opencv代码实现

1. **视频读写**

#打开摄像头

import cv2

cv2.namedWindow('video',cv2.WINDOW\_NORMAL)

cv2.resizeWindow('video',640,480)

cap=cv2.VideoCapture(0)

#循环读取摄像头的每一帧

while True:

ret,frame=cap.read()

if not ret:

break

cv2.imshow('video',frame)

key=cv2.waitKey(0)

if key==ord('q'):

break

cap.release()

cv2.destoryAllWindows()

import numpy as np

import cv2 as cv

# 1.获取视频对象

cap = cv.VideoCapture('parking\_video.mp4')

# 2.判断是否读取成功

while(cap.isOpened()):

# 3.获取每一帧图像

ret, frame = cap.read()

# 4. 获取成功显示图像

if ret == True:

cv.imshow('frame',frame)

# 5.每一帧间隔为25ms

if cv.waitKey(25) & 0xFF == ord('q'):

break

# 6.释放视频对象

cap.release()

cv.destoryAllwindows()

**二保存视频**

import cv2 as cvimport numpy as np

# 1. 读取视频

cap = cv.VideoCapture("DOG.wmv")

# 2. 获取图像的属性（宽和高，）,并将其转换为整数

frame\_width = int(cap.get(3))

frame\_height = int(cap.get(4))

# 3. 创建保存视频的对象，设置编码格式，帧率，图像的宽高等

out = cv.VideoWriter('outpy.avi',cv.VideoWriter\_fourcc('M','J','P','G'), 10, (frame\_width,frame\_height))while(True):

# 4.获取视频中的每一帧图像

ret, frame = cap.read()

if ret == True:

# 5.将每一帧图像写入到输出文件中

out.write(frame)

else:

break

# 6.释放资源

cap.release()

out.release()

cv.destroyAllWindows()

1. **几何变换**

**1.图像缩放**

import cv2 as cv# 1. 读取图片

img1 = cv.imread("./image/dog.jpeg")# 2.图像缩放# 2.1 绝对尺寸

rows,cols = img1.shape[:2]

res = cv.resize(img1,(2\*cols,2\*rows),interpolation=cv.INTER\_CUBIC)

# 2.2 相对尺寸

res1 = cv.resize(img1,None,fx=0.5,fy=0.5)

# 3 图像显示# 3.1 使用opencv显示图像(不推荐)

cv.imshow("orignal",img1)

cv.imshow("enlarge",res)

cv.imshow("shrink）",res1)

cv.waitKey(0)

## 2图像旋转

import numpy as np

import cv2 as cv

import matplotlib.pyplot as plt# 1 读取图像

img = cv.imread("./image/image2.jpg")

# 2 图像旋转

rows,cols = img.shape[:2]# 2.1 生成旋转矩阵

M = cv.getRotationMatrix2D((cols/2,rows/2),90,1)# 2.2 进行旋转变换

dst = cv.warpAffine(img,M,(cols,rows))

cv.imshow("orignal",img1)

cv.imshow("enlarge",res)

cv.imshow("shrink）",res1)

cv.waitKey(0)

## 仿射变换

import numpy as np

import cv2 as cv

import matplotlib.pyplot as plt

# 1 图像读取

img = cv.imread("./image/image2.jpg")

# 2 仿射变换

rows,cols = img.shape[:2]# 2.1 创建变换矩阵

pts1 = np.float32([[50,50],[200,50],[50,200]])

pts2 = np.float32([[100,100],[200,50],[100,250]])

M = cv.getAffineTransform(pts1,pts2)# 2.2 完成仿射变换

dst = cv.warpAffine(img,M,(cols,rows))

cv.imshow("orignal",img1)

cv.imshow("enlarge",res)

cv.imshow("shrink）",res1)

cv.waitKey(0)

## 透射变换

import numpy as np

import cv2 as cv

import matplotlib.pyplot as plt# 1 读取图像

img = cv.imread("./image/image2.jpg")# 2 透射变换

rows,cols = img.shape[:2]# 2.1 创建变换矩阵

pts1 = np.float32([[56,65],[368,52],[28,387],[389,390]])

pts2 = np.float32([[100,145],[300,100],[80,290],[310,300]])

T = cv.getPerspectiveTransform(pts1,pts2)# 2.2 进行变换

dst = cv.warpPerspective(img,T,(cols,rows))

cv.imshow("orignal",img1)

cv.imshow("enlarge",res)

cv.imshow("shrink）",res1)

cv.waitKey(0)

## 5.图像金字塔

import numpy as npimport cv2 as cvimport matplotlib.pyplot as plt# 1 图像读取

img = cv.imread("./image/image2.jpg")# 2 进行图像采样

up\_img = cv.pyrUp(img) # 上采样操作

img\_1 = cv.pyrDown(img) # 下采样操作# 3 图像显示

cv.imshow('enlarge', up\_img)

cv.imshow('original', img)

cv.imshow('shrink', img\_1)

cv.waitKey(0)

cv.destroyAllWindows()

1. **形态学操作**
2. **腐蚀，膨胀**

import numpy as np

import cv2 as cv

import matplotlib.pyplot as plt

# 1 读取图像

img = cv.imread("./image/image3.png")# 2 创建核结构

kernel = np.ones((5, 5), np.uint8)

# 3 图像腐蚀和膨胀

erosion = cv.erode(img, kernel) # 腐蚀

dilate = cv.dilate(img,kernel) # 膨胀

cv.imshow('enlarge', up\_img)

cv.imshow('original', img)

cv.imshow('shrink', img\_1)

cv.waitKey(0)

cv.destroyAllWindows()

