Opencv2知识点总结

2019年4月9日 9:43

一、高层GUI图形用户界面模块：highgui

1、HighGUI图形界面初步

1) 图像的载入、显示和输出到文件
   a. 图像的载入：imread()函数
   b. 图像的显示：imshow()函数
   c. 创建窗口：namedWindow()函数
   d. 输出图像至文件：imwrite()函数
2) 滑动条的创建和使用
   a. 创建滑动条：createTrackbar()函数
   b. 获取当前轨迹条的位置：getTrackbarPos()函数
   c. 为指定的窗口设置鼠标回调函数：SetMouseCallback()函数

二、核心功能模块：core组件

1、基础图像容器Mat数据结构

1) 各种数据结构
   a. Mat类的构造函数：Mat::Mat()函数
   b. Mat类的成员函数，可用于Mat类的初始化操作：Mat::Create()函数
   c. 点的数据结构：Point类
   d. 颜色的数据结构：Scalar类
   e. 尺寸的数据结构：Size类
   f. 矩阵的数据结构：Rect类
   g. 用于颜色空间转换：CvtColor()函数
2) core组件进阶——操作图像中的像素、图像混合、分离颜色通道、调节图像对比度和亮度、进行离散傅里叶变换，以及输入输出XML和YAML文件
   a. 计算两个数组（图像阵列）的加权和：addWeighted()函数
   b. 将一个多通道数组分离成几个单通道数组：split()函数
   c. 将多个数组组合合并成一个多通道的数组：merge()函数
   d. 对一维或二维浮点数数组进行正向或反向离散傅里叶变换：dft()函数
   e. 返回给定向量尺寸的傅里叶最优尺寸大小：getOptimalDFTSize()函数
   f. 扩充图像边界：copyMakeBorder()函数
   g. 计算二维矢量的赋值：magnitude()函数
   h. 计算每个数组元素绝对值的自然对数：log()函数
   i. 进行矩阵归一化：normalize()函数
   j. 进行文件操作的类：FileStorage类

对比度和亮度：g(x)=a*f(x)+b

其中，a为对比度，b为亮度，是对像素点的操作

三、图像处理模块：imgproc组件

1、线性和非线性的图像滤波、图像的几何变换、其他（Miscellaneous）图像转换、直方图相关、结构分析和形状描述、运动分析和对象跟踪、特征检测、目标检测

1) 图像处理（空间分量）
   a. 三种线性滤波
      - 方框滤波：boxFilter()函数
      - 均值滤波：blur()函数
      - 高斯滤波：GaussianBlur()函数
   b. 两种非线性滤波
      - 中值滤波：medianBlur()函数
      - 双边滤波：bilateralFilter()函数
   c. 七种图像处理形态学：
      - 腐蚀：dilate()函数
      - 膨胀：erode()函数
      - 开运算：morphologyEx(MORPH_OPEN)函数
      - 闭运算：morphologyEx(MORPH_CLOSE)函数
      - 形态学梯度：morphologyEx(MORPH_GRADIENT)函数
      - 顶帽：morphologyEx(MORPH_TOPHAT)函数
      - 黑帽：morphologyEx(MORPH_BLACKHAT)函数
   d. 漫水填充：floodFill()函数
   e. 图像缩放：pyrUp()函数
   f. 图像金字塔：pyrDown()函数
   g. 阈值化：
      - 对单通道数组应用固定阈值操作：Threshold()函数
      - 对矩阵采用自适应操作：adaptiveThreshold()函数
2) 图像变换（频谱分量）
   a. 边缘检测
      - 利用Canny算子来进行图像的边缘检测：Canny()函数
      - 使用扩展的Sobel算子计算一阶、二阶、三阶或混合图像差分：Sobel()函数
      - 计算出图像经过拉普拉斯变换后的结果：Laplacian()函数
      - 使用Scharr滤波器运算符计算x或y方向的图像差分：Scharr()函数
   b. 霍夫变换
      - 找出采用标准霍夫变换的二值图像中的直线：HoughLines()函数
      - 采用累积概率霍夫变换(PPHT)来找出二值图像中的直线：HoughLinesP()函数
      - 利用霍夫变换算法检测出灰度图中的圆：HoughCircles()函数
   c. 重映射
      - 根据指定的形式将源图像进行重映射几何变换：remap()函数
   d. 仿射变换

- 根据公式对图像做仿射变换：warpAffine()函数
- 计算二维旋转变换矩阵：getRotationMatrix2D()函数

e. 直方图均衡化
- 实现图像的直方图均衡化：equalizeHist()函数

3) 图像轮廓与图像分割修复

a. 查找并绘制轮廓
- 在二值图像中寻找轮廓：findContours()函数
- 在图像中绘制外部或内部轮廓：drawContours()函数

b. 寻找物体的凸包
- 寻找图像点集中的凸包：convexHull()函数
- 计算并返回指定点集最外面的矩形边界：BoundingRect()函数
- 寻找可旋转的最小面积的包围矩形：minAreaRect()函数
- 寻找最小面积的包围圆形：minEnclosingCircle()函数
- 用椭圆拟合二维点集：fitEllipse()函数

c. 使用多边形逼近物体
- 用指定精度逼近多边形曲线：approxPolyDP()函数

d. 认识图像的矩
- 计算多边形和光栅形状的最高达三阶的所有矩：moments()函数
- 计算整个轮廓或部分轮廓的面积：contourArea()函数
- 计算封闭轮廓的周长或曲线的长度：arcLength()函数
- 实现分水岭算法：watershed()函数

e. 图像的修补
- 进行图像修补、清楚划痕，去除不需要物体：inpaint()函数

4) 直方图与匹配

a. 直方图的计算与绘制
- 计算一个或多个阵列的直方图：calcHist()函数
- 在数值中找到全局最小值和最大值：minMaxLoc()函数

b. 直方图对比
- 对两幅直方图进行比较：compareHist()函数

c. 反向投影技术
- 计算直方图的反向投影：calcBackProject()函数
- 由输入参数赋值某通道到输出参数特定的通道中：mixChannels()函数

d. 模板匹配技术
- 匹配出和模板重叠的图像区域：matchTemplate()函数

四、2D功能框架模块：features2d组件，与其他组件配合使用

1、特征检测和描述、特征检测器通用接口、描述符提取器通用接口、描述符匹配器通用接口、通用描述符匹配器通用接口、关键点绘制函数和匹配功能绘制函数

1) 角点检测

a. Harris角点检测
   - 运行Harris角点检测算子进行角点检测：cornerHarris()函数

b. Shi-Tomasi角点检测
   - 结合Shi-Tomasi算子确定图像的强角点：goodFeaturesToTrack()函数

c. 亚像素级角点检测
   - 寻找亚像素角点位置：cornerSubPix()函数

2) 特征点的检测与匹配

a. SIFI算法
   - 进行暴力匹配相关的操作：BruteFroceMatcher类

b. SURF算法
   - SURF类、SurfFeatureDetector类、SurfDescriptorExtractor类：三者等价
   - 绘制关键点：drawKeypoints()函数
   - 绘制出相匹配的两个图像的关键点：drawMatches()函数
   - 用于表征特征点的信息：KeyPoint类
   - 实现FLANN特征匹配：FlannBasedMatch类
   - 从每个描述符查询集中找到最佳匹配：DescriptorMatcher::match()函数
   - 找到并返回源图像和目标图像之间的透视变换H：findHomography()函数
   - 进行向量透视矩阵变换：perspectiveTransform()函数

c. ORB算法
   - ORB类、ORBFeatureDetector类、OrbDescriptorExtractor类：