



Relatório

TPC 3

Unidade Curricular:
Análise e Transformação de Dados (ATD)

Licenciatura em Engenharia e Ciência de Dados

Realizado por:

Diogo Beltran Dória, 2020246139
Mariana Lopes Paulino, 2020190448

Ano Letivo 2021/2022

A partir da imagem dada 'pneumonia.jpg' este trabalho constrói-se à sua volta. Primeiramente, temos de carregar a imagem para o MATLAB e recorrendo ao *imshow* abrimo-la para que possamos verificar se a imagem é a que pretendíamos utilizar.

```
%% 1. Carregamento da Imagem  
I = imread('pneumonia.jpg');  
imshow(I)
```

Figura 1- Código para carregamento da imagem para o workspace do MATLAB

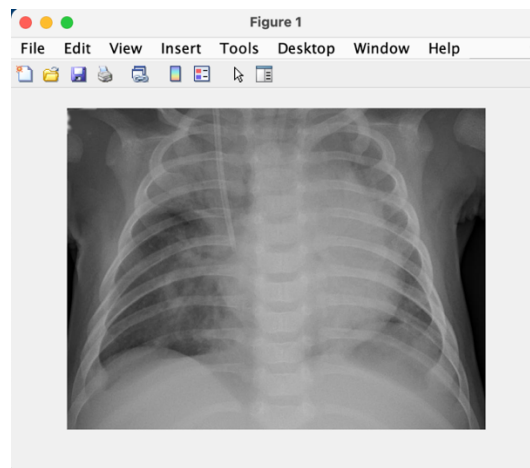


Figura 2- Imagem obtida aquando do carregamento inicial

No segundo ponto do trabalho é nos pedido que utilizando a função *rgb2gray* transformemos a imagem original dada para a escala de cinzas de modo a trabalhar com um único plano. Tal é possível a partir do código representado na figura 3, obtendo o resultado representado na figura 4.

```
%% 2. Conversão da Imagem para Escala de Cinzas  
I2 = rgb2gray(I);  
figure  
imshow(I2)
```

Figura 3- Conversão da Imagem Inicial para Escala de Cinzas

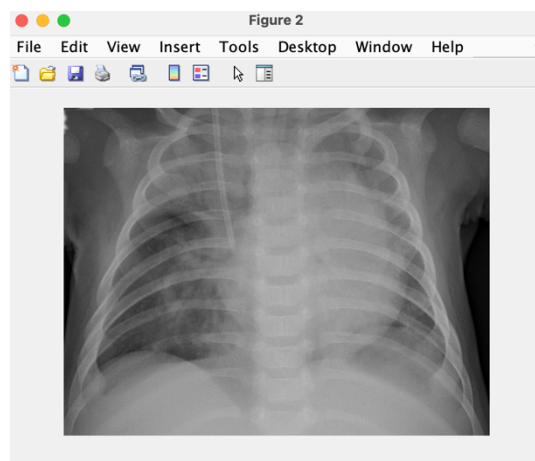


Figura 4- Resultado Obtido após a conversão

Após a conversão da imagem para uma escala de cinzas o exercício seguinte pede para definirmos as matrizes $h1$, $h2$, $h3$, $h4$ que representam a resposta a um impulso de quatro sistemas LTI (*Linearmente Invariantes no Tempo*) sendo as matrizes:

$$h1 = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad h2 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$h3 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \quad h4 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}$$

Estas matrizes representam operadores de Prewitt que permitem obter o gradiente das imagens utilizando a derivada parcial pixel a pixel.

Para as definir utilizamos a designação $h1$, $h2$, $h3$ e $h4$ como é demonstrado na figura seguinte.

