Content

void calcHist…………….…………. 1

void namedWindow……...…… 2

int createTrackbar……….…..…. 3

void LUT………………………...….. 4

void copyMakeBorder…...…… 5

int borderInterpolate..…….…. 6

void flip………………………………. 7

void filter2D……………………….. 8

void calcHist( const Mat\* images, int nimages,

const int\* channels, InputArray mask,

OutputArray hist, int dims, const int\* histSize,

const float\*\* ranges, bool uniform = true, bool accumulate = false );

函數功能:

建立直方圖

參數說明:

const Mat\* images: 原輸入(圖片)陣列

int nimages: 原輸入陣列元素個數(指幾張圖)

const int\* channels: 用來計算直方圖的通道維數陣列, [0]:灰階, [0, 1, 2]:RGB三色

InputArray mask: 若非空(Mat()), 則必須與images的大小相等, 且僅計算mask

中非零的部分

OutputArray hist: 輸出直方圖

int dims: 直方圖維度

const int\* histSize: 指出直方圖陣列每一維bin大小的陣列

const float\*\* ranges: 用來指出直方圖每一維的每個bin上下界範圍陣列的陣列

bool uniform = true: 直方圖是否均勻

bool accumulate = false: 是否累加

void namedWindow(const string& winname,

int flags=WINDOW\_AUTOSIZE )

函數功能:

建立一個視窗

參數說明:

const string& winname: 視窗名稱, 可當成視窗ID使用

int flags=WINDOW\_AUTOSIZE: 常用的2個設定值: WINDOW\_NORMAL:使用

者可自行縮放視窗大小, WINDOW\_AUTOSIZE: 自動調整成適合螢幕大

小, 不能自行縮放

int createTrackbar (conststring& trackbarname,

conststring& winname,

int\* value, int count,

TrackbarCallback onChange=0, void\* userdata=0);

函數功能:

建立一個滑桿

參數說明:

conststring& trackbarname: 滑桿名稱, 用來代表我們創建的這個滑桿

conststring& winname: 視窗名稱, 表示這個滑桿將會附到哪個視窗上, 即

對應創建視窗namedWindow()時所設的某個視窗名稱

int\* value: 一個指向整數的指標, 表示滑桿位置

int count: 滑桿位置的最大值

TrackbarCallback onChange=0: 這是一個指向回調函數的指針, 預設值

onchang=0,每次滑桿改變位置時, 這個函數都會進行回調, 並且這個函數原

型必須為void XXXX(int , void\*), 其中第一個參數為滑桿位置, 第二個參數

為用戶數據(看下一個 參數), 如果回調是NULL指針, 表示沒有回調函數

的調用, 僅第3個參數value有變化

void\* userdata=0: void\*類型的userdata, 預設值=0, 這個參數是用戶傳

給調用函數的數據, 用來處理滑桿事件, 如果第3個參數是全局變數, 可

以不用管這個參數

void LUT(InputArray src,

InputArray lut,

OutputArray dst, int interpolation=0 )

函數功能:

完成表單對應關係

參數說明:

InputArray src: 輸入圖片

InputArray lut: 有256個元素的look-up table, 就算圖片是多通道, 此

table只需要單通道(意思是會套用到全通道), 亦或是跟輸入圖片相同通

道

OutputArray dst: 目標圖片

void copyMakeBorder(InputArray src,

OutputArray dst,

int top, int bottom, int left, int right, int borderType, const Scalar& value=Scalar() )

函數功能:

將圖片擴大以利處理邊界, 同時調用cv::borderInterpolate

參數說明:

InputArray src: 輸入圖片

OutputArray dst: 輸出圖片

int top, int bottom, int left, int right: 表示想要在各方位擴大多少個

pixel, 可為0

int borderType: 插入方式, 在下一頁說明

const Scalar& value=Scalar(): 若是borderType==BORDER\_CONSTANT, 則輸入BGR

顏色(Scalar(B, G, R))

int borderInterpolate(int p, int len, int borderType)

函數功能:

計算擴充的像素對應原圖哪個座標的像素

參數說明:

int p: 該擴充圖的像素在原圖的座標, 若是對原圖左邊或上面擴充, 則p為負數

int len: 擴充像素對應原圖的長度(左右對應cols , 上下對應rows)

int borderType:

|  |  |
| --- | --- |
| BORDER\_CONSTANT | iiiiii|abcdefgh|iiiiiii |
| BORDER\_REPLICATE | aaaaaa|abcdefgh|hhhhhhh |
| BORDER\_REFLECT | fedcba|abcdefgh|hgfedcb |
| BORDER\_WRAP | cdefgh|abcdefgh|abcdefg |
| BORDER\_REFLECT\_101 | gfedcb|abcdefgh|gfedcba |
| BORDER\_TRANSPARENT | uvwxyz|abcdefgh|ijklmno |
| BORDER\_DEFAULT | BORDER\_REFLECT\_101 |

void flip(InputArray src,

OutputArray dst,

int flipCode)

函數功能:

翻轉矩陣

參數說明:

InputArray src: 輸入矩陣

OutputArray dst: 輸出矩陣

int flipCode: flipCode=0, 水平翻轉, flipCode>0, 垂直翻轉, flipCode<0, 轉180度

void filter2D(InputArray src,

OutputArray dst,

int ddepth,

InputArray kernel,

Point anchor=Point(-1,-1),

double delta=0,

int borderType=BORDER\_DEFAULT )

函數功能:

利用核對圖片進行卷積

參數說明:

InputArray src: 輸入圖片

OutputArray dst: 輸出圖片, 圖片大小和通道要與src相同

int ddepth: 待修正

InputArray kernel: 卷積核

Point anchor=Point(-1,-1): 錨點, 表示從核的錨點開始與圖片做卷積, 若核的長寬

皆為奇數, 則Point(-1, -1)等同為核中心

double delta=0: 在過濾後的像素存入dst之前可以加上這個選填的delta值

int borderType=BORDER\_DEFAULT: 看borderInterpolate