

《计算机视觉》第3周作业

北京大学-计算机视觉-张健

阮洁

2021年10月

《计算机视觉》第3周作业

一、课程任务

二、任务完成

1. 安装PyTorch
2. 运行所有ipynb文件
3. 视频人脸检测
4. 输出运行后的结果

一、课程任务

- 安装PyTorch
- 运行所有ipynb文件
- 视频人脸检测
- 输出运行后的结果

二、任务完成

1. 安装PyTorch

在PyTorch官网找到想下载的torch和torchvision版本的命令

官网链接: <https://pytorch.org/get-started/previous-versions/>

PyTorch

Get Started

Ecosystem

Mobile

Blog

Tutorials

Docs

Resources

GitHub

v1.2.0

Conda

OSX

```
# conda
conda install pytorch==1.2.0 torchvision==0.4.0 -c pytorch
```

Linux and Windows

```
# CUDA 9.2
conda install pytorch==1.2.0 torchvision==0.4.0 cudatoolkit=9.2 -c pytorch

# CUDA 10.0
conda install pytorch==1.2.0 torchvision==0.4.0 cudatoolkit=10.0 -c pytorch

# CPU Only
conda install pytorch==1.2.0 torchvision==0.4.0 cpuonly -c pytorch
```

在Linux中下载执行相关的命令
查看环境中已安装的PyTorch

```
root@a3q50csm9v792-0:/CIS20# conda activate PyTorch1.2
(PyTorch1.2) root@a3q50csm9v792-0:/CIS20# python
Python 3.6.13 |Anaconda, Inc.| (default, Jun  4 2021, 14:25:59)
[GCC 7.5.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> torch. version
'1.2.0'
>>>
```

