

# Worksheet2\_pdfversion

Creation date: 2024-11-08 15:54

Modification date: Friday 8th November 2024 15:54:10

PDF Ref: [Worksheet 2 - Espaços de cor e histogramas.pdf](#)

## Section 1 - Espaços de cor

**1. Construa uma função que crie imagens grayscale de cada canal a partir de uma imagem RGB. A função deve entrar como parâmetro o canal a processar. Mostre as imagens resultantes.**

Código:

```
1 void exercise1_ws2(int channel) {
2     // read the image
3     Mat image_extract = imread("J:/MEEC/PIVC/Projects/Project_PIVC_Worksheets/retina.tif");
4     // greyscale reference image / CV_8UC1 -> 1 channel
5     Mat grayImage(image_extract.rows, image_extract.cols, CV_8UC1, Scalar(0));
6     // go through all the pixels in the image / loop rows and columns
7     for (int r = 0; r < image_extract.rows; r++) {
8         for (int c = 0; c < image_extract.cols; c++) {
9             Vec3b pixel_value = image_extract.at<Vec3b>(r, c); // extract the values from the picture
10            // transfer the values to the reference image according to the index of the user chosen
11            channel
12            grayImage.at<uchar>(r, c) = pixel_value[channel];
13        }
14    }
15    imshow("Image Greyscale", grayImage); // show the image
16    waitKey(0); // key 0 to close
17    destroyAllWindows(); // destroy the window
18 }
```

Resultados:

- O resultado é uma imagem em greyscale com apenas o canal escolhido pelo utilizador.
- Inicialmente a imagem é percorrida pixel a pixel e são extraídos os valores dos pixels para cada matriz de cor. Os seus valores são guardados numa variável do tipo Vec3b.
- Por fim, apenas os pixels correspondentes ao canal escolhido são "transferidos" para a imagem greyscale de referência criada com o tipo CV\_8UC1 (tipo 1 canal por pixel).
- É possível também verificar uma diferença na intensidade dos pixels entre os três exemplos seguintes, onde no primeiro exemplo foi retirado o canal azul, no segundo o canal verde e no terceiro o canal vermelho (intensidade aumenta = claridade aumenta).
- Conforme uma pesquisa online, o canal azul normalmente tem menos influência nestes tipos de imagens médicas, capturando menos detalhes devido ao comprimento de onda mais curto e à maior dispersão no tecido. De seguida vêm o verde e o vermelho, respetivamente, em termos de influência na imagem, o que explica a diferença de intensidade entre eles.



Pasted image 20241116213433.png#center



Pasted image 20241116213457.png#center



Pasted image 20241108180152.png#center

## 2. Construa funções que elimine as componentes R, G ou B de uma imagem RGB. A função deve entrar como parâmetro o canal a extrair. Mostre as imagens resultantes.

Código:

```
1 void exercise2_ws2(int channel) {  
2  
3     // read the image  
4     Mat image_extract = imread("J:/MEEC/PIVC/Projects/Project_PIVC_Worksheets/retina.tif");  
5     // reference image / CV_8UC3 -> 3 channel  
6     Mat newImage(image_extract.rows, image_extract.cols, CV_8UC3, Scalar(0));  
7     // go through all the pixels in the image / loop rows and columns  
8     for (int r = 0; r < image_extract.rows; r++) {  
9         for (int c = 0; c < image_extract.cols; c++) {  
10             Vec3b pixel_value = image_extract.at<Vec3b>(r, c); // extract the values from the picture  
11             pixel_value[channel] = 0; // remove the channel that the user typed // all pixels to 0  
12             newImage.at<Vec3b>(r, c) = pixel_value; // save the other channels in the reference image  
13         }  
14     }  
15     imshow("Image 2 channels", newImage); // show the image  
16     waitKey(0); // key 0 to close  
17     destroyAllWindows(); // destroy the window  
18 }
```

Referências:

### Decomposição da imagem RGB



Pasted image 20241108163051.png > center

Operação	Fórmula	Efeito
<b>Negativo</b>	$C = 255 - C$	Calcula a cor oposta, por exemplo, o preto torna-se branco, o vermelho em ciano, etc.
<b>Escurecer</b>	$C = C - P$	Subtrai por uma constante, para a tornar mais escura.
<b>Clarear</b>	$C = C + P$	Adiciona uma constante, para a tornar mais clara.
<b>Cinzento</b>	$(R + G + B) / 3$	Calcula a média dos três canais (componentes) de cor, de modo a obter uma tonalidade cinzenta.
<b>Remover Canal</b>	$R = 0, G = 0$ e/ou $B = 0$	Colocando um ou mais canais a 0 (zero), remove-se completamente a contribuição dessa componente de cor.
<b>Trocar Canais</b>	$R = G, G = R, \dots$	Trocar valores entre dois canais. Esta operação altera a cor.

Pasted image 20241108174352.png#center

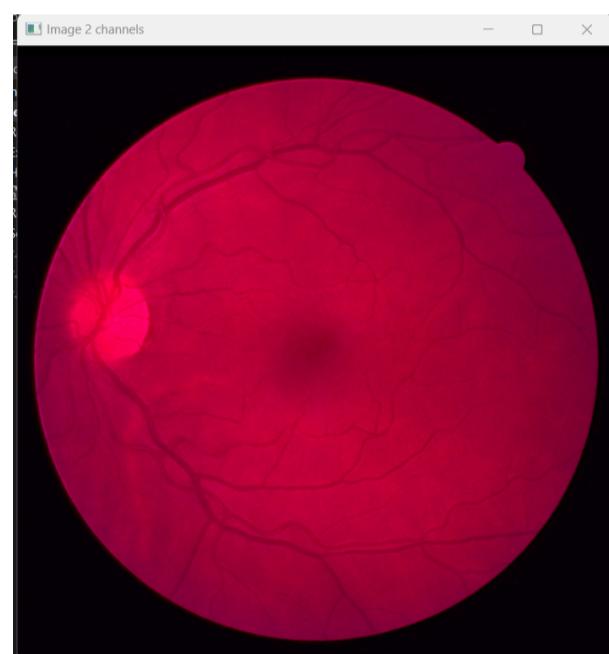
Resultados:

- Extrair um canal é o mesmo que o anular (todos os pixels = 0).
- A função percorre toda a imagem pixel a pixel, e extrai todos os valores das matrizes RGB para uma variável Vec3b.
- Em seguida anula todos os valores que correspondem ao index da matriz escolhida.
- Por último, copia os valores das outras duas matrizes para a nova imagem criada com o tipo CV\_8UC3 (tipo 3 canais por pixel).
- A imagem seguinte é o resultado da remoção do canal azul, que resulta numa imagem com um tom amarelado, da mistura dos canais vermelho e verde.
- O canal azul normalmente tem menos influência nesses tipos de imagens médicas, capturando menos detalhes devido ao comprimento de onda mais curto e à maior dispersão no tecido.



