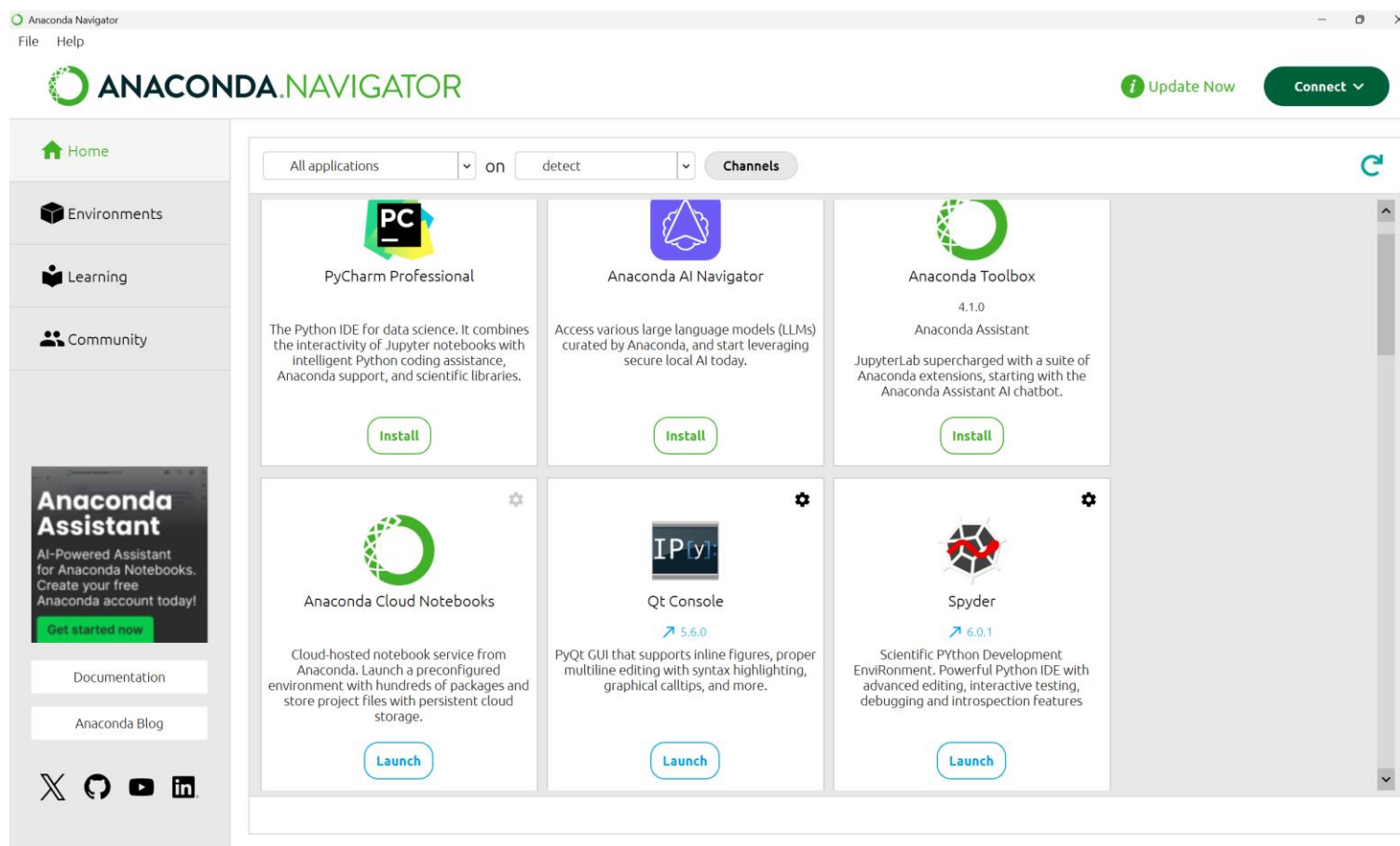
## Anaconda、Spyder安装

- 参考QQ群《控制导论-感知实验环境安装说明文档》完成
- Anaconda简介：
  - Anaconda 是一个免费且开源的**Python和R语言的发行版**，主要用于科学计算、数据分析、人工智能和机器学习等领域。
  - 主要优势在于能够帮助用户轻松**创建和管理隔离的虚拟环境**。支持主流操作系统，提供图形化界面（GUI）。
- Spyder简介：
  - Spyder 是一个开源的 **Python 集成开发环境（IDE）**。
  - 它提供了一个**功能齐全**的代码编辑器、交互式控制台、调试工具以及变量浏览器，使用户能够方便地编写、调试和分析 Python 代码。

