

小组课程设计问题汇总

1. Jetson Nano 识别新 USB 摄像头问题

问题：Jetson Nano 开发板原本配置了 CSI 接口摄像头（此摄像头调用存在问题，我们组选择更换另一种摄像头），而我们需要更换为 USB 摄像头。在初次接入 USB 摄像头时，系统无法自动识别并显示图像。这个问题主要源于 Jetson Nano 默认驱动支持的是 CSI 接口摄像头，因此 USB 摄像头在系统中无法直接识别。

解决方法：我们首先检查了 USB 摄像头的驱动是否正确安装。通过查看系统的设备管理器，确认没有识别到 USB 摄像头。接着，我们手动安装了与 USB 摄像头兼容的驱动程序，并根据 Jetson Nano 的操作系统（Ubuntu 18.04）配置了相关的依赖库。然后我们输入命令查看摄像头加载信息，同时查看摄像头参数信息。

```
jetson@yahboom: ~  
jetson@yahboom:~$ ls dev/video*  
ls: cannot access 'dev/video*': No such file or directory  
jetson@yahboom:~$ ls /dev/video*  
/dev/video0 /dev/video1  
jetson@yahboom:~$ v4l2-ctl --device=/dev/video1 --list-formats-ext  
bash: v4l2-ctl: command not found  
jetson@yahboom:~$ v4l2-ctl --device=/dev/video1 --list-formats-ext  
bash: v4l2-ctl: command not found  
jetson@yahboom:~$ v4l2-ctl --device=/dev/video1 --list-formats-ext  
ioctl: VIDIOC_ENUM_FMT  
  Index       : 0  
  Type        : Video Capture  
  Pixel Format: 'YUYV'  
  Name        : YUYV 4:2:2  
                Size: Discrete 640x480  
                  Interval: Discrete 0.033s (30.000 fps)  
  
  Index       : 1  
  Type        : Video Capture  
  Pixel Format: 'Z16 '  
  Name        : 16-bit Depth  
                Size: Discrete 640x480  
                  Interval: Discrete 0.017s (60.000 fps)  
                  Interval: Discrete 0.033s (30.000 fps)  
                  Interval: Discrete 0.100s (10.000 fps)  
                Size: Discrete 640x240
```

图 1 查看摄像头参数信息

2. 摄像头参数配置问题

问题：在使用新的 USB 摄像头进行实时图像捕捉时，出现了图像延迟、卡顿和不稳定的问题。通过调试发现，这与摄像头的分辨率、帧率以及 Jetson Nano 的硬件性能有关。Jetson Nano 的处理能力相对有限，过高的分辨率和帧率导致

视频流处理不及时，从而影响了实时性。

解决方法：我们对摄像头的分辨率和帧率进行了优化调整。首先，通过调低分辨率和帧率（如调整为 640x480 和 15 帧每秒），使得数据量减小，避免处理过载。其次，在 Jetson Nano 中配置了适合其硬件性能的摄像头参数，为了进一步优化图像质量和提高稳定性，我们还使用了 OpenCV 库中的内置函数对图像进行预处理。经过这些优化，摄像头能够更加流畅地捕捉用户的面部信息，保证了图像流的稳定性和实时性。

```
def open_camera(self):
    # 检测该设备是否能打开
    flag = self.cap.open(0)
    if flag is False:
        QMessageBox.critical(self, parent: "错误", QWidget: "无法打开摄像头。")
        return
    else:
        self.cap.set(propId: 6, cv2.VideoWriter.fourcc(c1: 'M', c2: 'J', c3: 'P', c4: 'G'))
        self.cap.set(cv2.CAP_PROP_FRAME_WIDTH, value: 640)
        self.cap.set(cv2.CAP_PROP_FRAME_HEIGHT, value: 480)
        if self.pushButton_monitor.text() == '开始监测':
            # 幕布可以播放
```

图 2 修改摄像头参数

3. facenet 模型的下载和测试

问题：表情识别的基础是首先要检测到人脸，人脸检测也需要一个模型来完成识别，其中 facenet 库就提供了一个检测人脸的方法库，于是我们打算调用这个方法来完成人脸识别功能。但如果只是简单的使用 `pip install facenet_pytorch`，会提示 python 和 facenet_pytorch 的版本不兼容