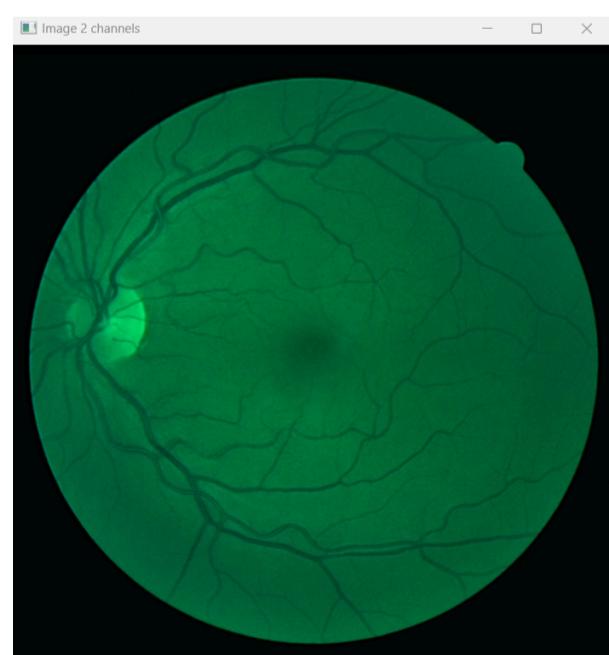
Pasted image 20241108180641.png#center

- A imagem seguinte representa a remoção do canal verde, e demonstra que o canal vermelho tem uma maior intensidade em relação ao azul (visto que o resultado é uma imagem avermelhada).
- O canal vermelho tende a capturar mais detalhes da estrutura geral e do fundo, à medida que a luz vermelha penetra mais profundamente no tecido.



Pasted image 20241108180714.png#center

- A imagem seguinte representa a remoção do canal vermelho.
- Verifica-se que o canal azul tem muito baixa influência (devido ao tom esverdeado do resultado).
- O canal verde fornece maior contraste para os vasos sanguíneos e certas estruturas dentro da retina devido à forma como o olho absorve a luz verde.



Pasted image 20241108180739.png#center

### 3. Construa uma função que converta uma imagem no espaço RGB para uma imagem em tons de cinzento

Código:

```
1 void exercise3_ws2(int userChoice) {  
2  
3     // read the image  
4     Mat image_extract = imread("J:/MEEC/PIVC/Projects/Project_PIVC_Worksheets/retina.tif");  
5     // greyscale reference image / CV_8UC1 - > 1 channel  
6     Mat grayImage(image_extract.rows, image_extract.cols, CV_8UC1, Scalar(0));  
7     cv::String imageLabel; // label to simplify the code  
8  
9     if (userChoice) { //weighted formula  
10        // go through all the pixels in the image / loop rows and columns  
11        for (int r = 0; r < image_extract.rows; r++) {  
12            for (int c = 0; c < image_extract.cols; c++) {  
13                Vec3b pixel_value = image_extract.at<Vec3b>(r, c); // extract the values from the picture  
14                // calculate the intensity using the weighted formula  
15                uchar intensity = (  
16                    (pixel_value[0] * 0.144) +  
17                    (pixel_value[1] * 0.587) +  
18                    (pixel_value[2] * 0.299));  
19                grayImage.at<uchar>(r, c) = intensity; // save the intensity values in the new image  
20            }  
21        }  
22        imageLabel = "Weighted Formula Greyscale Image"; // set the imshow() label to this option  
23    }  
24    else { //normal formula  
25        // go through all the pixels in the image / loop rows and columns  
26        for (int r = 0; r < image_extract.rows; r++) {  
27            for (int c = 0; c < image_extract.cols; c++) {  
28                Vec3b pixel_value = image_extract.at<Vec3b>(r, c); // extract the values from the picture  
29                // calculate the intesity between all channels  
30                uchar intensity = (pixel_value[0] + pixel_value[1] + pixel_value[2]) / 3;  
31                grayImage.at<uchar>(r, c) = intensity; // save the intensity values in the new image  
32            }  
33        }  
34        imageLabel = "Normal Formula Greyscale Image"; // set the imshow() label to this option  
35    }  
36    imshow(imageLabel, grayImage); // show the image  
37    waitKey(0); // key 0 to close  
38    destroyAllWindows(); // destroy the window  
39}
```

Referências:

Operação	Fórmula	Efeito
<b>Negativo</b>	$C = 255 - C$	Calcula a cor oposta, por exemplo, o preto torna-se branco, o vermelho em ciano, etc.
<b>Escurecer</b>	$C = C - P$	Subtrai por uma constante, para a tornar mais escura.
<b>Clarear</b>	$C = C + P$	Adiciona uma constante, para a tornar mais clara.
<b>Cinzento</b>	$(R + G + B) / 3$	Calcula a média dos três canais (componentes) de cor, de modo a obter uma tonalidade cinzenta.
<b>Remover Canal</b>	$R = 0, G = 0$ e/ou $B = 0$	Colocando um ou mais canais a 0 (zero), remove-se completamente a contribuição dessa componente de cor.
<b>Trocar Canais</b>	$R = G, G = R, \dots$	Trocar valores entre dois canais. Esta operação altera a cor.

Pasted image 20241108174352.png#center

$$\text{Intensidade} = R * 0.299 + G * 0.587 + B * 0.144$$

Pasted image 20241108173002.png > center

### Resultados:

- A imagem é percorrida pixel a pixel e os valores da intensidade para as 3 matrizes são extraídos da imagem original para um vetor Vec3b.
- Conforme a escolha do utilizador, será efetuado o calculo com a formula da média aritmética, ou com a fórmula da média pesada.
- No caso da média aritmética, calculou-se o valor da nova intensidade que corresponde a uma média dos valores dos pixels dos 3 canais (conforme a tabela nas referências).
- A seguinte imagem corresponde ao output do cálculo da média aritmética.



