

Visão por Computador 2016-17, Guia Prático N.º 2

Rui Oliveira, Tomás Rodrigues
 DETI, Universidade de Aveiro
 Aveiro, Portugal
 {ruipedrooliveira, tomasrodrigues}@ua.pt

Resumo –

Pretende-se através deste relatório expor sob forma escrita, o nosso desempenho e objetivos alcançados na aula prática n.º 2 da unidade curricular de Visão por Computador do Mestrado Integrado de Engenharia de Computadores e Telemática.

Neste relatório pretendemos explicar as soluções por nós encontradas para a resolução dos diferentes problemas propostos.

Palavras chave – visão, computador, imagem digital, opencv, c++,

I. REPOSITÓRIO: CÓDIGO FONTE

Todas as soluções dos problemas propostos estão disponíveis através do seguinte repositório (gitHub) criado para o efeito.

<http://github.com/toomyy94/CV1617-68779-68129>

A resolução dos problemas do presente guia encontram-se na pasta aula2.

II. PROBLEMAS PROPOSTOS

A. Problema #1 a #3

- Sistema operativo: Linux
- Instalação openCV: <https://github.com/jayrambhia/Install-OpenCV>
- IDE utilizado: Code::Blocks ¹

B. Problema #4

B.1 Enunciado

Implement, compile and test the OpenCV example “Load, Modify, and Save an Image”.

B.2 Resolução e principais conclusões

- Carregar imagem: usar método `imread()` especificando o tipo pretendido
- Aplicar modificação a uma imagem: foram utilizados os métodos `cvtColor` e `threshold`.
- Guardar imagem: foi usado o método `imwrite` especificando o caminho de destino.

Listing 1: Proposta de resolução do exercício 4

```
Mat img = imread("../images/lena.jpg",
                 CV_LOAD_IMAGE_COLOR);

if(img.empty())
    return -1;
cvtColor( img, img, CV_BGR2GRAY );
threshold(img, img, 100, 255, THRESH_BINARY);
imwrite( "../images/lena_bin.jpg", img );
imshow("lena_bin", img);
```



Figura 1: Imagem original lena.jpg

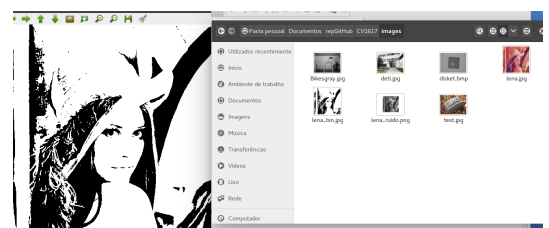


Figura 2: Resultado após execução do problema 4

Alternativas também aplicadas neste problema:

Carregar imagem, aplicar uma conversão para escala de cinza e guardar a imagem.

```
cvtColor( img, gray_image, CV_BGR2GRAY );
```

Carregar imagem, aplicar uma conversão para escala de YUV² e guardar a imagem.

```
cvtColor( img, yuv_image, CV_BGR2YUV );
```

¹<http://www.codeblocks.org/downloads>

²<https://en.wikipedia.org/wiki/YUV>

C. Problema #5

C.1 Enunciado

Implement, compile and test the OpenCV example 'Adding two images using OpenCV'

C.2 Resolução e principais conclusões

- Foi utilizado o método `addWeighted()` em que temos os seguintes argumentos:
 - `src1:..`
 - `alpha:..`
 - `src2:..`
 - `beta:..`
 - `dst:..`

Listing 2: Proposta de resolução do exercício 5

```
double alpha = 0.5; double beta; double input;
Mat src1, src2, dst;
namedWindow("Linear Blend", 1);
beta = ( 1.0 - alpha );
addWeighted( src1, alpha, src2, beta, 0.0,
             dst);
imshow( "Linear Blend", dst );
waitKey(0);
```

Com alpha igual 0.5 o resultado obtido aplicado à imagem da Figura 1 é o seguinte:

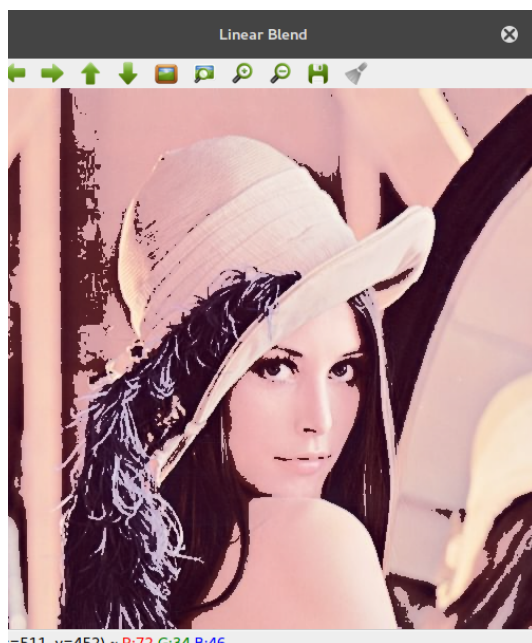


Figura 3: Resultado após execução do problema 5

D. Problema #6

D.1 Enunciado

Implement, compile and test the OpenCV example 'Changing the contrast and brightness of an image'.

D.2 Resolução e principais conclusões

Listing 3: Proposta de resolução do exercício 6

```
double alpha; int beta;
Mat img = imread("../images/lena.jpg",
                  CV_LOAD_IMAGE_COLOR);
Mat new_image = Mat::zeros( img.size(),
                             img.type() );

for( int y = 0; y < img.rows; y++ ){
    for( int x = 0; x < img.cols; x++ ){
        for( int c = 0; c < 3; c++ ){
            new_image.at<Vec3b>(y,x)[c] =
                saturate_cast<uchar>( alpha*(
                    img.at<Vec3b>(y,x)[c] ) + beta );
        }
    }
}
```

Com alpha igual 2.0 e beta igual a 60% o resultado obtido é o seguinte:

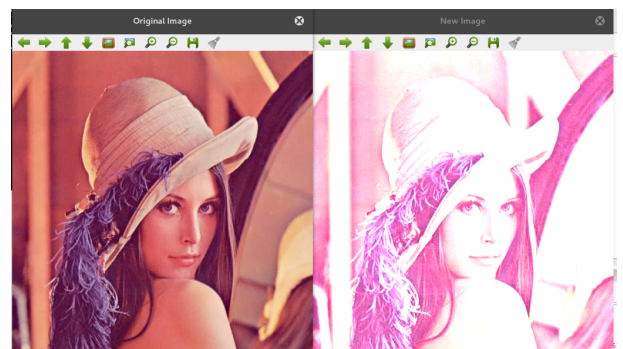


Figura 4: Resultado após execução do problema 6.

III. PRINCIPAIS CONCLUSÕES

Seguindo alguns templates de implementação e código que já usávamos da disciplina de Computação Visual executamos os primeiros exercícios sem problemas. No exercício 6 foi onde tivemos mais dificuldades mas com a ajuda dos tutoriais do opencv essa dificuldade foi ultrapassada com sucesso, sendo a usagem do `Vec_` uma forma nova para nós de resolvermos agora eventuais problemas futuros que apareçam.

REFERÊNCIAS

- [1] Neves, A. J. R.; Dias, P. Slides teóricos Visão por Computador - Aula 2 (2016)
- [2] OpenCV. OpenCV Documentation. Web. 24 Outubro 2016.