## 开闭运算

import numpy as np

import cv2 as cv

import matplotlib.pyplot as plt

# 1 读取图像

img1 = cv.imread("./image/image5.png")

img2 = cv.imread("./image/image6.png")# 2 创建核结构

kernel = np.ones((10, 10), np.uint8)# 3 图像的开闭运算

cvOpen = cv.morphologyEx(img1,cv.MORPH\_OPEN,kernel) # 开运算

cvClose = cv.morphologyEx(img2,cv.MORPH\_CLOSE,kernel)# 闭运算

cv.imshow('enlarge', up\_img)

cv.imshow('original', img)

cv.imshow('shrink', img\_1)

cv.waitKey(0)

cv.destroyAllWindows()

## 3.礼帽和黑帽

import numpy as np

import cv2 as cv

import matplotlib.pyplot as plt

# 1 读取图像

img1 = cv.imread("./image/image5.png")

img2 = cv.imread("./image/image6.png")# 2 创建核结构

kernel = np.ones((10, 10), np.uint8)# 3 图像的礼帽和黑帽运算

cvOpen = cv.morphologyEx(img1,cv.MORPH\_TOPHAT,kernel) # 礼帽运算

cvClose = cv.morphologyEx(img2,cv.MORPH\_BLACKHAT,kernel)# 黑帽运算

cv.imshow('enlarge', up\_img)

cv.imshow('original', img)

cv.imshow('shrink', img\_1)

cv.waitKey(0)

cv.destroyAllWindows()

**四．图像平滑**

## 1.均值滤波

import cv2 as cv

import numpy as np

from matplotlib import pyplot as plt# 1 图像读取

img = cv.imread('./image/dogsp.jpeg')# 2 均值滤波

blur = cv.blur(img,(5,5))

cv.imshow('enlarge', up\_img)

cv.imshow('original', img)

cv.imshow('shrink', img\_1)

cv.waitKey(0)

cv.destroyAllWindows()

## 2. 高斯滤波

import cv2 as cv

import numpy as np

from matplotlib import pyplot as plt

# 1 图像读取

img = cv.imread('./image/dogGasuss.jpeg')# 2 高斯滤波

blur = cv.GaussianBlur(img,(3,3),1)

cv.imshow('enlarge', up\_img)

cv.imshow('original', img)

cv.imshow('shrink', img\_1)

cv.waitKey(0)

cv.destroyAllWindows()

## 3. 中值滤波

import cv2 as cv

import numpy as np

from matplotlib import pyplot as plt

# 1 图像读取

img = cv.imread('./image/dogsp.jpeg')# 2 中值滤波

blur = cv.medianBlur(img,5)

cv.imshow('enlarge', up\_img)

cv.imshow('original', img)

cv.imshow('shrink', img\_1)

cv.waitKey(0)

五．直方图

## 5.1 直方图的计算和绘制

import cv2

import numpy as np

import matplotlib.pyplot as plt

img = cv2.imread('./lena.jpg')

hist=cv2.calcHist([img],[0],None,[256],[0,255])

print(hist)

plt.figure(figsize=(10,8))

plt.plot(hist)

plt.show()

cv2.waitKey(0)

cv2.destroyAllWindows()

import cv2

import numpy as np

import matplotlib.pyplot as plt

# 进行均衡化处理

img = cv2.imread('./lena.jpg')

gray=cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)

hist=cv2.calcHist([gray],[0],None,[256],[0,255]) #plt.plot(hist,lable='dark')

plt.plot(hist)

plt.show()

cv2.waitKey(0)

cv2.destroyAllWindows()

dst = cv2.equalizeHist(gray)

cv2.imshow('2',np.hstack((gray,dst)))

cv2.waitKey(0)

cv2.destroyAllWindows()

## 5.2掩膜的应用

import numpy as np

import cv2 as cv

from matplotlib import pyplot as plt

# 1. 直接以灰度图的方式读入

img = cv.imread('./image/cat.jpeg',0)

# 2. 创建蒙版

mask = np.zeros(img.shape[:2], np.uint8)

mask[400:650, 200:500] = 255

# 3.掩模 masked\_img = cv.bitwise\_and(img,img,mask = mask)

# 4. 统计掩膜后图像的灰度图

mask\_histr = cv.calcHist([img],[0],mask,[256],[1,256])

六．边缘检测

# 6.1Sobel检测算子

import cv2 as cv

import numpy as np

from matplotlib import pyplot as plt

# 1 读取图像 img = cv.imread('./image/horse.jpg',0)

# 2 计算Sobel卷积结果

x = cv.Sobel(img, cv.CV\_16S, 1, 0)

y = cv.Sobel(img, cv.CV\_16S, 0, 1)

# 3 将数据进行转换 Scale\_absX = cv.convertScaleAbs(x)

# convert 转换 scale 缩放 Scale\_absY = cv.convertScaleAbs(y)

# 4 结果合成 result = cv.addWeighted(Scale\_absX, 0.5, Scale\_absY, 0.5, 0)

# 6.2Canny边缘检测

import cv2 as cv

import numpy as np

from matplotlib import pyplot as plt

# 1 图像读取

img = cv.imread('./image/horse.jpg',0)

# 2 Canny边缘检测

lowThreshold = 0

max\_lowThreshold = 100

canny = cv.Canny(img, lowThreshold, max\_lowThreshold)