Pasted image 20241108181625.png#center

- No caso da média pesada, recorreu-se a uma fórmula que corresponde ao peso das cores na percepção do olho humano (conforme a imagem nas referências).
- A seguinte imagem corresponde ao output do cálculo da média pesada.



Pasted image 20241108181716.png#center

- Por fim, os valores da intensidade são "transferidos" para a nova imagem greyscale com apenas 1 canal (tipo CV\_8UC1).
- Em comparação aos dois outputs, não são detectáveis quaisquer diferenças consideráveis, no entanto, com a média pesada, a imagem parece ligeiramente mais clara na zona do nervo óptico.

## Section 2 - Histogramas e operações ponto-a-ponto

4. Construa uma função que calcule o negativo de uma imagem. Faça para uma imagem RGB e para uma imagem em grayscale.

- Use a imagem "breast.png";
- Use a imagem "retina.tif";

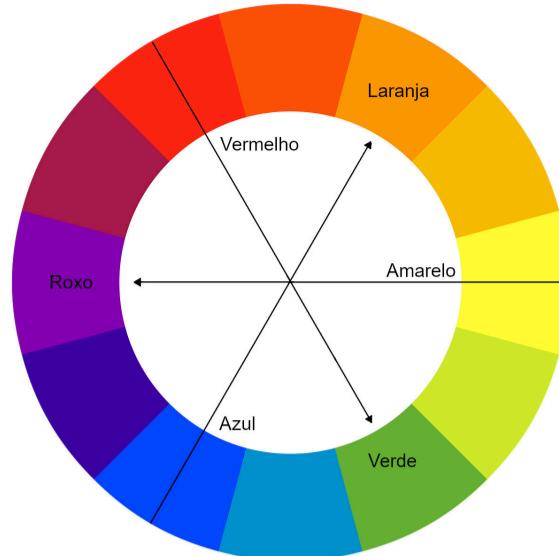
Código:

```
1 void exercise4_ws2(int userChoice) {  
2  
3     if (userChoice) { // greyscale option  
4         // read the breast image as greyscale  
5         Mat image = imread("J:/MEEC/PIVC/Projects/Project_PIVC_Worksheets/breast_image.png",  
6         IMREAD_GRAYSCALE);  
7         // greyscale reference image / CV_8UC1 -> Channel  
8         Mat newImage(image.rows, image.cols, CV_8UC1, Scalar(0));  
9  
10        //since all pixels and channels are modified using the same expression, we can avoid the use of 2  
11        loops  
12        for (int i = 0; i < image.rows * image.cols * image.channels(); i++) {  
13            newImage.data[i] = 255 - image.data[i]; // invert the values  
14        }  
15        imshow("Image Negative Greyscale", newImage); // show the image  
16    }  
17    else { // rgb option  
18        // read the breast image as rgb  
19        Mat image = imread("J:/MEEC/PIVC/Projects/Project_PIVC_Worksheets/retina.tif");  
20        // reference image / CV_8UC3 -> 3 channels  
21        Mat newImage(image.rows, image.cols, CV_8UC3, Scalar(0));  
22  
23        //since all pixels and channels are modified using the same expression, we can avoid the use of 2  
24        loops  
25        for (int i = 0; i < image.rows * image.cols * image.channels(); i++) {  
26            newImage.data[i] = 255 - image.data[i]; // invert the values  
27        }  
28        imshow("Image Negative RGB", newImage); // show the image  
29    }  
30    waitKey(0); // key 0 to close  
31    destroyAllWindows(); // destroy the window  
32}
```

Referências:

Operação	Fórmula	Efeito
<b>Negativo</b>	$C = 255 - C$	Calcula a cor oposta, por exemplo, o preto torna-se branco, o vermelho em ciano, etc.
<b>Escurecer</b>	$C = C - P$	Subtrai por uma constante, para a tornar mais escura.
<b>Clarear</b>	$C = C + P$	Adiciona uma constante, para a tornar mais clara.
<b>Cinzento</b>	$(R + G + B) / 3$	Calcula a média dos três canais (componentes) de cor, de modo a obter uma tonalidade cinzenta.
<b>Remover Canal</b>	$R = 0, G = 0$ e/ou $B = 0$	Colocando um ou mais canais a 0 (zero), remove-se completamente a contribuição dessa componente de cor.
<b>Trocar Canais</b>	$R = G, G = R, \dots$	Trocar valores entre dois canais. Esta operação altera a cor.

