

OpenCV 方法及应用速查表

序 号	方 法	说 明	技术分类	序 号	方 法	说 明	技术分类
1	image = cv2.imread(filename, flags)	读取图像	基本操作	34	dst = cv2.GaussianBlur(src, ksize, sigmaX, sigmaY, borderType)	高斯滤波	平滑处理
2	cv2.imshow(winname, mat)	显示图像	基本操作	35	dst = cv2.bilateralFilter(src, d, sigmaColor, sigmaSpace, borderType)	双边滤波	平滑处理
3	cv2.destroyAllWindows()	销毁所有正在显示图像的窗口	基本操作	36	dst = cv2.erode(src, kernel, anchor, iterations, borderType, borderValue)	腐蚀	形态学操作
4	cv2.imwrite(filename, img)	保存图像	基本操作	37	dst = cv2.dilate(src, kernel, anchor, iterations, borderType, borderValue)	膨胀	形态学操作
5	dst = cv2.cvtColor(src, code)	转换色彩空间	图像基础	38	dst = cv2.morphologyEx(src, op, kernel, anchor, iterations, borderType, borderValue)	形态学运算	形态学操作
6	h, s, v = cv2.split(hsv_image)	拆分图像通道	图像基础	39	contours, hierarchy = cv2.findContours(image, mode, methode)	检测轮廓	轮廓检测
7	bgr = cv2.merge([b, g, r])	合并通道	图像基础	40	image = cv2.drawContours(image, contours, contourIdx, color, thickness, lineTypee, hierarchy, maxLevel, offse)	绘制轮廓	轮廓检测
8	img = numpy.zeros((height, width), np.uint8)	纯黑图像	图像基础	41	retval = cv2.boundingRect (array)	最小矩形包围框	轮廓检测
9	img = numpy.ones((height, width), np.uint8) * 255	纯白图像	图像基础	42	center, radius = cv2.minEnclosingCircle(points)	最小圆形包围框	轮廓检测
10	img = numpy.hstack((img1,img2))	水平拼接	图像基础	43	hull = cv2.convexHull(points, clockwise, returnPoints)	获取凸包	轮廓检测
11	img = numpy.vstack((img1,img2))	垂直拼接	图像基础	44	edges = cv2.Canny(image, threshold1, threshold2, apertureSize, L2gradient)	Canny 边缘检测	边缘检测
12	cv2.line(img, pt1, pt2, color, thickness)	绘制线段	绘图	45	lines = cv2.HoughLinesP(image, rho, theta, threshold, minLineLength, maxLineGap)	检测直线	霍夫变换
13	cv2.rectangle(img, pt1, pt2, color, thickness)	绘制矩形	绘图	46	circles = cv2.HoughCircles(image, method, dp, minDist, param1, param2, minRadius, maxRadius)	检测圆环	霍夫变换
14	cv2.circle(img, center, radius, color, thickness)	绘制圆形	绘图	47	result = cv2.matchTemplate(image, templ, method, mask)	模板匹配	模板匹配
15	cv2.polylines(img, pts, isClosed, color, thickness)	绘制多边形	绘图	48	minValue, max Value, minLoc, maxLoc = cv2.minMaxLoc(src, mask)	解析二维数组中的最大值、最小值以及这两个值对应的位置	模板匹配
16	cv2.putText(img, text, org, fontFace, fontScale, color, thickness, lineType, bottomLeftOrigin)	绘制文字	绘图	49	capture = cv2.VideoCapture(index)	获取摄像头	摄像头操作
