

BÁO CÁO BÀI TẬP PHÁT HIỆN TIỀN BẮN

(Ngày 23/3)

Môn lập trình ứng dụng

Họ và tên: Nguyễn Hoàng Đức

MSSV: 20021518

1. Ý tưởng

Tìm ra các điểm bắn trên tờ tiền bằng cách trừ 2 ảnh (ảnh gốc và ảnh bị bắn) cho nhau do ảnh trong thư viện OpenCV được lưu trong một ma trận các điểm ảnh, do vậy có thể lấy từng điểm ảnh của mỗi ảnh trừ đi cho nhau để tìm điểm khác biệt. Khoanh vùng những điểm đánh dấu (chỗ bị bắn) vào ảnh bị bắn.

2. Triển khai

Bước 1: tải lên 2 ảnh

```
Mat real_money = imread("D:/Desktop/real.jpg", IMREAD_COLOR);  
Mat fake_money = imread("D:/Desktop/fake.jpg", IMREAD_COLOR);
```

Bước 2: áp dụng GaussianFilter và điều chỉnh độ sáng

```
GaussianFilter(fake_money);  
  
if (Brightness_Adjustment(real_money, fake_money)) {  
    fake_money = scaleHistogram(fake_money, (double)getMaxPixel(real_money) /  
getMaxPixel(fake_money));  
}
```

Code GaussianFilter:

```
void GaussianFilter(Mat image) {  
    int rows = image.rows;  
    int cols = image.cols;  
    for (int i = 0; i < rows; i++)  
    {  
        Vec3b* ptr = image.ptr<Vec3b>(i);  
        for (int j = 0; j < cols; j++)  
        {  
            Vec3b pixel = ptr[j];  
        }  
    }  
    //imshow("Truoc Gaussian Filter", image);  
    Mat image_Gauss = image.clone();  
    GaussianBlur(image, image_Gauss, Size(9, 9), 0, 0);  
    for (int i = 0; i < rows; i++)  
    {  
        Vec3b* ptr = image_Gauss.ptr<Vec3b>(i);  
        for (int j = 0; j < cols; j++)
```

```

        {
            Vec3b pixel = ptr[j];
        }
    }
    //imshow("Sau Gaussian Filter", image_Gauss);
}

```

Code HistogramFilter:

```

bool RedDevil2::Brightness_Adjustment(Mat templ, Mat sample)
{
    int* template_histogram = getHistogramCount(templ);
    int* sample_histogram = getHistogramCount(sample);

    int count = 0;

    for (int i = 5; i < HISTOGRAM - 5; i++) {
        if (abs(template_histogram[i] - sample_histogram[i]) >= 0.02 *
template_histogram[i]) count++;
    }

    return count >= 200;
}

```

Bước 3: Trừ 2 ảnh

```

Mat diff = real_money - fake_money;
diff = RedDevil::medianFilter(diff);
threshold(diff, diff, 70, 255, THRESH_BINARY);

imshow("real_money", real_money);
imshow("fake_momey", fake_money);
imshow("result", diff);

```

code MedianFilter:

```

void RedDevil::insertionSort(int window[])
{
    int temp, i, j;
    for (i = 0; i < 9; i++) {
        temp = window[i];
        for (j = i - 1; j >= 0 && temp < window[j]; j--)
        {
            window[j + 1] = window[j];
        }
        window[j + 1] = temp;
    }
}

Mat RedDevil::medianFilter(Mat src)
{
    int window[9];
    Mat dst = src.clone();

    for (int y = 0; y < src.rows; y++)
        for (int x = 0; x < src.cols; x++)
            dst.at<uchar>(y, x) = 0.0;

    for (int y = 1; y < src.rows - 1; y++)
        for (int x = 1; x < src.cols - 1; x++)

```

```

    {
        window[0] = src.at<uchar>(y - 1, x - 1);
        window[1] = src.at<uchar>(y, x - 1);
        window[2] = src.at<uchar>(y + 1, x - 1);
        window[3] = src.at<uchar>(y - 1, x);
        window[4] = src.at<uchar>(y, x);
        window[5] = src.at<uchar>(y + 1, x);
        window[6] = src.at<uchar>(y - 1, x + 1);
        window[7] = src.at<uchar>(y, x + 1);
        window[8] = src.at<uchar>(y + 1, x + 1);
        insertionSort(window);
        dst.at<uchar>(y, x) = window[4];
    }
    return dst;
}

```

Bước 4: khâu vùng bản

```

if (Torn_detection(torn_money)) {
    Mat torn_money_clone = torn_money.clone();
    threshold(torn_money_clone, torn_money_clone, 250, 255, THRESH_BINARY_INV);

    threshold(torn_money, torn_money, 250, 0, THRESH_TOZERO_INV);
    Mat missingPieces = real_money - torn_money_clone;

    Mat restore = torn_money + missingPieces;
    restore = RedDevil::medianFilter(restore);
    restore = RedDevil::USM(restore, 1, 0.5, 5);
    imshow("", restore);
}

```

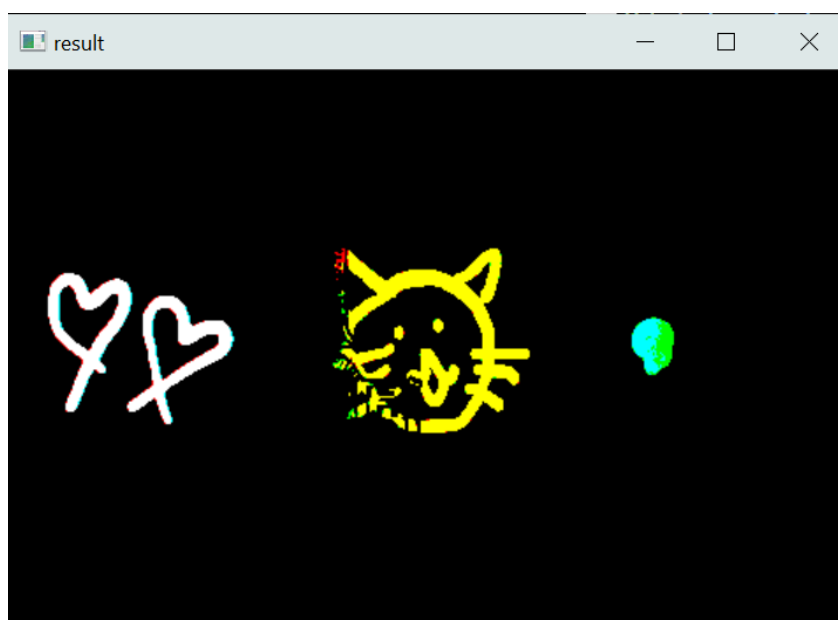
3. Kết quả



Hình 1: Tiền gốc



Hình 2: Tiền bị bẩn



Hình 3: Kết quả thu được sau khi trừ 2 tiền