

Computer Vision for Human

为人写的图形图像学课程

route for none-computer users' openCV lessons with examples and source code

🔔 重点利用Opencv积极联系Python的技能

课程特色 **Features**

第一阶段

- introduce to computer vision (cv2)
- Tricks of install cv 如何安装技巧
- loading picture/camera/movie 加载图片和摄像头或视频
- cvui 简单的图形界面 headfile GUI for openCV
- Tesseract-ocr 识别与训练

第二阶段

- basic image processing 基本图像处理
- TrainData DIY 字体识别库diy
- Image prop. correction methods 图片透视校正
- Table/grid/web data fetch 表格数据榨取

AI SEE

第一阶段

5 minutes introduction of computer vision (cv2)

5 分钟快速上手教程

Install

针对非**MS Microsoft VisualStudio** 的方法 比如**MAC**

- install python 2.6
- install cv2 numpy matplotlib cvui

```
pip install --upgrade setuptools  
pip install matplotlib  
pip install numpy  
pip install opencv-python==版本号
```

- Via pip you can specify the package version to install using the following: 指定版本号

```
pip install opencv-python==2.4.9
```

However, that package does not seem to be available on pypi. A little trick for checking available versions:

```
pip install opencv-python==
```

Which returns: Could not find a version that satisfies the requirement opencv-python== (from versions: 3.1.0.0, 3.1.0.1, 3.1.0.2, 3.1 .0.3, 3.1.0.5, 3.2.0.6, 3.2.0.7)

如果以上都不行，把

D:\opencv\opencv\build\python\2.7\x64 下的
cv2.pyd 复制到 **C:\Python27\Lib**的目录下

安装 **Visual Studio 2017 Community**

目前仅在 **Visual Studio for Windows** 中提供 **Python** 支持；在 **Mac** 和 **Linux** 上，可通过 **Visual Studio Code** 获取 **Python** 支持。

下载并运行适用于 Windows 的最新 Visual Studio 2017 安装程序（版本 15.2 及更高版本提供 Python 支持）。如果已安装 Visual Studio，请运行 Visual Studio 安装程序

对于 **Python**，请选择 **Python** 开发工作负载，然后选择“安装”：

若要快速测试 **Python** 支持，请启动 **Visual Studio**，按 **Alt+I** 打开 **Python** 交互窗口，然后输入

```
2+2
```

如果看不到输出 **4**，请重新检查步骤。

测试安装是否成功环境 在**Python**环境

- test code

```
import cv2  
print cv2.CV_AA
```

First examples in CV

入门案例

- loading picture
- loading video data

Show Image Example

显示图片

```
import cv2
import numpy as np

img = cv2.imread("girl.jpg",0)
height, width = img.shape
cv2.imshow("Hello World",img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Cam Example

加载摄像头

```
# use computer
import cv2
import numpy as np

cap = cv2.VideoCapture(0)

cap.set(cv2.cv.CV_CAP_PROP_FRAME_WIDTH,1080)
cap.set(cv2.cv.CV_CAP_PROP_FRAME_HEIGHT,720)

while(1):
    # get a frame
    ret, frame = cap.read()
    # show a frame
    cv2.imshow(" Video example exit by 'q' ", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()

cv2.destroyAllWindows()
```

使用**matplotlib** 图像显示库

```
import numpy as np
import cv2
from matplotlib import pyplot as plt

img = cv2.imread('messi5.jpg',0)
plt.imshow(img, cmap = 'gray', interpolation = 'bicubic')
plt.xticks([]), plt.yticks([]) # to hide tick values on X and Y axis
plt.show()
```


CVUI

A (very) simple UI lib built on top of OpenCV drawing primitives. Other UI libs, such as ImGui, require a graphical backend (e.g. OpenGL) to work, so if you want to use ImGui in a OpenCV app, you must make it OpenGL enabled, for instance. It is not the case with cvui, which uses only OpenCV drawing primitives to do all the rendering (no OpenGL or Qt required).

CVUI 是一个基于 **OpenCV** 自建绘图模块的图形界面库，其他的图形界面库，如**ImGui**，需要界面后台库**Opengl**或**QT**支持，如果我们只需要使用一些比较常见的**UI**功能，我们只需要**OpenCV**来完成所有的渲染功能。

UI example

```
import numpy as np
import cv2
import cvui

WINDOW_NAME = 'CVUI Test'

cvui.init(WINDOW_NAME)
frame = np.zeros((200, 400, 3), np.uint8)

while True:
    frame[:] = (49, 52, 49)
    cvui.text(frame, 10, 15, 'Hello world!')
    # Update cvui internal stuff
    cvui.update()
    # Show window content
    cv2.imshow(WINDOW_NAME, frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```

- NOTES change code CV2_LINE to CV_AA

Homework

作业

创建一个可以显示摄像头内容的窗体，在窗体内部放置一个checkbox，通过checkbox 来改变图形的性质 比如颜色图像或黑白图像

Easy OCR Introduction

简单**OCR**教程

project Tesseract-OCR

Tesseract最早**1985**由**HP**公司开发，后转为开源项目，由于获得**GOOGLE**公司的大力资助，项目得以延续，现在的版本为**4.0 beta**,内核上支持**RNN**技术

Tesseract 安装与使用

安装pytesseract

```
pip install pytesseract
```

C:\Python27\Lib\site-packages\pytesseract

```
tesseract_cmd = 'tesseract'
```

改成如下

```
tesseract_cmd = 'c:/Program Files (x86)/Tesseract-OCR/tesseract.exe'
```

在**python** 环境调用 **Tesseract**

字符转文字 (英文字符)

识别图片中的英文文字

```
import cv2
import pytesseract
from PIL import Image
import os

text = pytesseract.image_to_string(Image.open("testocr.jpg"))
print(text)
```


识别图片中的中文文字

```
#coding=utf-8
#coding=utf-8import cv2
import pytesseract
try:
    from PIL import Image
except ImportError:
    import Image
import pytesseract
import os

text = pytesseract.image_to_string(Image.open("testocr.tif"), lang='chi_sim')
print(text)
```

Tesseract 训练教程

手工训练教程

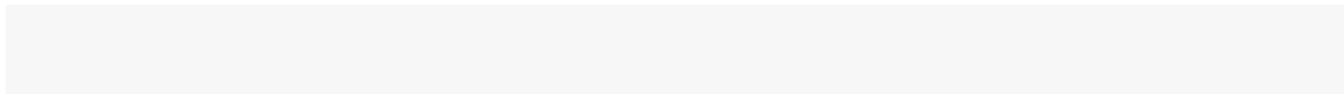
重命名图像文件

产生**box**文件

打包字库



上述过程 **easy_font_batch.py**



Homework

作业

训练一个能识别“**ABCD**”手写字符视屏
动态识别的“**robot**”

AI SEE

第二阶段

可以进阶了吗？

basic image processing

基本图像处理

1.认识色彩通道

2.直方图

3.图像混合

4.降噪、模糊函数

5.二值处理

1.区域处理

2.基础抖动函数

3.**dilate()**函数

4.基础边界处理

1.图像透视修正

通过四个点进行画面修正

综合案例

其他内容

openCV 中文显示

多线程处理

Fin