2. 运行所有ipynb文件

已运行所有ipynb文件，弄懂不理解的地方，并在代码中加上相应的注释。

3.视频人脸检测

```
import cv2
# 保存好的视频检测人脸并截图
def CatchPICFromVideo(window_name, camera_idx, catch_pic_num, path_name):
    cv2.namedWindow(window_name)

    # 视频来源
    cap = cv2.VideoCapture(camera_idx)

    # 告诉OpenCV使用人脸识别分类器
    classifier = cv2.CascadeClassifier("haarcascade_frontalface_alt.xml")

    # 识别出人脸后要画的边框的颜色，RGB格式，color是一个不可增删的数组
    color = (0, 255, 0)

    num = 0
    while cap.isOpened():
```

```

ok, frame = cap.read() # 读取一帧数据
if not ok:
    break

# 将当前帧图像转换成灰度图像
grey = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

# 人脸检测, 1.2和2分别为图片缩放比例和需要检测的有效点数
faceRects = classifier.detectMultiScale(grey, scaleFactor=1.2, minNeighbors=3, minSize=(32, 32))
if len(faceRects) > 0: # 大于0则检测到人脸
    for faceRect in faceRects: # 单独框出每一张人脸
        x, y, w, h = faceRect

        # 将当前帧保存为图片
        img_name = "%s/%d.jpg" % (path_name, num)
        image = frame[y - 10: y + h + 10, x - 10: x + w + 10]
        cv2.imwrite(img_name, image, [int(cv2.IMWRITE_PNG_COMPRESSION), 9])

        num += 1
        if num > (catch_pic_num): # 如果超过指定最大保存数量退出循环
            break

        # 画出矩形框
        cv2.rectangle(frame, (x - 10, y - 10), (x + w + 10, y + h + 10), color, 2)

        # 显示当前捕捉到了多少人脸图片
        font = cv2.FONT_HERSHEY_SIMPLEX
        cv2.putText(frame, 'num:%d/100' % (num), (x + 30, y + 30), font, 1, (255, 0, 255), 4)

    # 超过指定最大保存数量结束程序
    if num > (catch_pic_num):
        break

# 显示图像
cv2.imshow(window_name, frame)
c = cv2.waitKey(10)
if c & 0xFF == ord('q'):
    break

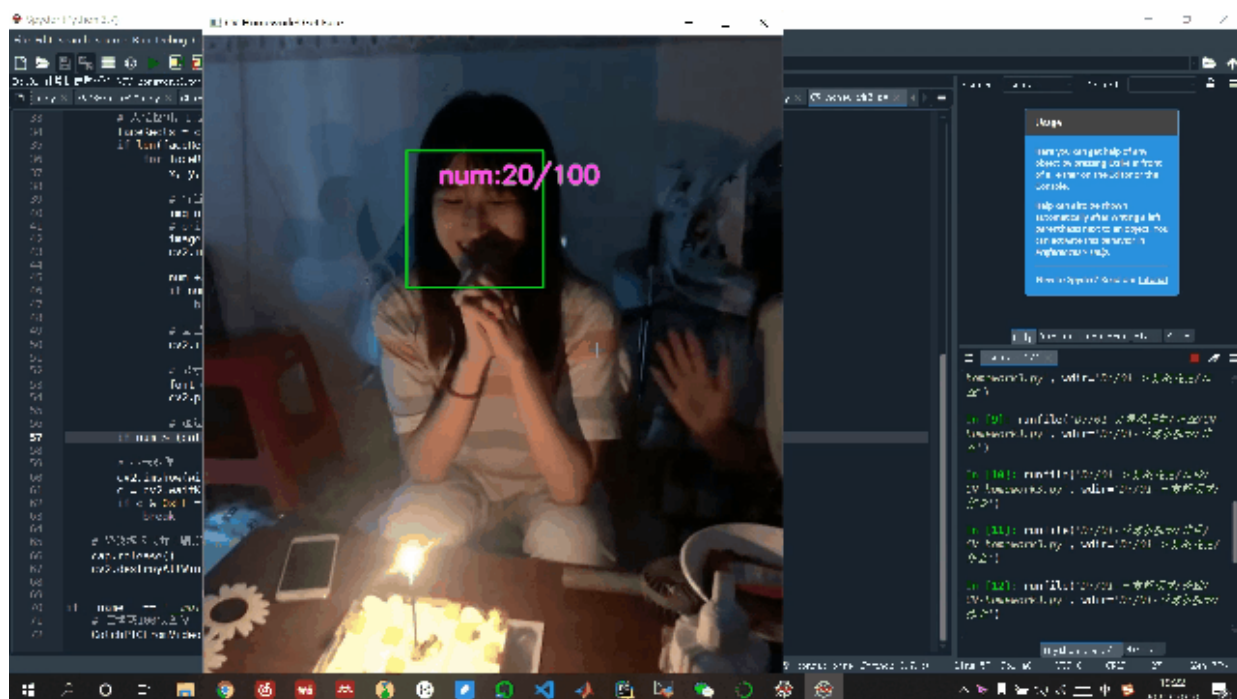
# 释放摄像头并销毁所有窗口
cap.release()
cv2.destroyAllWindows()

if __name__ == '__main__':
    # 连续截50张图片
    CatchPICFromVideo("CV-Homework3 Get Face", "CV-Homework3-2.mp4", 50,
        "./VideoCapture")

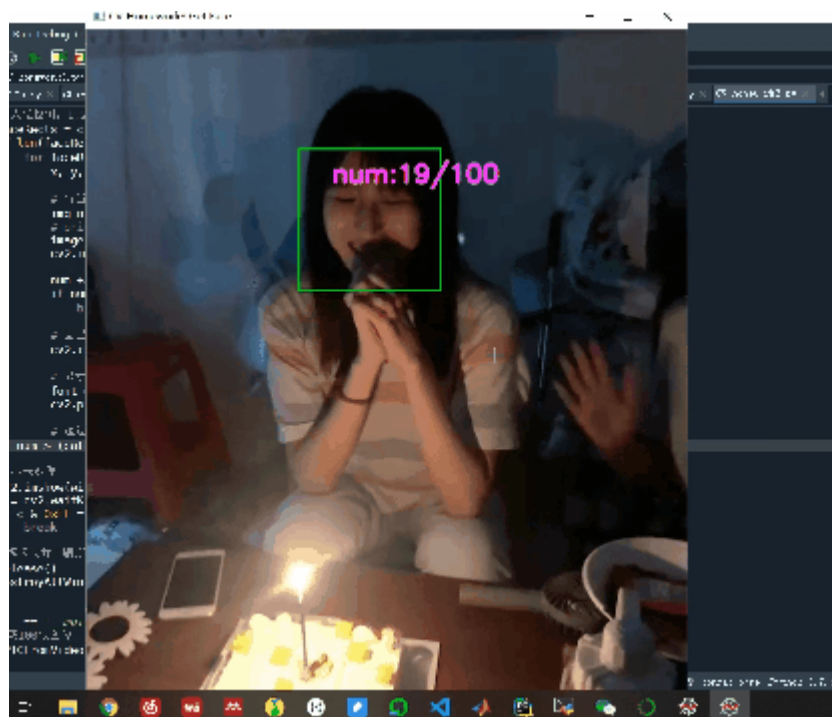
```

4.输出运行后的结果

运行过程及结果视频：



gif动图如果无法播放，则查看github同目录下的gif图片即可



在VideoCapture中截取了五十张识别到的人脸图片：

