




基于opencv的隐写术

原创

断水客  于 2020-11-07 22:50:48 发布  105  收藏 1

分类专栏: [OpenCV](#) 文章标签: [opencv c++](#) [隐写术](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/qq_33904382/article/details/109553705

版权



[OpenCV 专栏收录该内容](#)

7 篇文章 0 订阅

订阅专栏

基于opencv的隐写术

基本思路:

- A、以无符号char类型读取二进制文件, 与图片中的红色channel数据求平均, 得到混合后的图片。
- B、逆向可以得到嵌入图片的信息。

自建类:

```

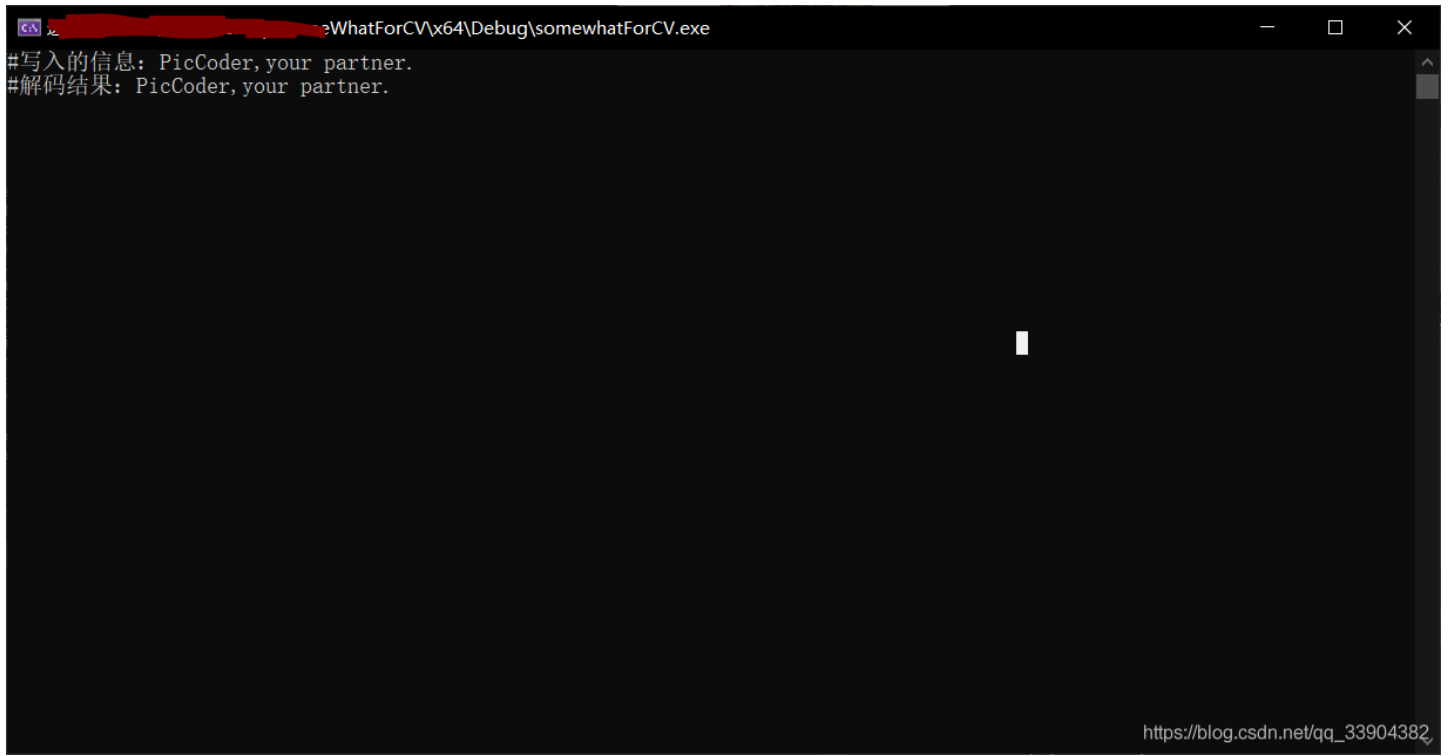
#pragma once

#include<iostream>
#include <string>
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
using namespace cv;
class PicCoder
{
public:
    inline PicCoder();
    inline ~PicCoder();
    /* 将数据和图片的源矩阵关联到类私有成员_srcInfo和_srcPic */
    inline void GetSrcMat_Code(Mat& info,Mat& pic) {
        _srcInfo = info;
        _srcPic = pic;
    };
    inline void GetSrcMat_DeCode(Mat& pic, Mat& mixed) {
        _srcMixed = mixed;
        _srcPic = pic;
    };
    /* 将指定的info和pic数据合并 */
    inline void AddInfo2Pic(const Mat& info,const Mat& pic,Mat& dst) {
        addWeighted(pic, 0.5, info, 0.5, 0.0, dst);
    };
    /* 合并数据并返回合并后的矩阵 */
    inline Mat GetCodedMat(Mat& info, Mat& pic) {
        GetSrcMat_Code(info,pic);
        /* 构造与图片等大的矩阵 */
        Mat dst(_srcPic.size(),_srcPic.type());
        /* 合并数据 */
        AddInfo2Pic(_srcInfo,_srcPic,dst);
        return dst;
    };
    inline void GetInfoFromMixed(const Mat& pic, const Mat& mixed, Mat& dst) {
        addWeighted(pic, -0.5, mixed, 1, 0.0, dst);
        addWeighted(dst, 1, dst, 1, 0.0, dst);
    };
    /* 利用原始图片从处理过的图片中解码信息 返回包含信息的矩阵 */
    inline Mat GetDecodedMat(Mat& pic,Mat& mixed) {
        GetSrcMat_DeCode(pic,mixed);
        /* 构造与图片等大的矩阵 */
        Mat dst(_srcPic.size(), _srcPic.type());
        /* 取出数据 */
        GetInfoFromMixed(_srcPic,_srcMixed,dst);
        return dst;
    };

private:
    //似乎并不需要预备空间去存储数据, 只需要动态处理并返回即可

    Mat _srcInfo;
    Mat _srcPic;
    Mat _srcMixed;
};