17	retval, dst = cv2.threshold(src, thresh, maxval, type)	阈值处理	阈值处理	50	retval = cv2.VideoCapture.isOpened()	检验摄像头初始化是否成功	摄像头操作
18	dst = cv2.adaptiveThreshold(src, max Value, adaptiveMethod, thresholdType, blockSize, C)	自适应阈值	阈值处理	51	cv2.VideoCapture.read()	读取一帧	摄像头操作
19	dst = cv2.resize(src, dsize, fx, fy, interpolation)	缩放	几何变换	52	cv2.VideoCapture.release()	释放摄像头	摄像头操作
20	dst = cv2.flip(src, flipCode)	翻转	几何变换	53	video = cv2.VideoCapture(filename)	读取视频文件	视频文件操作
21	dst = cv2.warpAffine(src, M, dsize, flags, borderMode, borderValue)	仿射	几何变换	54	retval = cv2.VideoCapture.get(propId)	获取视频文件的属性	视频文件操作
22	M = cv2.getRotationMatrix2D(center, angle, scale)	计算旋转 M 矩阵	几何变换	55	<VideoWriter object> = cv2.VideoWriter(filename, fourcc, fps, frameSize)	创建视频输出流，保存视频文件	视频文件操作
23	M = cv2.getAffineTransform(src, dst)	计算仿射 M 矩阵	几何变换	56	cv2.VideoWriter.write(frame)	保存视频文件（简写方式）	视频文件操作
24	dst = cv2.warpPerspective(src, M, dsize, flags, borderMode, borderValue)	透视	几何变换	57	cv2.VideoWriter.release()	释放视频输出流	视频文件操作
25	M = cv2.getPerspectiveTransform(src, dst)	计算透视 M 矩阵	几何变换	58	<CascadeClassifier object> = cv2.CascadeClassifier(filename)	加载级联分类器	人脸识别
26	dst = cv2.add(src1, src2, mask, dtype)	图像加运算	图像运算	59	objects = cascade.detectMultiScale(image, scaleFactor, minNeighbors, flags, minSize, maxSize)	级联分类器对象识别图像	人脸识别
27	dst = cv2.bitwise_and(src1, src2, mask)	图像与运算	图像运算	60	recognizer = cv2.face.EigenFaceRecognizer_create(num_components, threshold)	创建 Eigenfaces 人脸识别器	人脸识别
28	dst = cv2.bitwise_or(src1, src2, mask)	图像或运算	图像运算	61	recognizer = cv2.face.FisherFaceRecognizer_create(num_components, threshold)	创建 Fisherfaces 人脸识别器	人脸识别
29	dst = cv2.bitwise_not(src, mask)	图像取反运算	图像运算	62	recognizer = cv2.face.LBPHFaceRecognizer_create(radius, neighbors, grid_x, grid_y, threshold)	创建 LBPH 人脸识别器	人脸识别
30	dst = cv2.bitwise_xor(src, mask)	图像异或运算	图像运算	63	recognizer.train(src, labels)	训练人脸识别器	人脸识别
31	dst = cv2.addWeighted(src1, alpha, src2, beta, gamma)	图像加权和运算	图像运算	64	label, confidence = recognizer.predict(src)	人脸识别器对图像进行识别	人脸识别
32	dst = cv2.blur(src, ksize, anchor, borderType)	均值滤波	平滑处理				
33	dst = cv2.medianBlur(src, ksize)	中值滤波	平滑处理				