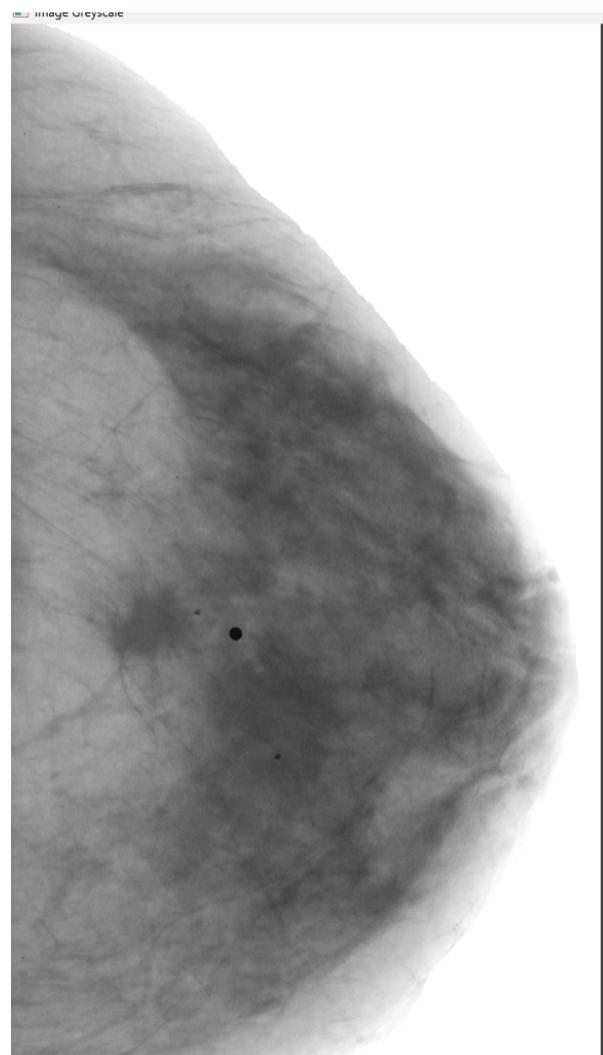
Pasted image 20241108174352.png#center



Pasted image 20241112154044.png#center

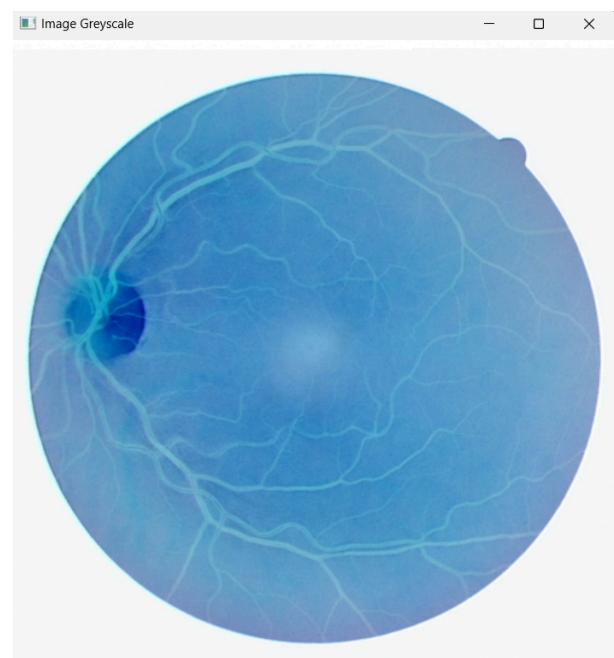
### Resultados:

- O código funciona da mesma maneira em termos de inversão de cor, as diferenças estão na quantidade de canais, que para RGB são 3 e greyscale é 1. Logo, a imagem de referência criada terá os tipos CV\_8UC1 para 1 canal e CV\_8UC3 para 3 canais.
- No caso greyscale, a chamada da função `imread()`, tem o parâmetro adicional `IMREAD_GRAYSCALE` para realizar a leitura direta, sem necessidade de converter.
- Como todos os pixels sofrem alteração, não existe necessidade dos dois loops para fazer as alterações ponto a ponto.
- Para calcular o negativo recorreu-se à tabela nos slides, respeitando a fórmula de  $C = 255 - C$  (como pode ser verificado através da tabela na aba referências).
- Na imagem Breast verificou-se que a inversão de cor vai colocar o background da imagem a branco, e as zonas dentro do RaioX que anteriormente eram mais claras, são agora as mais escuras e vice-versa (por exemplo, os pontos/nódulos são agora de cor preta).



Pasted image 20241108175223.png#center

- Conforme verificado nas referências o negativo de uma cor alaranjada é uma cor azulada, e na conversão realizada na imagem Retina, o output corresponde ao inverso de uma cor preta e laranja (background e foreground), que é branco e azul.
- Tal como na imagem Breast, as zonas que na imagem original da Retina eram mais escuras, são agora as mais claras e vice-versa (por exemplo, a zona do nervo óptico que era mais clara na imagem original).



Pasted image 20241109154427.png#center

**5. Repita o exercício anterior mas evitando a alteração onde não é pretendido (background da imagem). Nota: para a imagem "breast.png", utilize a mascara fornecida. Para a imagem "retina.tif" deverá utilizar uma condição para verificar os valores que não devem ser alterados.**

Código:

```

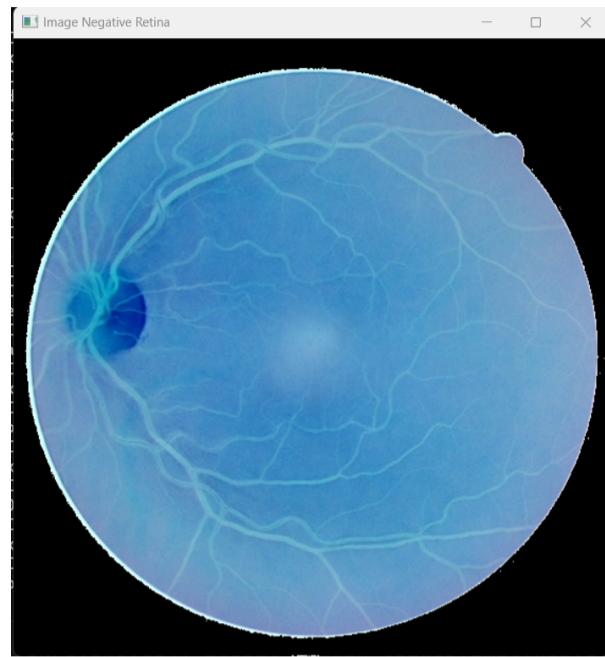
1 void exercise5_ws2(int userChoice) {
2
3     if (userChoice) { //breast option
4         // read the breast image as greyscale and the mask too
5         Mat image = imread("J:/MEEC/PIVC/Projects/Project_PIVC_Worksheets/breast_image.png",
6         IMREAD_GRAYSCALE);
7         Mat mask = imread("J:/MEEC/PIVC/Projects/Project_PIVC_Worksheets/breast_mask.png",
8         IMREAD_GRAYSCALE);
9         // greyscale reference image / CV_8UC1 -> Channel
10        Mat invertedImage(image.rows, image.cols, CV_8UC1, Scalar(0));
11        //since all pixels and channels are modified using the same expression, we can avoid the use of 2
12        loops
13        for (int i = 0; i < image.rows * image.cols * image.channels(); i++) {
14            if (mask.data[i] != 0) { // apply the negative only on the xray (foreground)
15                invertedImage.data[i] = 255 - image.data[i]; // invert the pixels
16            }
17        }
18        imshow("Image Negative Breast Mask", invertedImage); // show the image
19    }
20    else { // retina option
21        Mat image = imread("J:/MEEC/PIVC/Projects/Project_PIVC_Worksheets/retina.tif");
22        // rgb reference image / CV_8UC3 -> Channel
23        Mat inverted_img(image.rows, image.cols, CV_8UC3, Scalar(0));
24
25        //since all pixels and channels are modified using the same expression, we can avoid the use of 2
26        loops
27        for (int i = 0; i < image.rows; i++) {
28            for (int j = 0; j < image.cols; j++) {
29                Vec3b vec_value = image.at<Vec3b>(i, j);
30                // inversion of all channels in a threshold (intensity value 15)
31                if ((vec_value[0] > 15) || (vec_value[1] > 15) || (vec_value[2] > 15)) {
32                    vec_value[0] = 255 - vec_value[0]; // invert the pixels on channel B
33                    vec_value[1] = 255 - vec_value[1]; // invert the pixels on channel G
34                    vec_value[2] = 255 - vec_value[2]; // invert the pixels on channel R
35                    inverted_img.at<Vec3b>(i, j) = vec_value; // save on the reference image
36                }
37            }
38        }
39        imshow("Image Negative Retina", inverted_img); // show the image
40    }
41    waitKey(0); // key 0 to close
42    destroyAllWindows(); // destroy the window
43 }
```

