

Manuel d'utilisation

double CharacterRecognition(char* TargetImage, char* TemplateFileGroup[])	Correspondance de caractères, prise en charge des images BMP, la valeur de retour est le numéro de série du fichier modèle auquel l'image cible correspond, si la valeur de retour est 2, l'image correspond au modèle dont le numéro de série est 2 (le numéro de série commence à zéro). Référence : TemplateFileGroup[]={ "0.txt", "1.txt", "2.txt", "3.txt", "4.txt", "5.txt", "6.txt", "7.txt", "8.txt", "9.txt" };
double CharacterRecognition1(char* TargetImage, char* TemplateFileGroup[])	Correspondance de caractères, prise en charge des images BMP, la valeur de retour est le numéro de série du fichier modèle auquel l'image cible correspond, si la valeur de retour est 2, l'image correspond au modèle dont le numéro de série est 2 (le numéro de série commence à zéro). Référence : TemplateFileGroup[]={ "0.txt", "1.txt", "2.txt", "3.txt", "4.txt", "5.txt", "6.txt", "7.txt", "8.txt", "9.txt" };
void CodeEncoding(std::string input, char* output, int width, int height, int margin, int eccLevel, int stride_bytes, int comp, int a)	Codes 2D et codes à barres. Input est la chaîne à encoder et Output est le nom du fichier image du Code QR généré. margin: marges autour du Code à barres eccLevel: niveau de correction d'erreur, [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 Référence: margin=10, eccLevel=-1, stride_bytes=0, comp=1.
std::string CodeDecoding(char* input, int req_comp, int a)	Décodage de codes QR et de codes à barres. Input est le nom du fichier image du Code QR entré qui renvoie le résultat du décodage. a=1: Lum a=2: RGB a=3: BGR a=4: RGBX a=5: XRGB a=6: BGRX a=7: XBGR Référence: req_comp=4, a=4.