Numpy 模块常用函数速查表

序 号	方 法	说 明	使用度	序 号	方 法	说 明	使用度
1	array(object[, dtype, copy, order, subok, ndmin])	创建一个数组	★★★★★	27	transpose(a[, axes])	排列数组的尺寸	★☆☆☆☆
2	asarray(a[, dtype, order])	将输入转换为数组	★★☆☆☆	28	concatenate((a1, a2, ...))	沿现有轴连接一系列数组。	★☆☆☆☆
3	copy(a[, order])	返回给定对象的数组副本	★★★★★	29	stack(arrays[, axis, out])	沿新轴连接一系列数组。	★☆☆☆☆
4	fromstring(string[, dtype, count, sep])	从字符串中的文本数据初始化的新一维数组	★☆☆☆☆	30	column_stack(tup)	将一维数组作为列堆叠到二维数组中。	★☆☆☆☆
5	loadtxt(fname[, dtype, comments, delimiter, ...])	从文本文件加载数据	★☆☆☆☆	31	dstack(tup)	沿深度方向（沿第三轴）按顺序堆叠数组。	★☆☆☆☆
6	empty(shape[, dtype, order])	返回给定形状和类型的新数组，而无需初始化条目	★☆☆☆☆	32	hstack(tup)	水平（按列）顺序堆叠数组。	★★★★★
7	empty_like(prototype[, dtype, order, subok, ...])	返回形状和类型与给定数组相同的新数组	★☆☆☆☆	33	vstack(tup)	垂直（行）按顺序堆叠数组。	★★★★★
8	eye(N[, M, k, dtype, order])	返回一个二维数组，对角线上有一个，其他地方为零	★☆☆☆☆	34	block(arrays)	从块的嵌套列表中组装一个nd数组。	★☆☆☆☆
9	identity(n[, dtype])	返回标识数组	★☆☆☆☆	35	split(ary, indices_or_sections[, axis])	将数组拆分为多个子数组，作为ary的视图。	★☆☆☆☆
10	ones(shape[, dtype, order])	返回给定形状和类型的新数组，并填充为1	★★★★★	36	array_split(ary, indices_or_sections[, axis])	将一个数组拆分为多个子数组。	★☆☆☆☆
11	ones_like(a[, dtype, order, subok, shape])	返回形状与类型与给定数组相同的数组	★☆☆☆☆	37	dsplit(ary, indices_or_sections)	沿第3轴（深度）将数组拆分为多个子数组。	★☆☆☆☆
12	zeros(shape[, dtype, order])	返回给定形状和类型的新数组，并用零填充	★★★★★	38	hsplit(ary, indices_or_sections)	水平（按列）将一个数组拆分为多个子数组。	★☆☆☆☆
13	zeros_like(a[, dtype, order, subok, shape])	返回形状与类型与给定数组相同的零数组。	★☆☆☆☆	39	vsplit(ary, indices_or_sections)	垂直（行）将数组拆分为多个子数组。	★☆☆☆☆
14	full(shape, fill_value[, dtype, order])	返回给定形状和类型的新数组，并用fill_value填充	★★★★★	40	delete(arr, obj[, axis])	返回一个新的数组，该数组具有沿删除的轴的子数组。	★☆☆☆☆
15	full_like(a, fill_value[, dtype, order, ...])	返回形状和类型与给定数组相同的完整数组	★☆☆☆☆	41	insert(arr, obj, values[, axis])	沿给定轴在给定索引之前插入值。	★☆☆☆☆
16	diag(v[, k])	提取对角线或构造对角线数组	★☆☆☆☆	42	append(arr, values[, axis])	将值附加到数组的末尾。	★★★★★
17	diagflat(v[, k])	使用展平的输入作为对角线创建二维数组	★☆☆☆☆	43	resize(a, new_shape)	返回具有指定形状的新数组。	★☆☆☆☆
18	tri(N[, M, k, dtype])	在给定对角线处及以下且在其他位置为零的数组	★☆☆☆☆	44	trim_zeros(filt[, trim])	修剪一维数组或序列中的前导和/或尾随零。	★☆☆☆☆
19	tril(m[, k])	数组的下三角	★☆☆☆☆	45	unique(ar[, return_index, return_inverse, ...])	查找数组的唯一元素。	★☆☆☆☆
20	triu(m[, k])	数组的上三角	★☆☆☆☆	46	flip(m[, axis])	沿给定轴颠倒数组中元素的顺序。	★☆☆☆☆
21	copyto(dst, src[, casting, where])	将值从一个数组复制到另一个数组，并根据需要进行广播	★☆☆☆☆	47	fliplr(m)	左右翻转数组。	★★★☆☆
22	reshape(a, newshape[, order])	在不更改数据的情况下为数组赋予新的形状。	★☆☆☆☆	48	flipud(m)	上下翻转阵列。	★★★☆☆
23	ravel(a[, order])	返回一个连续的扁平数组。	★☆☆☆☆	49	reshape(a, newshape[, order])	在不更改数据的情况下为数组赋予新的形状。	★☆☆☆☆
24	rollaxis(a, axis[, start])	向后滚动指定的轴，直到其位于给定的位置。	★☆☆☆☆	50	roll(a, shift[, axis])	沿给定轴滚动数组元素。	★☆☆☆☆
25	swapaxes(a, axis1, axis2)	互换数组的两个轴。	★☆☆☆☆	51	rot90(m[, k, axes])	在轴指定的平面中将阵列旋转90度。	★☆☆☆☆
26	ndarray.T	转置数组。	★★★★☆				