Referências:

Operação	Fórmula	Efeito
<b>Negativo</b>	$C = 255 - C$	Calcula a cor oposta, por exemplo, o preto torna-se branco, o vermelho em ciano, etc.
<b>Escurecer</b>	$C = C - P$	Subtrai por uma constante, para a tornar mais escura.
<b>Clarear</b>	$C = C + P$	Adiciona uma constante, para a tornar mais clara.
<b>Cinzento</b>	$(R + G + B) / 3$	Calcula a média dos três canais (componentes) de cor, de modo a obter uma tonalidade cinzenta.
<b>Remover Canal</b>	$R = 0, G = 0$ e/ou $B = 0$	Colocando um ou mais canais a 0 (zero), remove-se completamente a contribuição dessa componente de cor.
<b>Trocar Canais</b>	$R = G, G = R, \dots$	Trocar valores entre dois canais. Esta operação altera a cor.

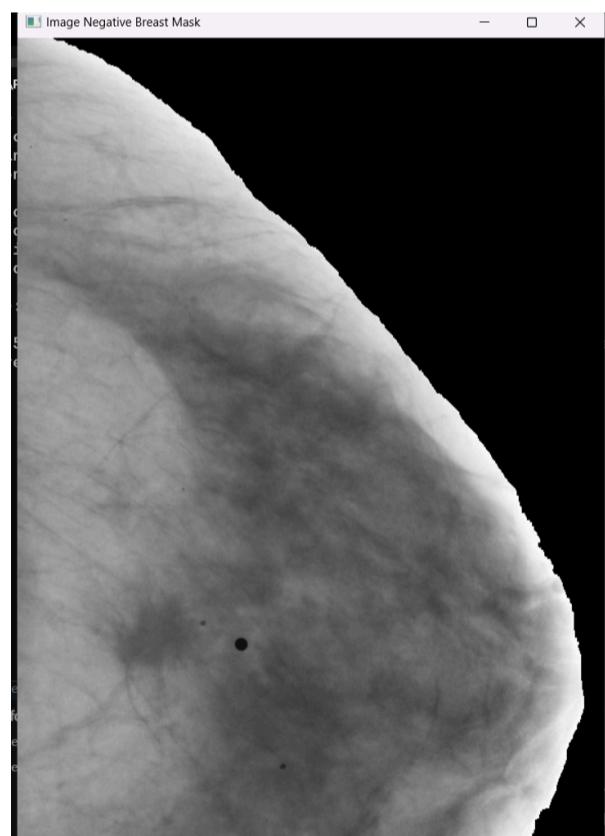
### Resultados:

- A abordagem para realizar o exercício da imagem Retina passou-se pela realização da inversão de cada canal conforme um threshold de cor. Esse threshold foi definido para isolar a cor branca do background, e eliminar essa zona.
- Como não será aplicado a toda a imagem o negativo, voltou-se a utilizar os dois for loops para percorrer a imagem pixel a pixel, e para todas as intensidades acima do valor 15, aplica-se a fórmula a cada canal individualmente.



Pasted image 2024111151303.png#center

- No caso da imagem Breast, utilizou-se uma máscara binária para isolar o background da imagem.
- Neste caso a imagem é percorrida pixel a pixel através de apenas 1 for loop.
- Como condição, toda a zona que estiver a preto na máscara é correspondente ao background e o valor das intensidades dos pixels nessa zona, é ignorado.
- Apenas a zona de interesse do RaioX é transferida e calculado o negativo para a imagem de referência final.



Pasted image 2024111151338.png#center

- Os resultados finais das imagens coincidem com o exercício anterior, com apenas a diferença da extração da zona de interesse.

**6. Faça uma função para alterar o brilho de uma imagem. Aplique à imagem “breast.png”. Mostre o resultado de duas operações (diminuir e aumentar o brilho).**

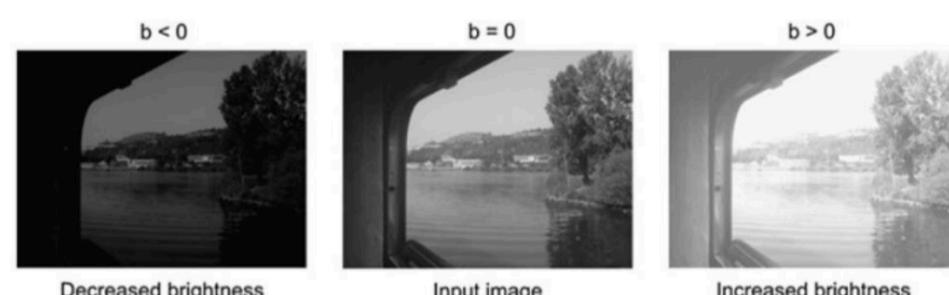
Código:

```
1 void exercise6_ws2(int userChoice, int bvalue) {  
2  
3     // read the image  
4     Mat image = imread("J:/MEEC/PIVC/Projects/Project_PIVC_Worksheets/breast_image.png");  
5     //reference image / CV_8UC3 -> 3 channels  
6     Mat outputImage(image.rows, image.cols, CV_8UC3, Scalar(0));  
7     cv::String imageLabel; // label to simplify the code  
8  
9     if (userChoice) { // brigther  
10        //since all pixels and channels are modified using the same expression, we can avoid the use of 2  
11        loops  
12        for (int i = 0; i < image.rows * image.cols * image.channels(); i++) {  
13            outputImage.data[i] = image.data[i] + bvalue; // addition of a constant to increase the  
14            brightness  
15        }  
16        imageLabel = "Increased Brightness"; // set the imshow() label to this  
17    }  
18    else {  
19        //since all pixels and channels are modified using the same expression, we can avoid the use of 2  
20        loops  
21        for (int i = 0; i < image.rows * image.cols * image.channels(); i++) {  
22            outputImage.data[i] = image.data[i] - bvalue; // addition of a constant to decrease the  
23            brightness  
24        }  
25        imageLabel = "Decreased Brightness"; // set the imshow() label to this  
26    }  
27    imshow(imageLabel, outputImage); // show the image  
28    waitKey(0); // key 0 to close  
29    destroyAllWindows(); // destroy the window  
30 }
```