解决办法：jetson nano 本身就有 python 解释器环境（默认的是 python 2.7），如果直接下载 facenet_pytorch 会显示和 python2.7 的版本不兼容，解决方法需要指定 python 的版本（我们开发板上还有 python 3.6 的版本），需要执行 `python3 -m pip install facenet_pytorch` 才能消除版本兼容问题。

如下图所示：

```
Terminal: Local x +
jetson@yahboom:~/Desktop/facial$ python3 -m pip install facenet_pytorch
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: facenet_pytorch in /home/jetson/.local/lib/python3.6/site-packages (2.5.3)
Requirement already satisfied: requests in /home/jetson/.local/lib/python3.6/site-packages (from facenet_pytorch) (2.27.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages (from facenet_pytorch) (1.19.5)
Requirement already satisfied: torchvision in /home/jetson/.local/lib/python3.6/site-packages/torchvision-0.9.0-py3.6-linux-aarch64.whl (from facenet_pytorch) (0.9.0)
Requirement already satisfied: pillow in /home/jetson/.local/lib/python3.6/site-packages (from facenet_pytorch) (8.4.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.6/dist-packages (from requests->facenet_pytorch) (3.4)
Requirement already satisfied: charset-normalizer<=2.0.0 in /home/jetson/.local/lib/python3.6/site-packages (from requests->facenet_pytorch) (2.0.12)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/lib/python3/dist-packages (from requests->facenet_pytorch) (1.22)
```

图 3 python 解释器版本问题

模型检测效率问题: 虽然能够在开发板调用模型, 但是模型的效果并不理想, 可以看到调用模型后, 摄像头已经打开 (出现绿光就表示成功打开), 但是集成环境还没有加载出识别界面, 模型调用相当卡顿, 在识别过程中, 延迟也比较大。因此, 我们最终决定更换为 haarcascade_frontalface_default.xml, 添加配置文件。

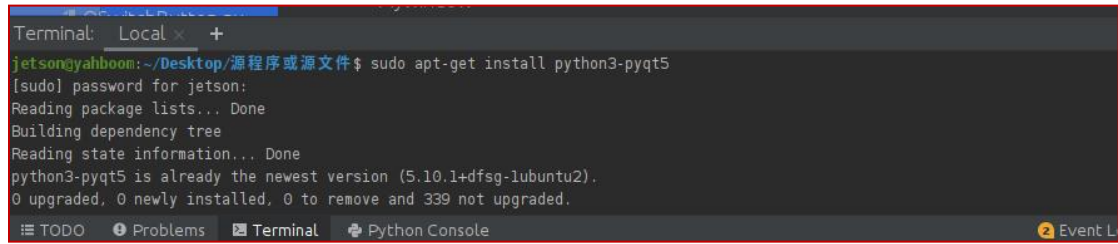


图 4 facenet 模型效果测试

4. 在本地安装 pyqt5

问题: PyQt5 是一个重要的前端开发工具, 为了能够在开发板中完成前端界面的开发, 我们决定使用 pyqt5 进行开发, 但是如果安装 pyqt5 时只是调用 pip install pyqt5 (即使添加 python 版本解释器), 其也无法正常下载, 会出现包的依赖一直无法下载成功。

解决办法：在本地调用命令 `sudo apt-get install python3-pyqt5` 才能够正常安装，如下图所示，这样 pyqt5 才能正常安装。

A screenshot of a terminal window with a dark background. The prompt is 'jetson@yahboom: ~/Desktop/源程序或源文件\$'. The command 'sudo apt-get install python3-pyqt5' has been entered. The output shows the system reading package lists, building a dependency tree, and reading state information. It then states that 'python3-pyqt5 is already the newest version (5.10.1+dfsg-1ubuntu2)' and that '0 upgraded, 0 newly installed, 0 to remove and 339 not upgraded.' The terminal window has tabs for 'Terminal' and 'Python Console' at the bottom.

```
Terminal: Local x +
jetson@yahboom: ~/Desktop/源程序或源文件$ sudo apt-get install python3-pyqt5
[sudo] password for jetson:
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-pyqt5 is already the newest version (5.10.1+dfsg-1ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 339 not upgraded.
```

图 5 pyqt5 的安装