图形、视频及编码速查表

图像属性	
属 性 值	说 明
image_Color.shape	垂直像素，水平像素，通道数
image_Color.size	像素个数
image_Color.dtype	图像深度

视频文件属性	
属 性 值	说 明
cv2.CAP_PROP_POS_MSEC	视频文件播放时的当前位置（单位：ms）
cv2.CAP_PROP_POS_FRAMES	帧的索引，从0开始
cv2.CAP_PROP_POS_AVI_RATIO	视频文件的相对位置（0表示开始播放，1表示结束播放）
cv2.CAP_PROP_FRAME_WIDTH	视频文件的帧宽度
cv2.CAP_PROP_FRAME_HEIGHT	视频文件的帧高度
cv2.CAP_PROP_FPS	帧速率
cv2.CAP_PROP_FOURCC	用4个字符表示的视频编码格式
cv2.CAP_PROP_FRAME_COUNT	视频文件的帧数
cv2.CAP_PROP_FORMAT	retrieve()方法返回的Mat对象的格式
cv2.CAP_PROP_MODE	指示当前捕获模式的后端专用的值
cv2.CAP_PROP_CONVERT_RGB	指示是否应将图像转换为RGB

常用的视频编码格式		
fourcc的值	视频编码格式	文件扩展名
cv2.VideoWriter_fourcc('I', '4', '2', '0')	未压缩的YUV颜色编码格式，兼容性好，但文件较大	.avi
cv2.VideoWriter_fourcc('P', 'I', 'M', 'I')	MPEG-1 编码格式	.avi
cv2.VideoWriter_fourcc('X', 'V', 'I', 'D')	MPEG-4 编码格式，视频文件的大小为平均值	.avi
cv2.VideoWriter_fourcc('T', 'H', 'E', 'O')	Ogg Vorbis 编码格式，兼容性差	.ogv
cv2.VideoWriter_fourcc('F', 'L', 'V', 'I')	Flash 视频编码格式	.flv

OpenCV 常量值含义表

色彩空间转换常量	
常 量 值	说 明
cv2.COLOR_BGR2GRAY	从 BGR 色彩空间转换到 GRAY 色彩空间
cv2.COLOR_RGB2GRAY	从 RGB 色彩空间转换到 GRAY 色彩空间
cv2.COLOR_BGR2HSV	从 BGR 色彩空间转换到 HSV 色彩空间
cv2.COLOR_RGB2HSV	从 RGB 色彩空间转换到 HSV 色彩空间

字体常量	
题 目 名 称	问 题 描 述
cv2.FONT_HERSHEY_SIMPLEX	正常大小的 sans-serif 字体
cv2.FONT_HERSHEY_PLAIN	小号的 sans-serif 字体
cv2.FONT_HERSHEY_DUPLEX	正常大小的 sans-serif 字体（比 FONT_HERSHEY_SIMPLEX 字体样式更复杂）
cv2.FONT_HERSHEY_COMPLEX	正常大小的 serif 字体
cv2.FONT_HERSHEY_TRIPLEX	正常大小的 serif 字体（比 FONT_HERSHEY_COMPLEX 字体样式更复杂）
cv2.FONT_HERSHEY_COMPLEX_SMALL	FONT_HERSHEY_COMPLEX 字体样式的的简化版
cv2.FONT_HERSHEY_SCRIPT_SIMPLEX	手写风格的字体
cv2.FONT_HERSHEY_SCRIPT_COMPLEX	FONT_HERSHEY_SCRIPT_SIMPLEX 字体样式的进阶版
cv2.FONT_ITALIC	斜体