# Anaconda



## Anaconda Navigator: 图形化界面，管理环境和应用



# Anaconda



## Anaconda Prompt: 命令行, 管理环境和包

```
Anaconda Prompt
(base) C:\Users\95811>conda env list
# conda environments:
#
base                * C:\Users\95811\anaconda3
detect              C:\Users\95811\anaconda3\envs\detect

(base) C:\Users\95811>conda activate detect
(detect) C:\Users\95811>pip install|
```

# Spyder



文件、运行、调试功能键

变量浏览器、调试器、绘图、文件功能区

The screenshot displays the Spyder Python IDE interface. The main window is titled "Spyder (Python 3.8)". The menu bar includes: 文件(F), 编辑(E), 查找(S), 源代码(C), 运行(R), 调试(D), 控制台(O), 项目(P), 工具(T), 查看(V), 帮助(H). The toolbar contains icons for file operations, running, and debugging. A dashed orange box highlights the "Run" button (a green play icon) and the "Debug" button (a blue play icon with a bug). The code editor shows a Python script named "opencv\_example.py" with the following content:

```
1  # -*- coding: utf-8 -*-
2  import cv2
3  import os
4
5  """在图像上添加文字标签"""
6  > def add_text_to_image(img, text, pos=(45, 750), bgr=(0, 255, 0)):
14
15  """处理单个图像文件"""
16  > def process_image_file(input_path, output_path, group_number):
50
51  """主函数"""
52  > def main(input_dir, output_dir, group_number=666):
73
74  if __name__ == "__main__":
75      # 配置输入图像的路径和输出结果的路径
76      input_dir = "E:\\projects\\opencv_project\\images"
77      output_dir = "E:\\projects\\opencv_project\\output"
78      # 组号
79      group_number = 6
80      # 执行主函数功能
81  main(input_dir, output_dir, group_number)
```

The variable explorer on the right shows the following variables:

名称	类型	大小	值
group_number	int	1	6
input_dir	str	33	E:\\projects\\opencv_project\\images
output_dir	str	33	E:\\projects\\opencv_project\\output

The IPython console at the bottom shows the execution of the script:

```
In [3]: %debugfile E:/projects/opencv_project/code/
opencv_example.py --wdir
> E:/projects/opencv_project/code/opencv_example.py (81)<module>()
77      output_dir = "E:\\projects\\opencv_project\\output"
78      # 组号
79      group_number = 6
80      # 执行主函数功能
3--> 81      main(input_dir, output_dir, group_number)

TDDh f11
```

The status bar at the bottom indicates: Conda: detect (Python 3.8.20) ✓ LSP: Python Line 76, Col 55 UTF-8 CRLF RW Mem 74%

运行结果

# OpenCV



## OpenCV简介与安装

- OpenCV简介：
  - OpenCV (Open Source Computer Vision Library) 是一个开源的计算机视觉库，旨在为**计算机视觉和图像处理**提供一系列工具和算法。
  - 支持多种编程语言（如Python、C++），提供**丰富的图像和视频处理功能**，包括图像基础操作、特征检测、目标识别、实时视频分析等。
- OpenCV-Python安装：
  - 打开Anaconda Prompt，进入创建好的detect实验环境
  - 输入**pip install opencv-python**，若下载较慢，可在指令后接安装源 `pip install opencv-python -i https://pypi.tuna.tsinghua.edu.cn/simple`

# 实验内容



- 阅读示例程序opencv\_example.py
  - 实现了图像1.BGR转RGB模式, 2.转灰度图
  - 注意在代码底部**修改文件路径**input\_dir和ouput\_dir
- 实现**水平翻转、高斯模糊、Canny边缘检测**三个操作
  - 查询OpenCV官方文档: <https://docs.opencv.org/4.11.0/> 实现
- 水平拼接实验结果图, 并在图的**左上方标注组号**
  - 结果示意图:





