

使用说明书

double CharacterRecognition(char* TargetImage, char* TemplateFileGroup[])	<p>字符匹配，支持 BMP 图像，返回值是目标图像匹配到的模板文件的序号，如返回值是 2 则说明图像与序号为 2(序号从零开始)的模板匹配。</p> <p>参 考： TemplateFileGroup[]={ "0.txt", "1.txt", "2.txt", "3.txt", "4.txt", "5.txt", "6.txt", "7.txt", "8.txt", "9.txt" };</p>
double CharacterRecognition1(char* TargetImage, char* TemplateFileGroup[])	<p>字符匹配，支持 BMP 图像，返回值是目标图像匹配到的模板文件的序号，如返回值是 2 则说明图像与序号为 2(序号从零开始)的模板匹配。</p> <p>参 考： TemplateFileGroup[]={ "0.txt", "1.txt", "2.txt", "3.txt", "4.txt", "5.txt", "6.txt", "7.txt", "8.txt", "9.txt" };</p>
void CodeEncoding(std::string input, char* output, int width, int height, int margin, int eccLevel, int stride_bytes, int comp, int a)	<p>二维码和条形码编码。input 是要编码的字符串，output 是生成的二维码图像文件名。</p> <p>margin:条形码周围的边距 ecc: 纠错级别，[0-8] a=1: AZTEC a=2: CODABAR a=3: CODE_39 a=4: CODE_93 a=5: CODE_128 a=6: DATA_MATRIX a=7: EAN_8 a=8: EAN_13 a=9: ITF a=10: MAXICODE a=11: PDF_417 a=12: QR_CODE a=13: RSS_14 a=14: RSS_EXPANDED a=15: UPC_A a=16: UPC_E a=17: UPC_EAN_EXTENSION 参考: margin=10, eccLevel=-1, stride_bytes=0, comp=1。</p>
std::string CodeDecoding(char*	二维码和条形码解码。input 是输入

input, int req_comp, int a)

的二维码图像文件名，返回解码结果。

a=1: Lum

a=2: RGB

a=3: BGR

a=4: RGBX

a=5: XRGB

a=6: BGRX

a=7: XBGR

参考: req_comp=4, a=4。