Referências:

Operação	Fórmula	Efeito
<b>Negativo</b>	$C = 255 - C$	Calcula a cor oposta, por exemplo, o preto torna-se branco, o vermelho em ciano, etc.
<b>Escurecer</b>	$C = C - P$	Subtrai por uma constante, para a tornar mais escura.
<b>Clarear</b>	$C = C + P$	Adiciona uma constante, para a tornar mais clara.
<b>Cinzento</b>	$(R + G + B) / 3$	Calcula a média dos três canais (componentes) de cor, de modo a obter uma tonalidade cinzenta.
<b>Remover Canal</b>	$R = 0, G = 0$ e/ou $B = 0$	Colocando um ou mais canais a 0 (zero), remove-se completamente a contribuição dessa componente de cor.
<b>Trocar Canais</b>	$R = G, G = R, \dots$	Trocar valores entre dois canais. Esta operação altera a cor.

Pasted image 20241108174352.png#center



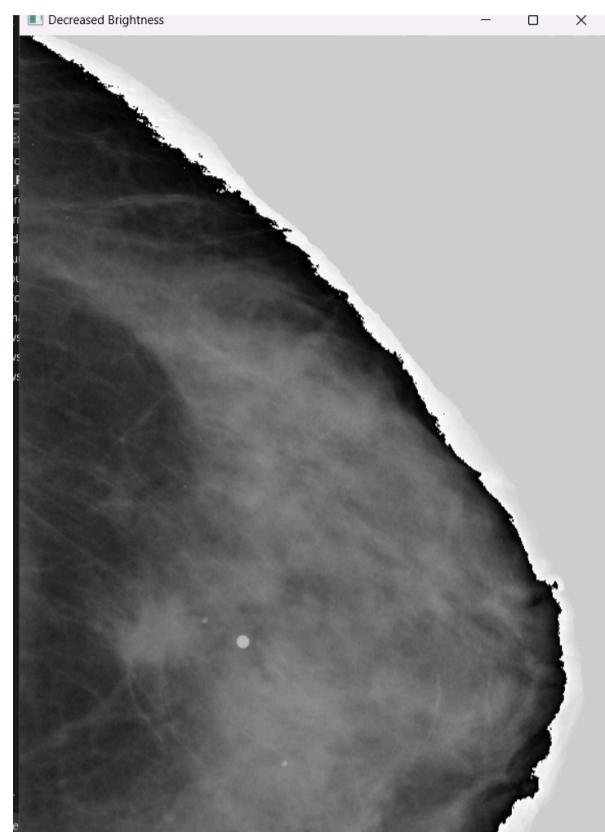
Pasted image 20241109225514.png#center

### *Resultados:*

- Foi adicionada uma constante b, para manipular os valores da intensidade dos pixels.
- O código percorre ponto a ponto por todos os pixels de todos os canais, para poder adicionar ou subtrair o valor de b.
- A abordagem para o aumento ou decréscimo da luminosidade foi baseada nos exemplos dos slides, onde é introduzida uma constante na fórmula de "transferência" entre a imagem original e a imagem de referência final (tabela nas referências).