阈值处理类型	
常 量 值	说 明
cv2.THRESH_BINARY	二值化阈值处理
cv2.THRESH_BINARY_INV	反二值化阈值处理
cv2.THRESH_TOZERO	低于阈值零处理
cv2.THRESH_TOZERO_INV	超出阈值零处理
cv2.THRESH_TRUNC	截断阈值处理

形态学运算的操作类型	
常 量 值	说 明
cv2.MORPH_ERODE	腐蚀操作
cv2.MORPH_DILATE	膨胀操作
cv2.MORPH_OPEN	开运算，先腐蚀后膨胀
cv2.MORPH_CLOSE	闭运算，先膨胀后腐蚀
cv2.MORPH_GRADIENT	梯度运算，膨胀图减腐蚀图，可以得出简易的轮廓
cv2.MORPH_TOPHAT	顶帽运算，原始图像减开运算图像
cv2.MORPH_BLACKHAT	黑帽运算，闭运算图像减原始图像

自适应阈值采用的算法	
常 量 值	说 明
cv2.ADAPTIVE_THRESH_MEAN_C	对一个正方形区域内的所有像素平均加权
cv2.ADAPTIVE_THRESH_GAUSSIAN_C	根据高斯函数按照像素与中心点的距离对一个正方形区域内的所有像素进行加权计算

OpenCV自带的级联分类器及其功能	
常 量 值	说 明
haarcascade_eye.xml	眼睛检测
haarcascade_eye_tree_eyeglasses.xml	眼镜检测
haarcascade_frontalcatface.xml	正面猫脸检测
haarcascade_frontalface_default.xml	正面人脸检测
haarcascade_fullbody.xml	身形检测
haarcascade_lefteye_2splits.xml	左眼检测
haarcascade_lowerbody.xml	下半身检测
haarcascade_profileface.xml	侧面人脸检测
haarcascade_righteye_2splits.xml	右眼检测
haarcascade_russian_plate_number.xml	车牌检测
haarcascade_smile.xml	笑容检测
haarcascade_upperbody.xml	上半身检测

轮廓检测模式	
常 量 值	说 明
cv2.RETR_EXTERNAL	只检测外轮廓
cv2.RETR_LIST	检测所有轮廓，但不建立层次关系
cv2.RETR_CCOMP	检测所有轮廓，并建立两级层次关系
cv2.RETR_TREE	检测所有轮廓，并建立树状结构的层次关系

模板匹配采用的方法	
常 量 值	说 明
cv2.TM_SQDIFF	差值平方和匹配，也叫平方差匹配。可以理解为差异程度。匹配程度越高，计算结果越小。完全匹配的结果为0
cv2.TM_SQDIFF_NORMED	标准差值平方和匹配，也叫标准平方差匹配，规则同上
cv2.TM_CCORR	相关匹配。可以理解为相似程度，匹配程度越高，计算结果越大
cv2.TM_CCORR_NORMED	标准相关匹配，规则同上
cv2.TM_CCOEFF	相关系数匹配，也属于相似程度，计算结果为-1~1之间的浮点数，1表示完全匹配，0表示毫无关系，-1表示两张图片亮度刚好相反
cv2.TM_CCOEFF_NORMED	标准相关系数匹配，规则同上
cv2.MORPH_BLACKHAT	黑帽运算，闭运算图像减原始图像

轮廓检测采用的方法	
常 量 值	说 明
cv2.CHAIN_APPROX_NONE	储存轮廓上的所有点
cv2.CHAIN_APPROX_SIMPLE	只保存水平、垂直或对角线轮廓的端点
cv2.CHAIN_APPROX_TC89_L1	Ten-Chinl 近似算法中的一种
cv2.CHAIN_APPROX_TC89_KCOS	Ten-Chinl 近似算法中的一种