```



The screenshot shows a Windows command prompt window with the title bar text: "WhatForCV\x64\Debug\somewhatForCV.exe". The window content displays two lines of text: "#写入的信息: PicCoder, your partner." and "#解码结果: PicCoder, your partner." A white cursor is visible in the center of the window. In the bottom right corner of the window, there is a URL: https://blog.csdn.net/qq_33904382.

example:

```

/* 加密过程 */
/* 读入用于嵌入信息的图片 */
Mat imga = imread("D:\\pic.jpg");
try
{
/* 信息文件 */
FILE* infoFile = fopen("D:\\info.txt", "rb");
uchar buffer[1024] = { 0 };
uchar inforeadbuffer[1024] = { 0 };
size_t readsize = fread(buffer, sizeof(uchar), sizeof(buffer), infoFile);
cout << "#写入的信息: " << buffer << endl;
Mat infoSrcMat = Mat(imga.size(), imga.type());
for (int i = 0; i < imga.rows - 1; i++) {
    for (int j = 0; j < imga.cols - 1; j++) {
        if ((i + j) < readsize)
            infoSrcMat.at<Vec3b>(i, j)[0] = buffer[i + j];
        else {
            infoSrcMat.at<Vec3b>(i, j)[0] = imga.at<Vec3b>(i, j)[0];
        }
        infoSrcMat.at<Vec3b>(i, j)[1] = imga.at<Vec3b>(i, j)[1];
        infoSrcMat.at<Vec3b>(i, j)[2] = imga.at<Vec3b>(i, j)[2];
    }
}
/* 解密过程 */
PicCoder pcx;
Mat mixedMat = pcx.GetCodedMat(infoSrcMat, imga);
imshow("测试piccoder的编码信息结果", mixedMat);
waitKey(3000);
Mat infoReadMat = pcx.GetDecodedMat(imga, mixedMat);
for (int i = 0; i < imga.rows - 1; i++) {
    for (int j = 0; j < imga.cols - 1; j++) {
        if ((i + j) < readsize)
            inforeadbuffer[i + j] = infoSrcMat.at<Vec3b>(i, j)[0];
    }
}
cout << "#解码结果: " << inforeadbuffer << endl;
}
catch (const std::exception&)
{
;
}

waitKey(0);

```

效果:

可以看到左上角有一个泛黄的区域，就是嵌入数据的区域