%% 3. Definir as Matrizes

```
h1 = [1 0 -1;1 0 -1;1 0 -1];
h2 = [1 1 1;0 0 0;-1 -1 -1];
h3 = [0 1 1;-1 0 1;-1 -1 0];
h4 = [1 1 0;1 0 -1;0 -1 -1];
```

Figura 5- Código utilizado na definição das matrizes

Após a definição das matrizes é pedido que utilizemos a função de convolução 2D à imagem obtida no exercício 2 utilizando cada um dos kernels (h_i) a fim de obtermos quatro imagens de saída (y_i).

%% 4. Aplicar convolução 2D

```
y1 = conv2(I2,h1,'same');
figure, imshow(y1)

y2 = conv2(I2,h2,'same');
figure, imshow(y2)

y3 = conv2(I2,h3,'same');
figure, imshow(y3)

y4 = conv2(I2,h4,'same');
figure, imshow(y4)
```

Figura 6- Código utilizado na Convolução 2D

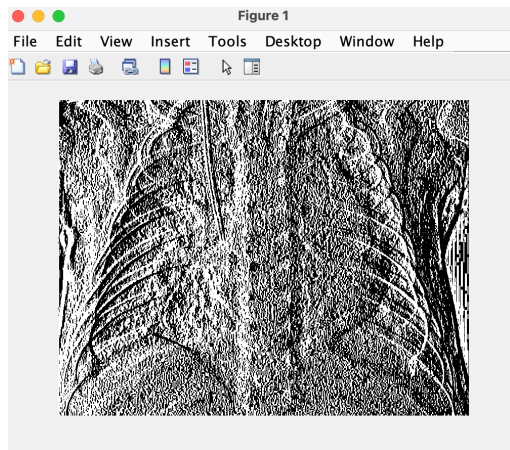


Figura 7- y1 resultante da convolução 2D com a matriz h1

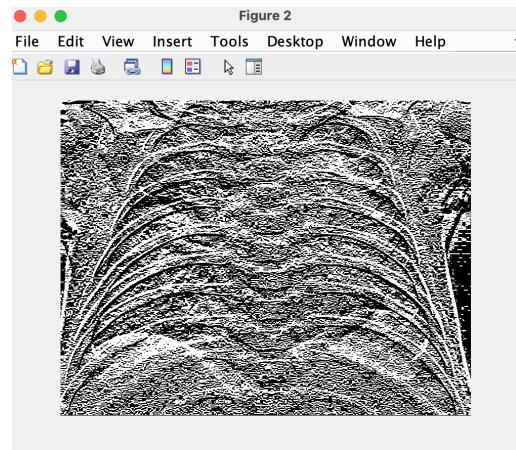


Figura 8- y2 resultante da convolução 2D com a matriz h2

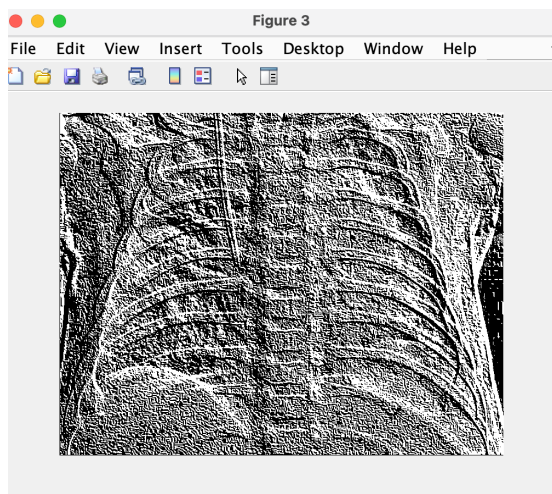


Figura 9- y3 resultante da convolução 2D com a matriz h3

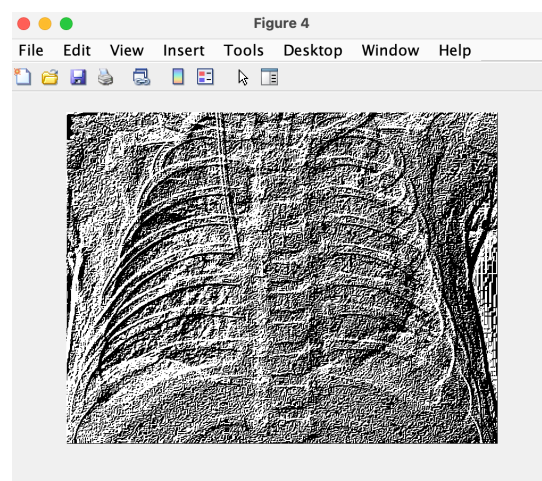


Figura 10- y4 resultante da convolução 2D com a matriz h4

Aplicadas os operadores com orientações em diferentes direções sobre a imagem, obtivemos os anteriores resultados, que correspondem a figura 7,8,9 e 10, nas quais começamos a obter uma imagem mais nítida que a original, isto é, cada operador tem o seu diferente efeito sobre a imagem. Resultando em algumas imagens mais nítidas que outras.

