

DRVideoCV 视频文件采集控件用法

DRVideoCV 视频文件采集控件是利用 OPenCV 库实现；因此，使用前选需安装 Python 的 OpenCV 支持库。注意，该控件只采集视频画面，不采集音频信号；另外，该控件为后台工作控件，GUI 界面上无显示。

DRVideoCV 视频文件采集控件封装在 drvi.drOpenCV 中：

```
import drvi.drOpenCV as drcv
```

在 Python 程序 Tkinter GUI 界面中创建一个 DRVideoCV 控件的命令和语法格式如下：

```
hl=DRVideoCV(win,x,y,w,h,option=value, ...)
```

其中输入参数 win 为 Tkinter 库创建的 GUI 窗口句柄，[x,y,w,h]四个参数为该控件插入 GUI 窗口 win 时的左上角坐标[x,y]和该控件的宽、高[w,h]。另外是一组可选的输入参数，包括：

mfile: 视频文件名，默认值=''

imgType: 输出图像格式，默认值=0，0（RGB Array）,1(PIL Image)

rate: 图像采样选抽率，默认值=15

返回值 hl 则是该控件创建后的句柄变量名，可以通过 hl 对创建后的控件进行进一步的设定。

DRVideoCV 控件有 2 个属性函数，其功能定义如下：

hl.setValueSingle(v): 视频信号采集开关，0/1=运行/停止。

hl.setValueString(txt): 设定播放的视频文件名。

下面是一个全参数的 DRVideoCV 控件创建实例：

```
hl=drcv.DRVideoCV(self.win,583,114,80,40,mfile="",imgType=0,rate=10)
```

通常，可以选择设定部分选择参数的形式进行设定，除基本参数 win、[x,y,w,h]和用变量名设定的可选参数外，其余可选参数采用默认值。创建实例如下：

```
hl=drcv.DRVideoCV(win,277,23,80,40)
```

创建后可以通过返回的控件句柄对其属性函数进行操作。

DRVideoCV 控件可以通过绑定的方式将采集的视频图像传递给绑定的回调函数或其他控件，格式为：

```
hl.addCallBackImage(fun)
```

被绑定的回调函数或另一控件的 set 函数是一个图像函数，收到的是采集的图像数据。回调函数形式为：

```
def setImage(im):
```

```
...
```

下面是 DRVideoCV 控件的创建和应用示例小程序，功能是在 GUI 界面上插入 1 个按钮控件，一个文件选择控件，一个 DRVideoCV 控件和 1 个 DRImage 图像显示控，示例小程序代码如下：

```
import tkinter as tk
import drvi.drviControlls as dr
import drvi.drOpenCV as drcv

#==Main Window=====
win= tk.Tk()
win.wm_title('Main')
winWidthCur=816
winHeightCur=497
win.geometry('816x497')
```

```

win.config(bg="#ddee")
#==Layout of controls=====
mButton1=dr.DRButton(win,55,33,100,40,'#00aacc','#eee00','运行/停止',12,101)
mFileBut2=dr.DRFileBut(win,181,31,110,40,'#00aacc','#eee00','文件选择','mp4',12)
mImage3=dr.DRImage(win,63,106,420,260,'#ddddd')
mVideo4=drcv.DRVideoCV(win,486,65,80,40,mfile='',mid=700,imgType=0,rate=10)
#User defined data probe=====
mButton1.addCallBackSingle(mVideo4.setValueSingle)
mFileBut2.addCallBackString(mVideo4.setValueString)
mVideo4.addCallBackImage(mImage3.setValueImage)
#==Main Loop=====
win.mainloop()

```

pyqt 版本中的用法说明:

均继承自 OpenCV，使用无区别，但 pyqt 版建议将该控件的输出图像格式设为 0（BGR Array）而不是 1(PIL Image)，否则需要将 PIL 的 Image 格式转换为 QImage 才方便显示。

另外，pyqt 版的 DRPlayr 控件也支持视频采集，视频采集更直接和高效。