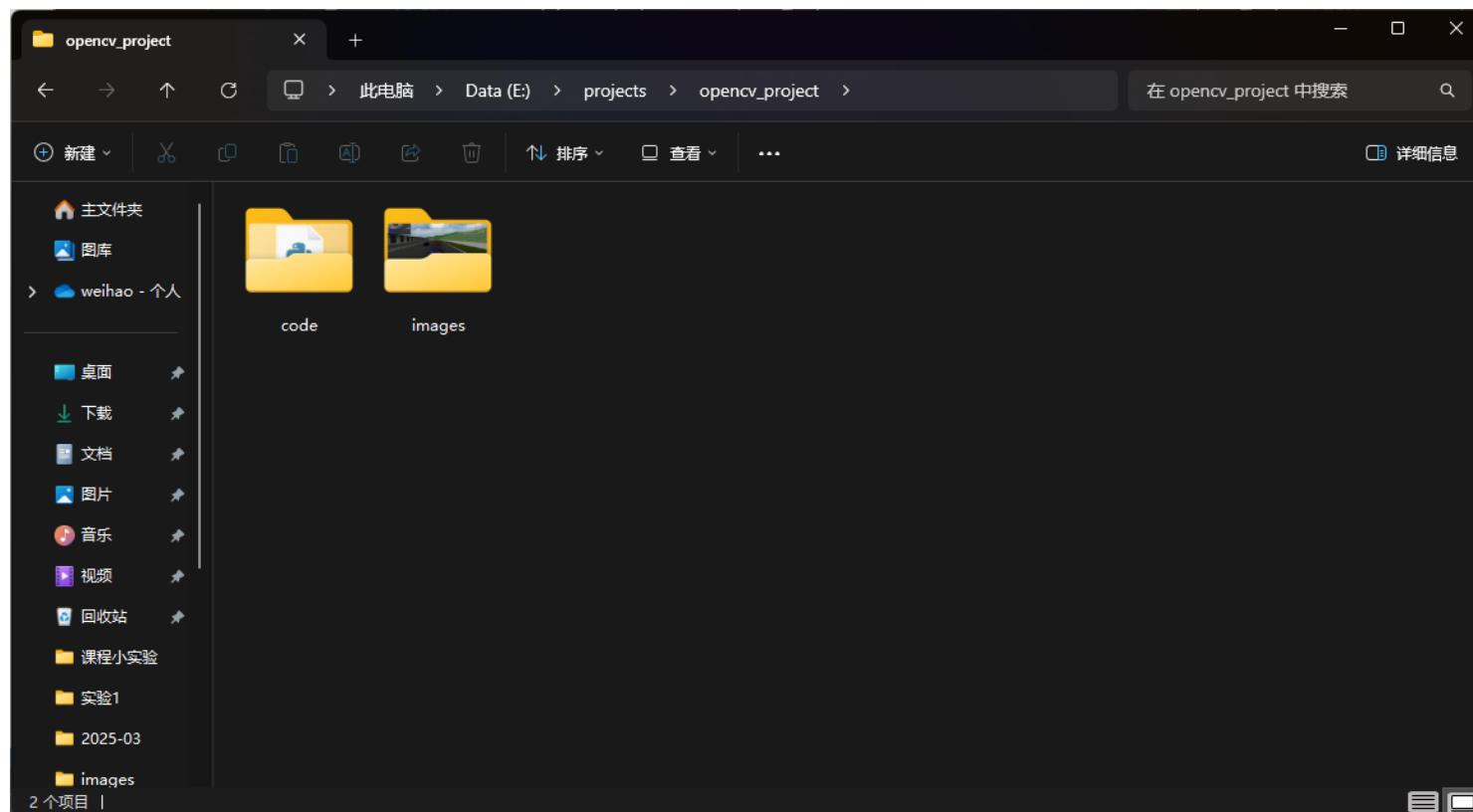
# 示例程序说明



文件结构：

解压缩opencv\_project.zip

code文件夹中为示例程序，images文件夹中为实验图片





# 示例程序说明



```
opencv_example.py X
1  # -*- coding: utf-8 -*-
2  import cv2
3  import os
4
5  """在图像上添加文字标签"""
6  > def add_text_to_image(img, text, pos=(45, 750), bgr=(0, 255, 0)):
14
15  """处理单个图像文件"""
16  > def process_image_file(input_path, output_path, group_number):
50
51  """主函数"""
52  > def main(input_dir, output_dir, group_number=666):
73
74  if __name__ == "__main__":
75      # 配置输入图像的路径和输出结果的路径
76      input_dir = "E:\\projects\\opencv_project\\images"
77      output_dir = "E:\\projects\\opencv_project\\output"
78      # 组号
79      group_number = 6
80      # 执行主函数功能
81      main(input_dir, output_dir, group_number)
```