No último exercício deste trabalho é nos pedido para realizar a soma de todas as imagens resultantes da convolução com a inicial. As imagens resultantes da convolução devem ser convertidas primeiro em uint8 através do seguinte código cumprimos todos os objetivos do exercício levando ao resultado representado na figura 12.

%% 6. Soma das Imagens

```
sumall = I + uint8(y1) + uint8(y2) + uint8(y3) + uint8(y4);
figure
imshow(sumall);
```

Figura 11- Código utilizado na soma das imagens transformando as imagens resultantes da convolução em uint8

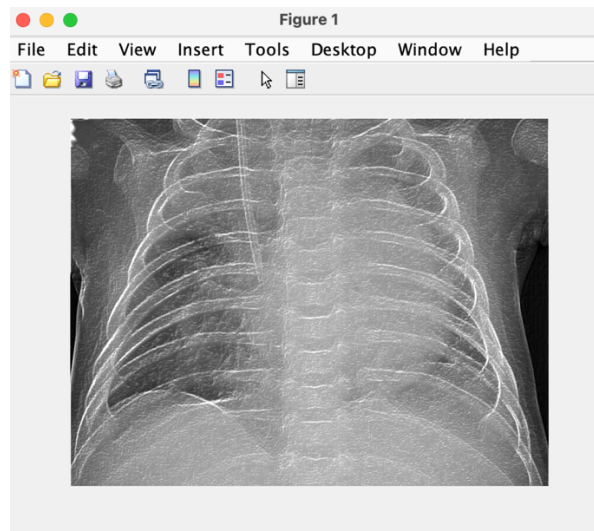


Figura 12- Imagem resultante da soma de todas as imagens resultantes da convolução2D com a imagem original

Nesta figura 12, uma vez somadas os resultados dos operadores com a imagem rgb convertidos a uint8, com a nossa imagem original. O resultado permite uma melhor visualização da amostra.

Todo o código utilizado na realização deste trabalho encontra-se abaixo:

```
%% TPC3_ATD_DiogoDoria_MarianaPaulino

%% 1. Carregamento da Imagem
I = imread('pneumonia.jpg');
imshow(I)

%% 2. Conversão da Imagem para Escala de Cinzas
I2 = rgb2gray(I);
figure
imshow(I2)

%% 3. Definir as Matrizes
h1 = [1 0 -1;1 0 -1;1 0 -1];
h2 = [1 1 1;0 0 0;-1 -1 -1];
h3 = [0 1 1;-1 0 1;-1 -1 0];
h4 = [1 1 0;1 0 -1;0 -1 -1];

%% 4. Aplicar convolução 2D
y1 = conv2(I2,h1,'same');
figure, imshow(y1)
y2 = conv2(I2,h2,'same');
figure, imshow(y2)
y3 = conv2(I2,h3,'same');
figure, imshow(y3)
y4 = conv2(I2,h4,'same');
figure, imshow(y4)

%% 6. Soma das Imagens
sumall = I + uint8(y1) + uint8(y2) + uint8(y3) + uint8(y4);
figure
imshow(sumall);
```