### *Diminuição da luminosidade:*

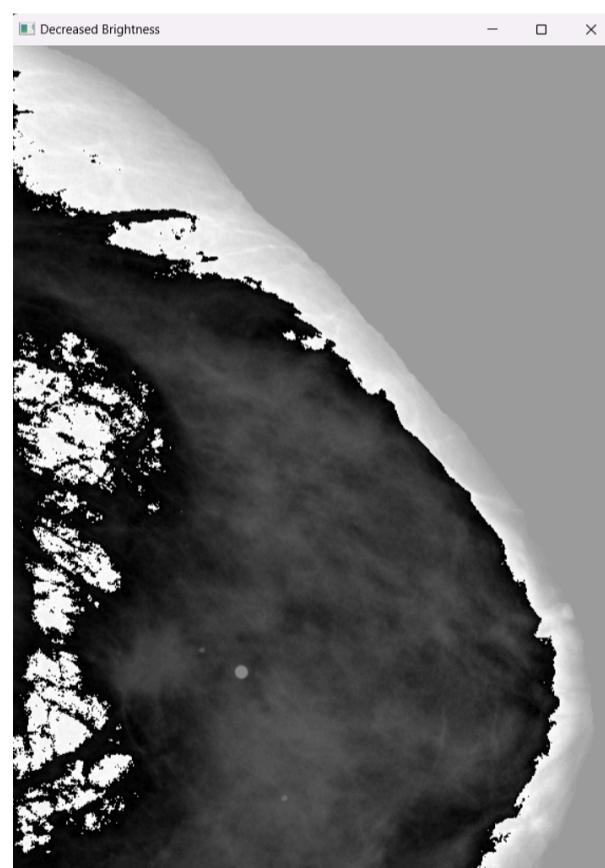
b=50



Pasted image 20241109224906.png#center

### *Diminuição brusca de luminosidade:*

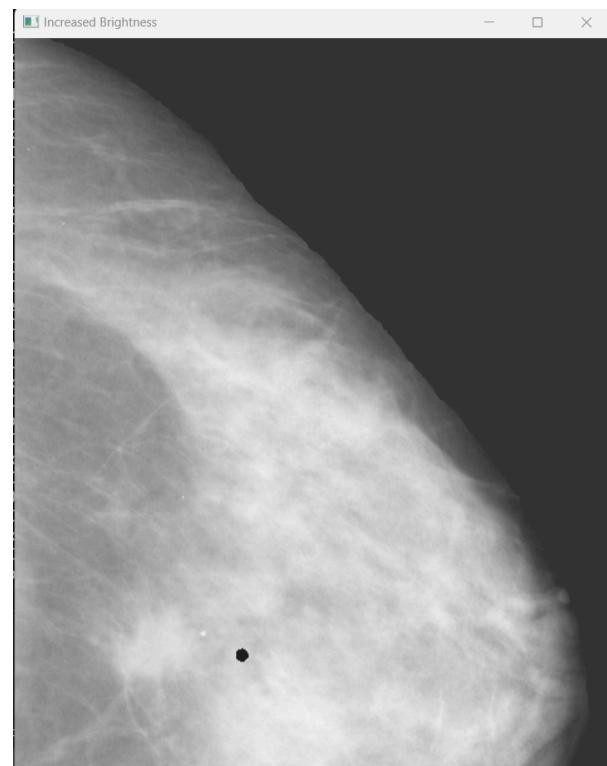
b=100



Pasted image 20241112181832.png#center

*Aumento de luminosidade:*

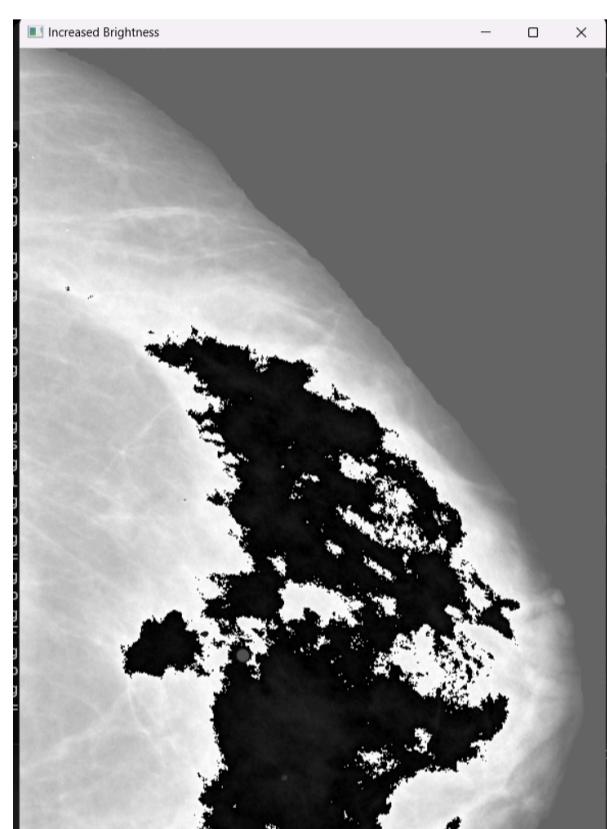
b=50



Pasted image 20241109225014.png#center

*Aumento brusco de luminosidade:*

b=100



Pasted image 20241112181753.png#center

- Se o valor for muito alto, vai danificar a imagem com o aparecimento de artefatos, tanto na diminuição como no aumento.
- Em comparação entre as duas operações, é possível verificar que ao diminuir 50 unidades na intensidade dos pixels, já é possível ver a perda de informação (zonas brancas) na periferia do raioX, com efeito maior no exemplo 100 unidades.
- No caso do aumento de luminosidade com o aumento de 50 unidades, a imagem fica conforme o que é esperado, sem perder informação (em contrapartida à diminuição). No entanto, nas 100 unidades já é possível verificar perdas (efeito de clipping).

**7. Faça uma função para alterar o contraste de uma imagem. Aplique à imagem “breast.png”. Mostre o resultado de duas operações (diminuir e aumentar o contraste).**

Código:

```
1 void exercise7_ws2(float alphavalue) {  
2     // read the breast image  
3     Mat image = imread("J:/MEEC/PIVC/Projects/Project_PIVC_Worksheets/breast_image.png");  
4     // reference image / CV_8UC3 -> 3 channels  
5     Mat outputImage(image.rows, image.cols, CV_8UC3, Scalar(0));  
6     cv::String imageLabel; // label to simplify the code  
7  
8     //since all pixels and channels are modified using the same expression, we can avoid the use of 2  
loops  
9     for (int i = 0; i < image.rows * image.cols * image.channels(); i++) {  
10         // addition of a constant increase the contrast  
11         // alphavalue > 1 = more contrast / alphavalue < 1 = less contrast  
12         outputImage.data[i] = image.data[i] * alphavalue;  
13     }  
14     if (alphavalue > 1) { // more contrast  
15         imageLabel = "Increased Contrast"; // set the imshow() label to this example  
16     }  
17     else{ //less contrast  
18         imageLabel = "Decreased Contrast"; // set the imshow() label to this example  
19     }  
20     imshow(imageLabel, outputImage);  
21     waitKey(0); // key 0 to close  
22     destroyAllWindows(); // destroy the window  
23 }
```

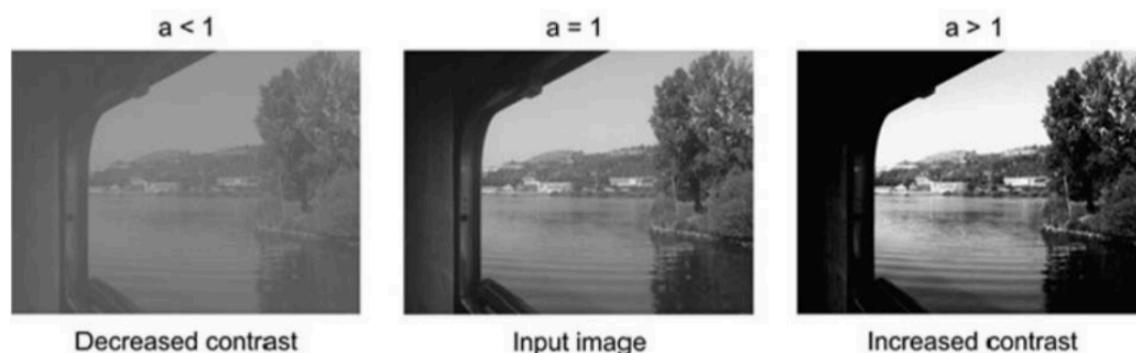
Referências:

#### MANIPULAÇÃO DO CONTRASTE

$$g(i,j) = f(i,j) \times \alpha + b;$$

- se  $\alpha < 1$ , há uma atenuação do contraste;
- se  $\alpha > 1$ , há um aumento do contraste;
- Devemos fazer o clipping dos valores que excedam a gama dinâmica permitida.

Pasted image 20241109225405.png#center



Pasted image 20241109225635.png#center

### Resultados:

- Foi adicionada uma constante alpha, para manipular os valores da intensidade dos pixels.
- O código percorre ponto a ponto por todos os pixels de todos os canais, para poder manipular a intensidade conforme valor de alpha.
- A abordagem para o aumento ou decréscimo da contraste foi baseada nos exemplos dos slides, onde é introduzida uma constante na fórmula de "transferência" entre a imagem original e a imagem de referência final (fórmula nas referências).
- Neste caso o valor da constante é multiplicado pelo valor da intensidade do pixel.

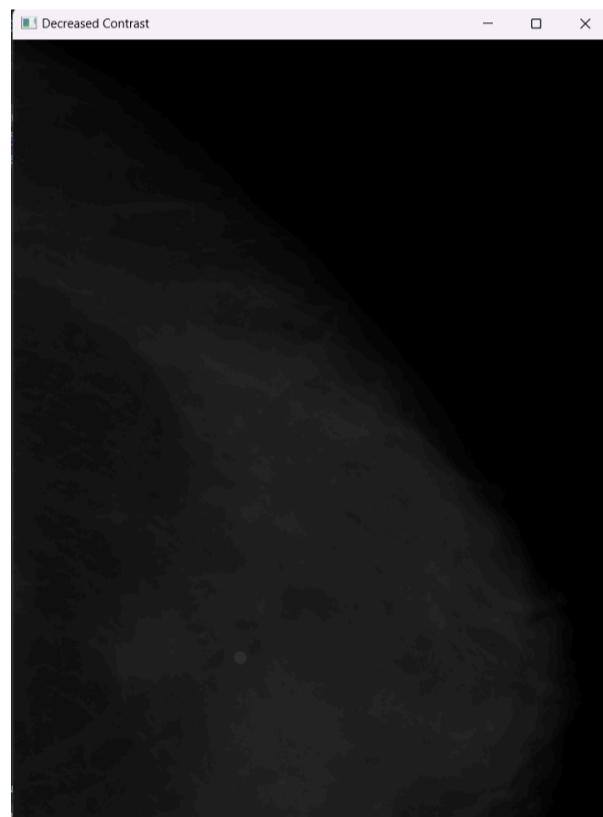
### Menor contraste:

- $a = 0.5$



Pasted image 20241109225942.png#center

- $a = 0.2$

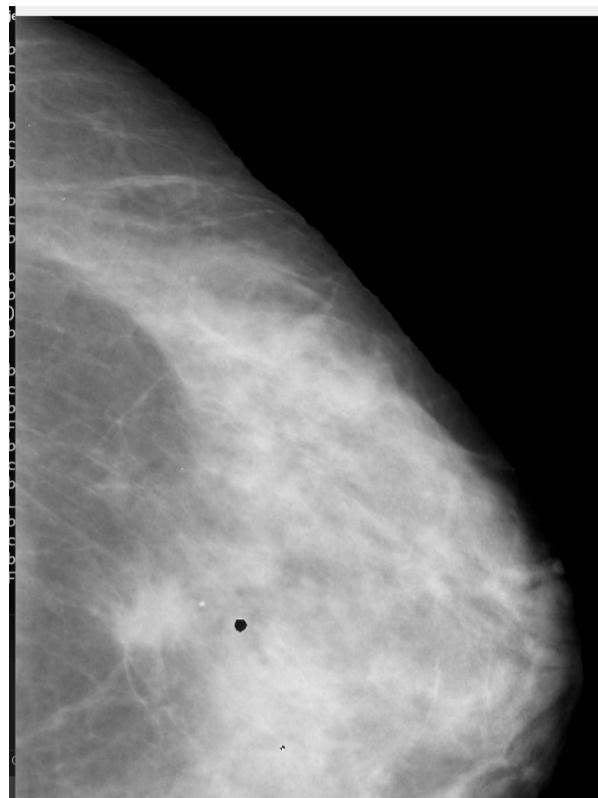


Pasted image 20241116223711.png#center

- Quanto menor o valor, mais "escurecida" a imagem fica.
- O intervalo entre as áreas claras e escuras diminui, fazendo com que a imagem pareça opaca.

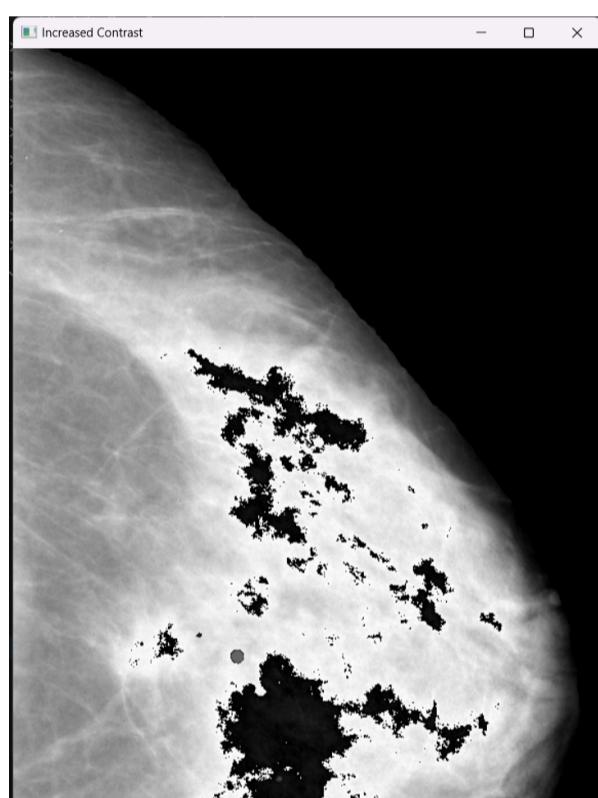
### Maior contraste:

- $a = 1.2$



Pasted image 20241116224114.png#center

- $a = 1.5$



Pasted image 20241109230026.png#center

- Aumentando o contraste, é possível verificar que as áreas claras ficam mais claras e as áreas escuras ficam mais escuras (tudo mais definido).
- Se o valor for muito alto, vai danificar a imagem com o aparecimento de artefatos (efeito de clipping).