需要添加  
三个操作

需要修改  
路径和组号

# 查询OpenCV官方文档



右上角关键词搜索：以水平翻转为例，输入flip，点击cv::flip()进入说明

ENHANCED BY Google

OpenCV 4.11.0  
Open Source Computer Vision

Main Page Related Pages Namespaces Classes Files Examples Java documentation

## OpenCV modules

- Introduction
- OpenCV Tutorials
- OpenCV-Python Tutorials
- OpenCV.js Tutorials
- Tutorials for contrib modules
- Frequently Asked Questions
- Bibliography
- Main modules:
  - core. Core functionality
  - imgproc. Image Processing
  - imgcodecs. Image file reading and writing
  - videoio. Video I/O
  - highgui. High-level GUI
  - video. Video Analysis
  - calib3d. Camera Calibration and 3D Reconstruction
  - features2d. 2D Features Framework
  - objdetect. Object Detection
  - dnn. Deep Neural Network module
  - ml. Machine Learning
  - flann. Clustering and Search in Multi-Dimensional Spaces
  - photo. Computational Photography
  - stitching. Images stitching
  - gapi. Graph API
- Extra modules:
  - alphasat. Alpha Matting
  - aruco. Aruco markers, module functionality was moved to objdetect module
  - bgsegm. Improved Background-Foreground Segmentation Methods
  - bioinspired. Biologically inspired vision models and derived tools
  - cannops. Ascend-accelerated Computer Vision
  - ccalib. Custom Calibration Pattern for 3D reconstruction

flip

```
cv::flip()
cv::gapi::flip()
cv::cann::flip(InputArray src, OutputArray dst, int
flipCode, AscendStream
&stream=AscendStream::Null())
cv::cann::flip(const AscendMat &src, AscendMat &dst,
int flipCode, AscendStream
&stream=AscendStream::Null())
cv::cann::flip(const AscendMat &src, std::vector<
int32_t> &axes, AscendMat &dst, AscendStream
&stream)
cv::cuda::flip()
```

flipND

# 查询OpenCV官方文档



## 查看Python指令、功能和参数说明

ENHANCED BY Google

OpenCV 4.11.0  
Open Source Computer Vision

[Main Page](#) [Related Pages](#) [Namespaces](#) [Classes](#) [Files](#) [Examples](#) [Java documentation](#)

### Python指令

```
• flip()  
void cv::flip ( InputArray src,  
               OutputArray dst,  
               int flipCode )  
  
Python:  
cv.flip( src, flipCode[, dst] ) -> dst
```

### 功能说明

```
#include <opencv2/core.hpp>  
  
Flips a 2D array around vertical, horizontal, or both axes.  
  
The function cv::flip flips the array in one of three different ways (row and column indices are 0-based):  
  

$$dst_{ij} = \begin{cases} src_{src.rows-i-1,j} & \text{if } flipCode = 0 \\ src_{i,src.cols-j-1} & \text{if } flipCode > 0 \\ src_{src.rows-i-1,src.cols-j-1} & \text{if } flipCode < 0 \end{cases}$$
  
  
The example scenarios of using the function are the following: Vertical flipping of the image (flipCode == 0) to switch between top-left and bottom-left image origin. This is a typical operation in video processing on Microsoft Windows® OS. Horizontal flipping of the image with the subsequent horizontal shift and absolute difference calculation to check for a vertical-axis symmetry (flipCode > 0). Simultaneous horizontal and vertical flipping of the image with the subsequent shift and absolute difference calculation to check for a central symmetry (flipCode < 0). Reversing the order of point arrays (flipCode > 0 or flipCode == 0).
```

### 参数说明

**Parameters**

- src** input array.
- dst** output array of the same size and type as src.
- flipCode** a flag to specify how to flip the array; 0 means flipping around the x-axis and positive value (for example, 1) means flipping around y-axis. Negative value (for example, -1) means flipping around both axes.

**See also**  
transpose, repeat, completeSymm

Here is the call graph for this function: