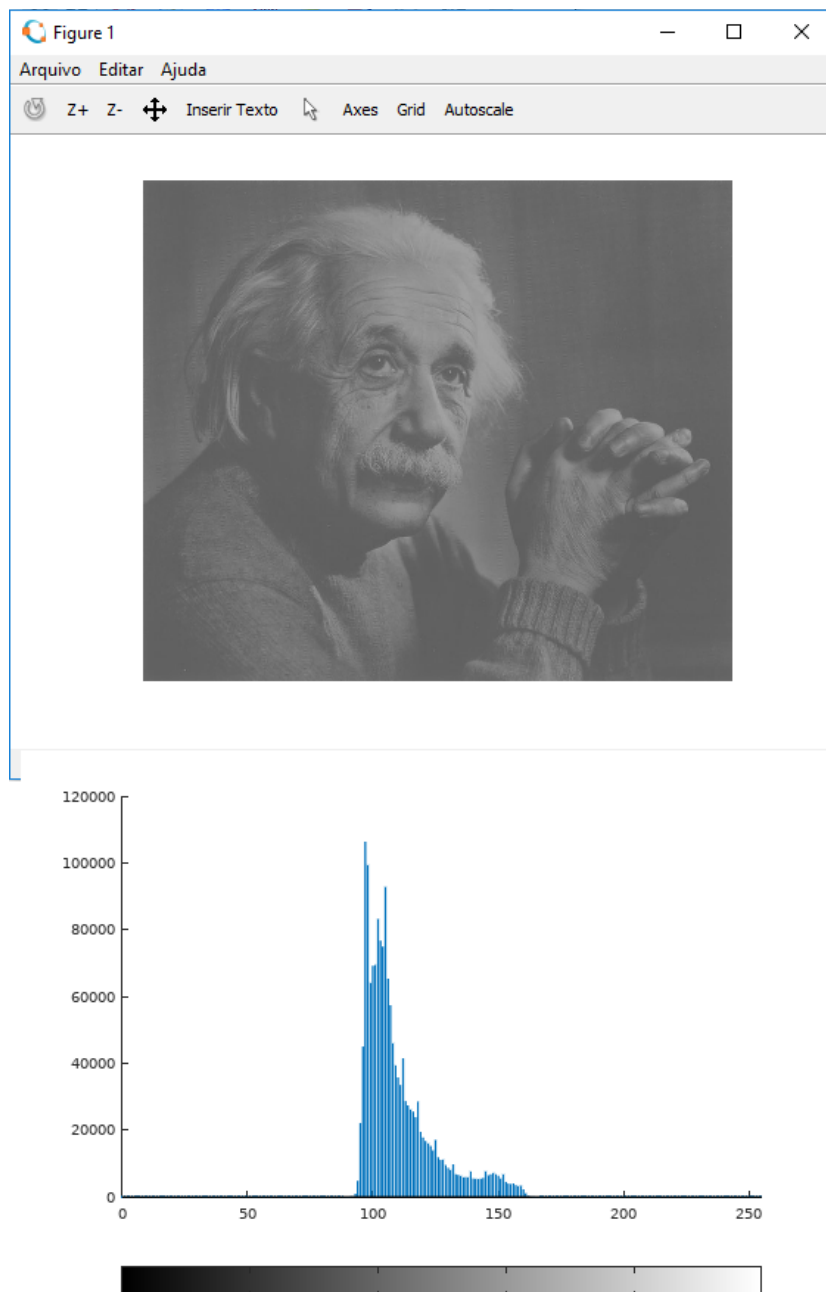


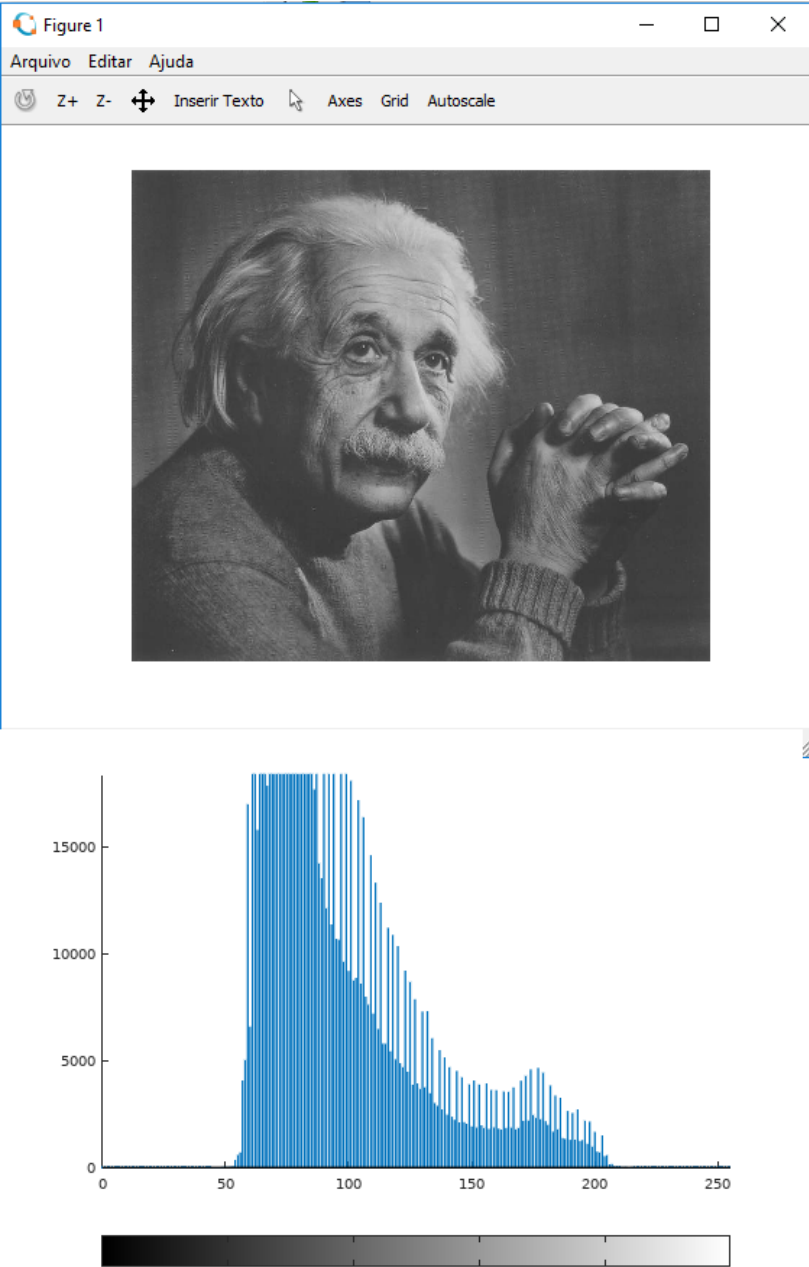
Exercício 1:

- Leia a ajuda para as funções `imadjust` e `imhist`.
`imadjust` ajusta a intensidade da imagem ou do mapa de cores (valores);
`imhist` produz um histograma da imagem.
- Leia as imagens `Einstein_low_contrast.png`, `Einstein_med_contrast.png` e `Einstein_high_contrast.png`.
`low = imread("Einstein_low_contrast.png");`
`med = imread("Einstein_med_contrast.png");`
`high = imread("Einstein_high_contrast.png");`
- Apresente cada imagem e seu respectivo histograma em uma figura diferente. Compare os histogramas.

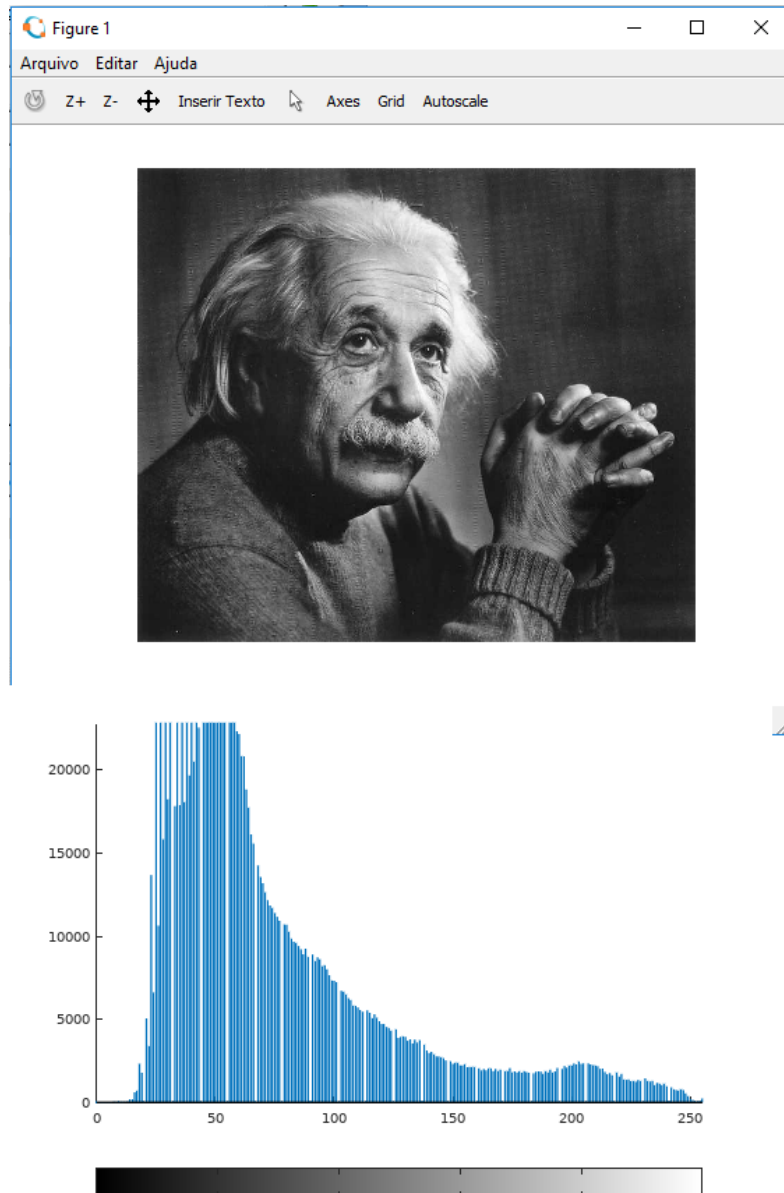
Low



Med



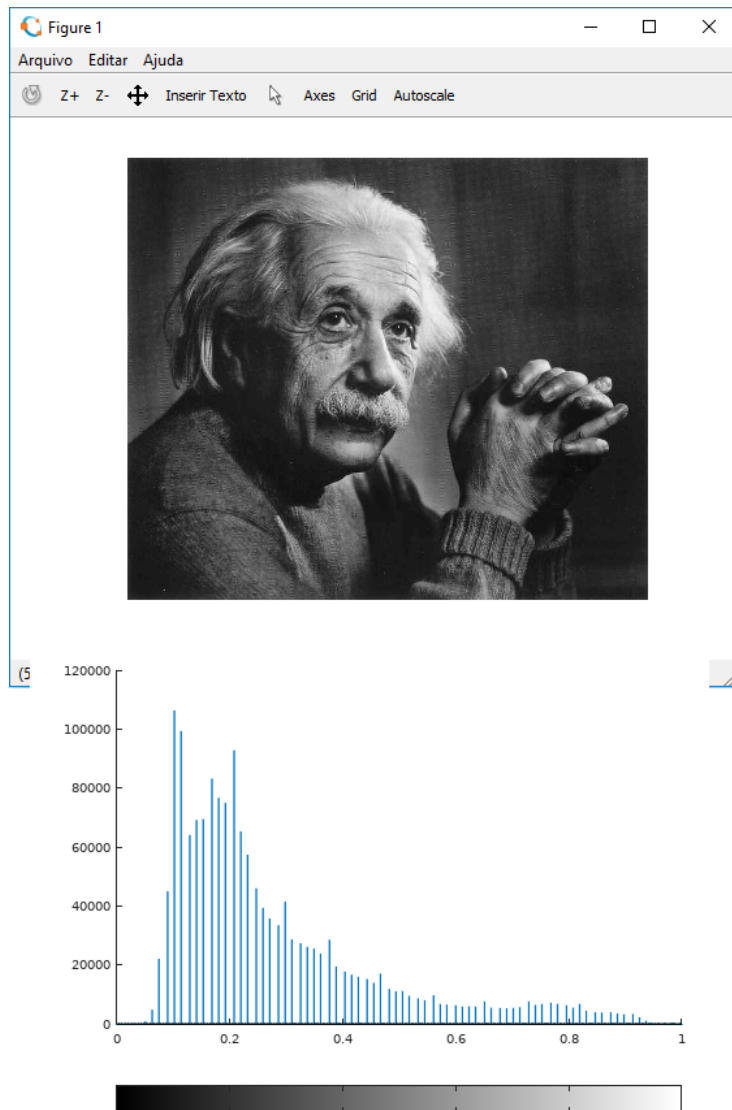
High



- Utilize a função `imadjust` para melhorar o contraste da imagem `Einstein_low_contrast.png`, de forma a que o histograma da imagem resultante seja parecido com o da imagem `Einstein_high_contrast.png`.

```
new_low = imadjust(double(low)/255,[0.35 0.65],[]);
```

- Apresente a nova imagem e seu histograma em uma única figura.



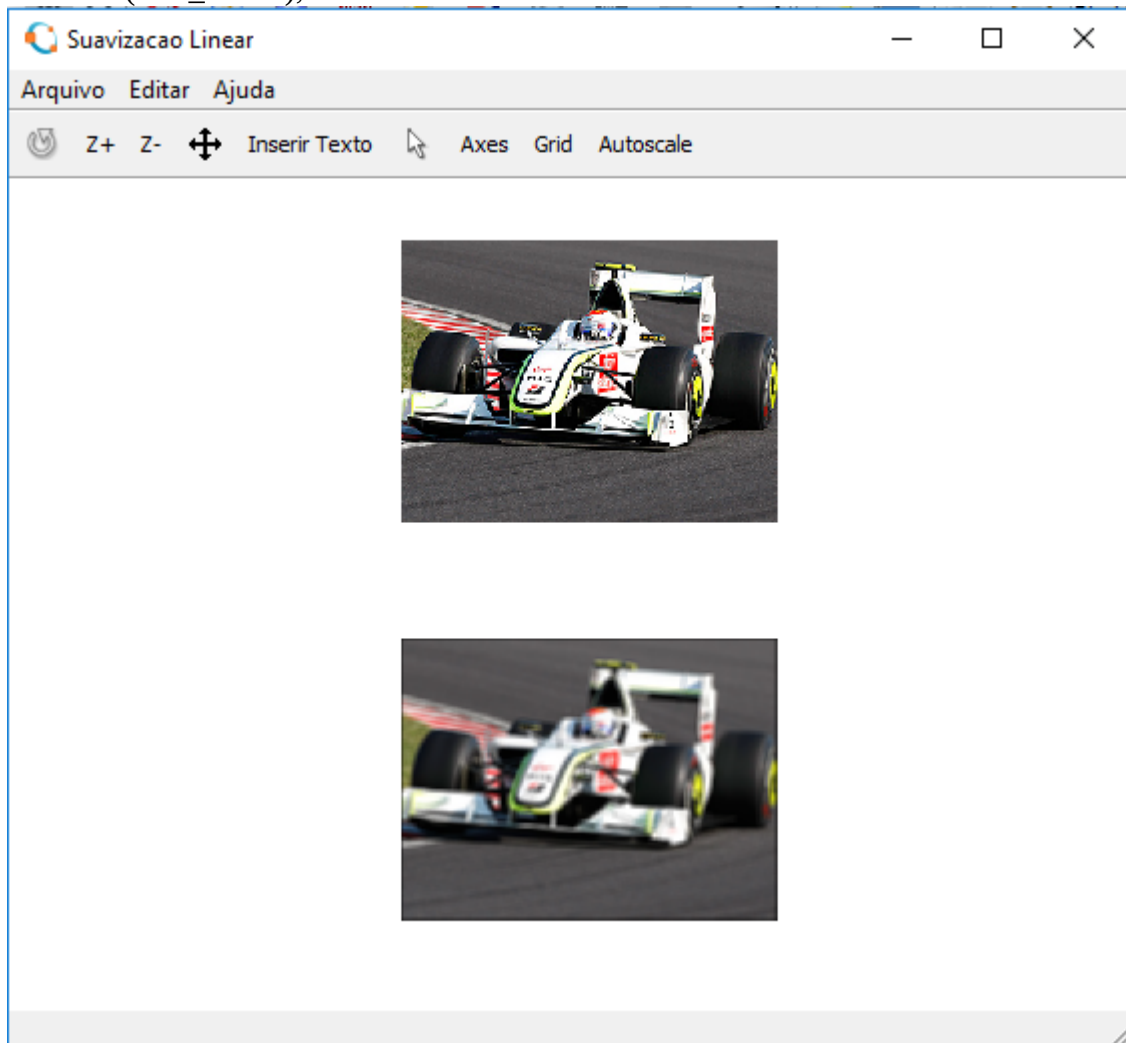
Exercício 2:

- Leia a imagem leme.bmp.
`leme = imread("leme.bmp");`
- Utilizando a função `imadjust`, crie uma nova imagem colorida em que as áreas escuras da imagem original fiquem mais claras, mas as áreas claras da imagem original não mudem muito de intensidade na nova imagem. OBS: Para utilizar a função `imadjust`, é necessário carregar o pacote de imagens com o comando `pkg load image`.
`R = leme(:,:,1);`
`G = leme(:,:,2);`
`B = leme(:,:,3);`
`L1 = imadjust(double(R)/255,[], [], 0.6);`
`L2 = imadjust(double(G)/255,[], [], 0.6);`
`L3 = imadjust(double(B)/255,[], [], 0.6);`
`new_leme = cat (3,L1,L2,L3);`
`figure('Name','Leme Edited','NumberTitle','off');`
`imshow(leme)`
`imshow(new_leme)`
- Apresente a imagem original e a nova imagem em uma única figura.

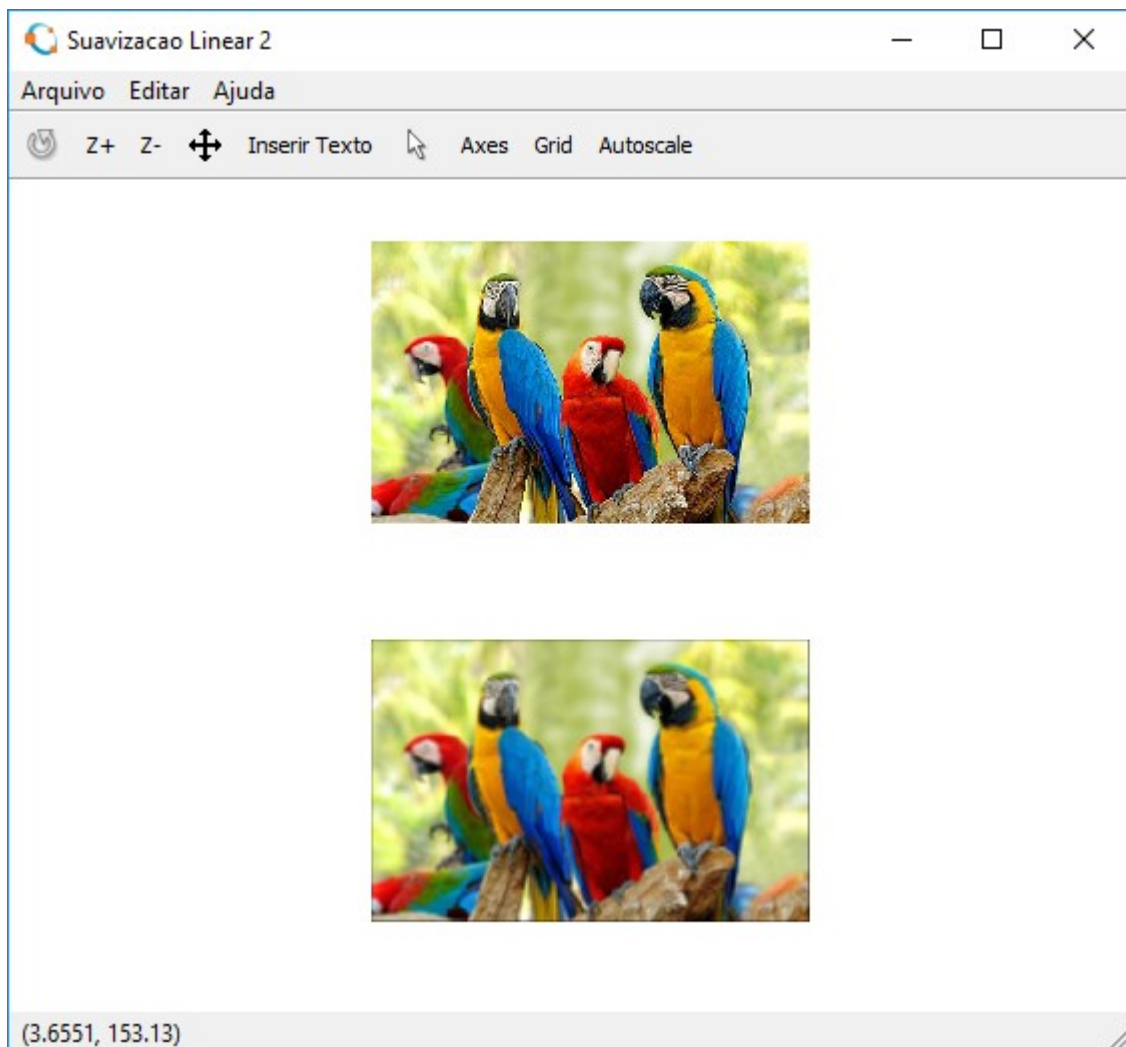


Exercício 3:

- A função `imfilter` calcula a filtragem linear da imagem `I` e do filtro `f`. A execução é feita usando números de ponto flutuante de precisão dupla, mas a classe da imagem de entrada é preservada.
- ```
brawn = imread("brawn.BMP");
f = ones(5,5)/25;
R1 = brawn(:,:,1);
G1 = brawn(:,:,2);
B1 = brawn(:,:,3);
L1_brawn = imfilter(R1,f);
L2_brawn = imfilter(G1,f);
L3_brawn = imfilter(B1,f);
new_brawn = cat(3,L1_brawn,L2_brawn,L3_brawn);
figure('Name','Suavizacao Linear','NumberTitle','off')
subplot(2,2,[1,2]);
imshow(brawn);
subplot(2,2,[3,4]);
imshow(new_brawn);
```



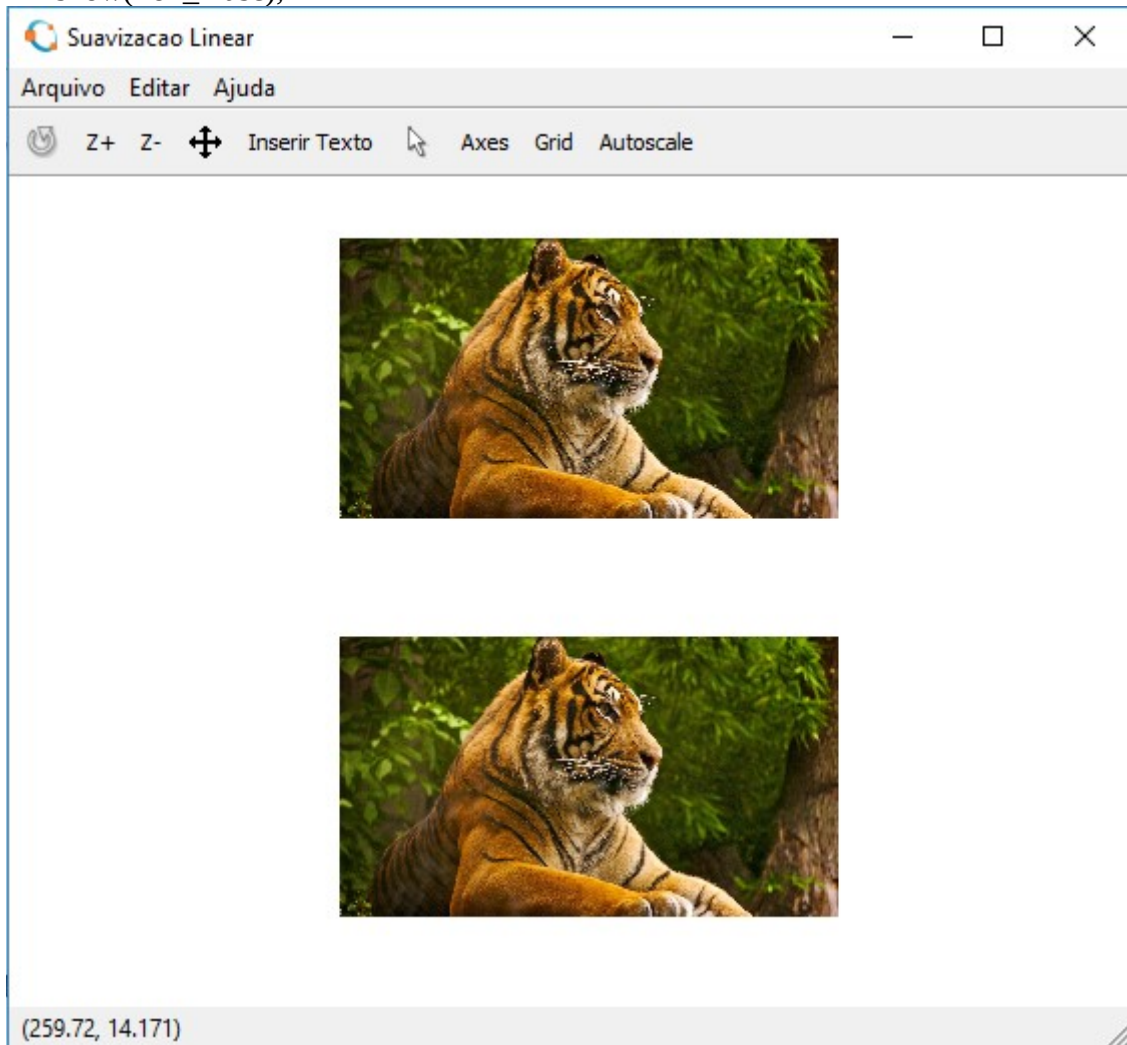
```
arara = imread("arara_full.png");
R2 = arara(:,:,1);
G2 = arara(:,:,2);
B2 = arara(:,:,3);
L1_arara = imfilter(R2,f);
L2_arara = imfilter(G2,f);
L3_arara = imfilter(B2,f);
new_arara = cat(3,L1_arara,L2_arara,L3_arara);
figure('Name','Suavizacao Linear 2','NumberTitle','off')
subplot(2,2,[1,2]);
imshow(arara);
subplot(2,2,[3,4]);
imshow(new_arara);
```



Exercício 4:

- Gaussian

```
lion = imread("lion.png");
f = fspecial("gaussian", 60);
lion_masc = imfilter(lion, f);
figure('Name','Suavizacao Linear','NumberTitle','off')
subplot(2,2,[1,2]);
imshow(lion);
subplot(2,2,[3,4]);
imshow(lion_masc);
```

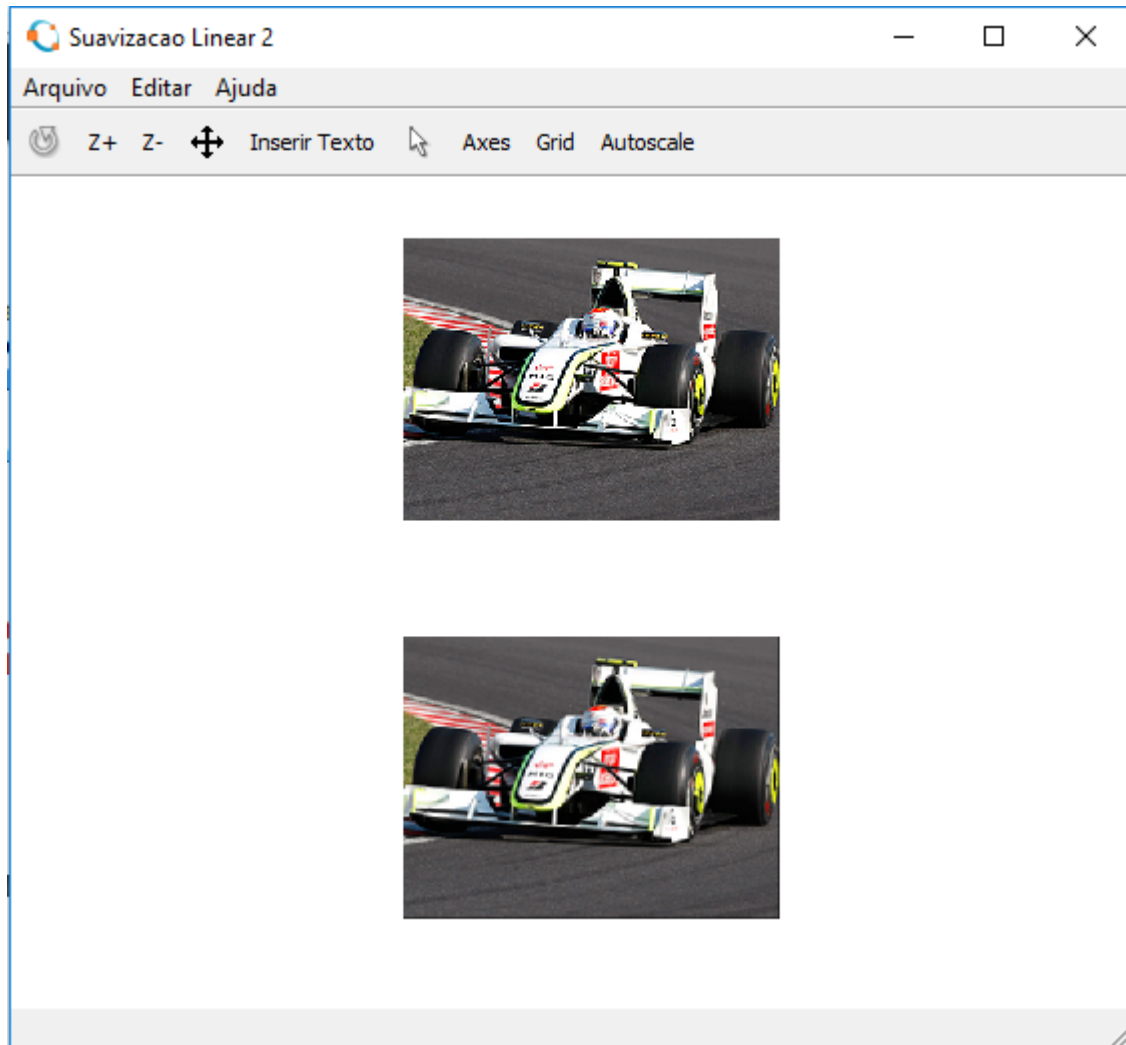




```

brawn = imread("brawn.BMP");
brawn_masc = imfilter(brawn,f);
figure('Name','Suavizacao Linear 2','NumberTitle','off')
subplot(2,2,[1,2]);
imshow(brawn);
subplot(2,2,[3,4]);
imshow(brawn_masc);

```

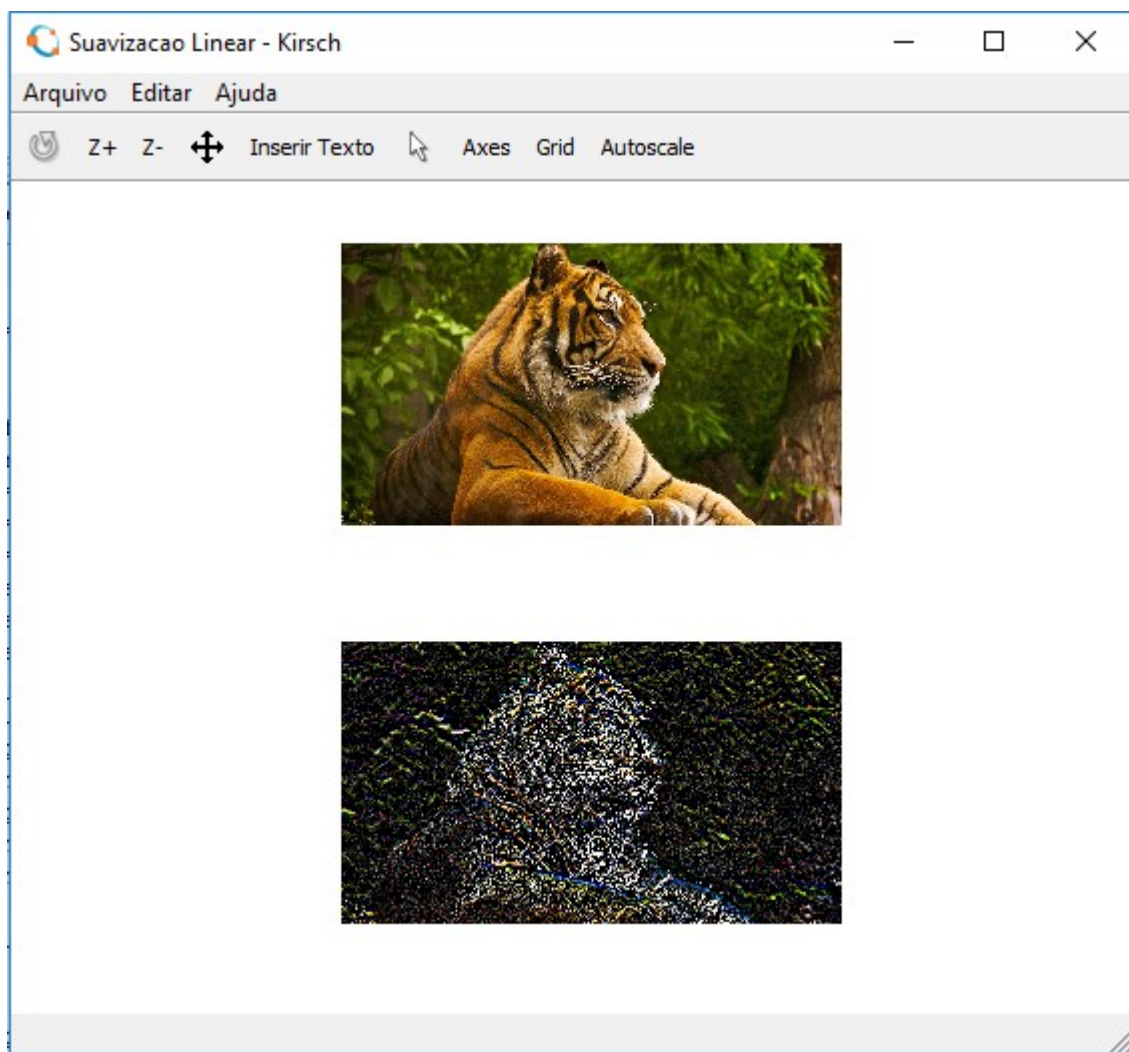


- Kirsch

```

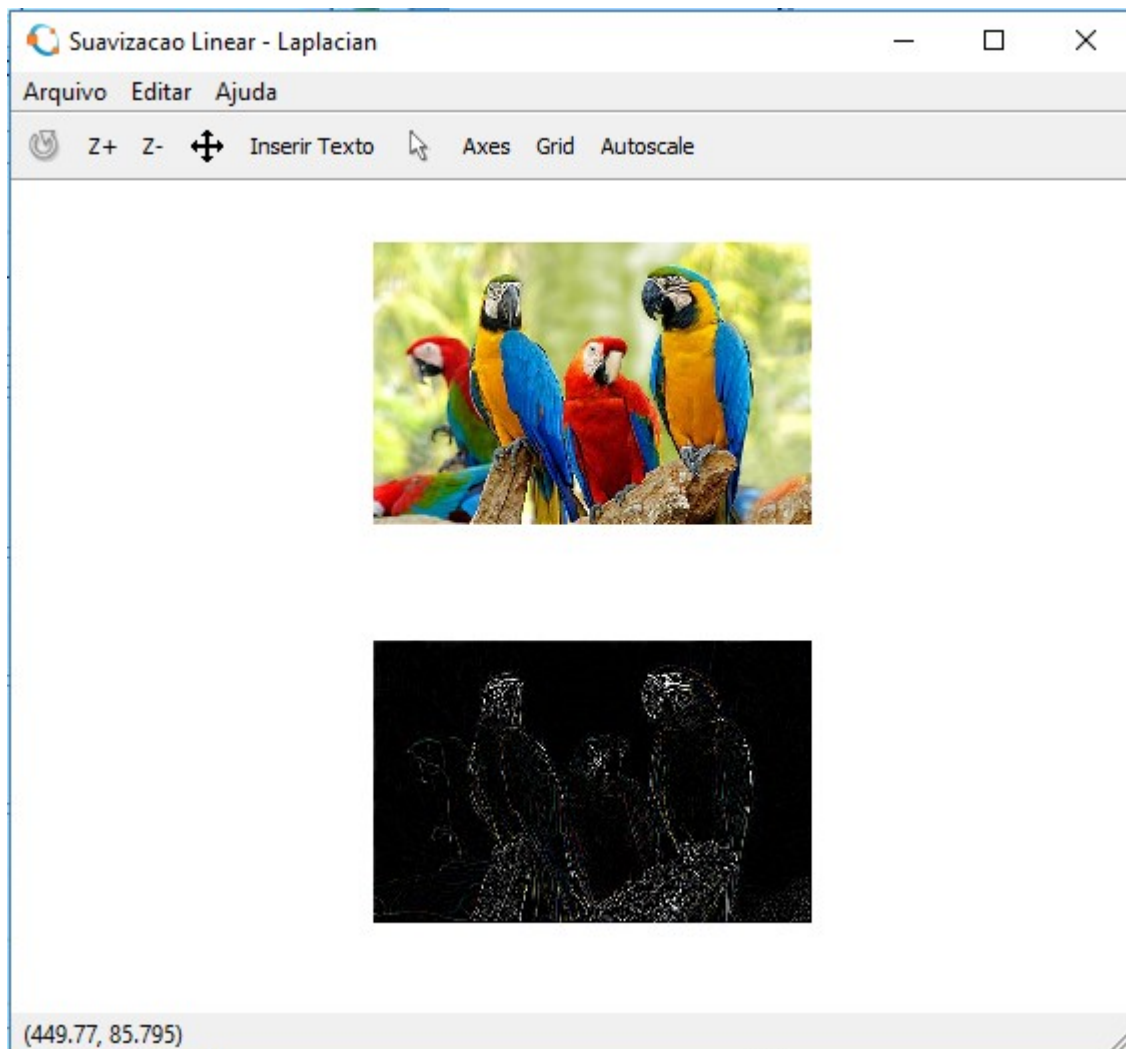
lion = imread("lion.png");
f = fspecial("kirsch");
lion_kirsch = imfilter(lion,f);
figure('Name','Suavizacao Linear - Kirsch','NumberTitle','off')
subplot(2,2,[1,2]);
imshow(lion);
subplot(2,2,[3,4]);
imshow(lion_kirsch);

```



- Laplacian

```
arara = imread("arara_full.png");
f = fspecial("laplacian",0.3);
arara_lap = imfilter(arara,f);
figure('Name','Suavizacao Linear - Laplacian','NumberTitle','off')
subplot(2,2,[1,2]);
imshow(arara);
subplot(2,2,[3,4]);
imshow(arara_lap);
```



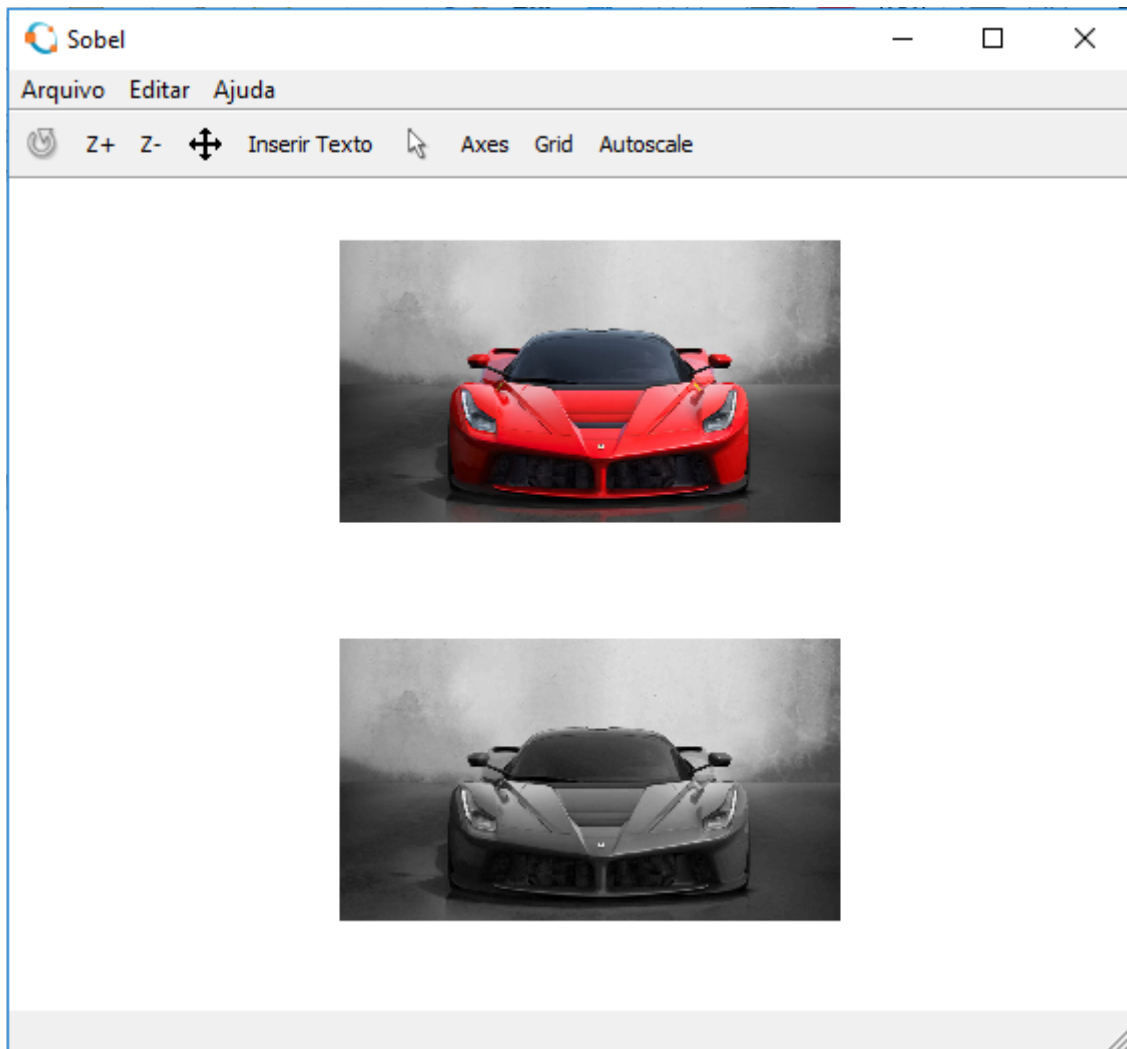
OBS: Fiz utilizando 3 máscaras pois as máscaras Laplacian e Kirsch apresentam erro quando a imagem lida é a “lion.png” ou “brawn.jpg”.

Erro:  
error: imfilter: IM must be an image  
error: called from  
imfilter at line 63 column 5

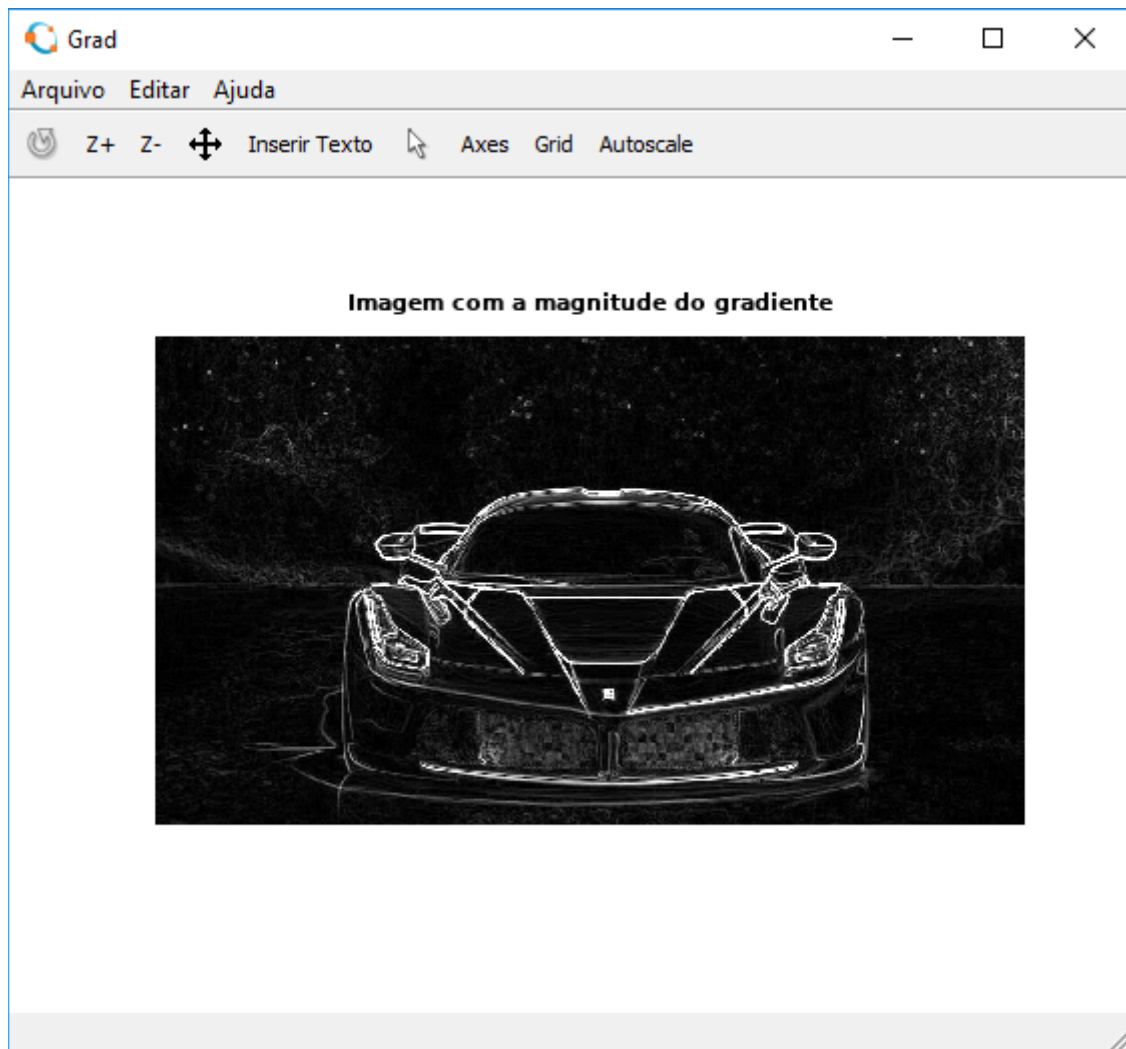
Exercício 5:

```
ferrari = imread("Ferrari.jpg");
imshow(ferrari);

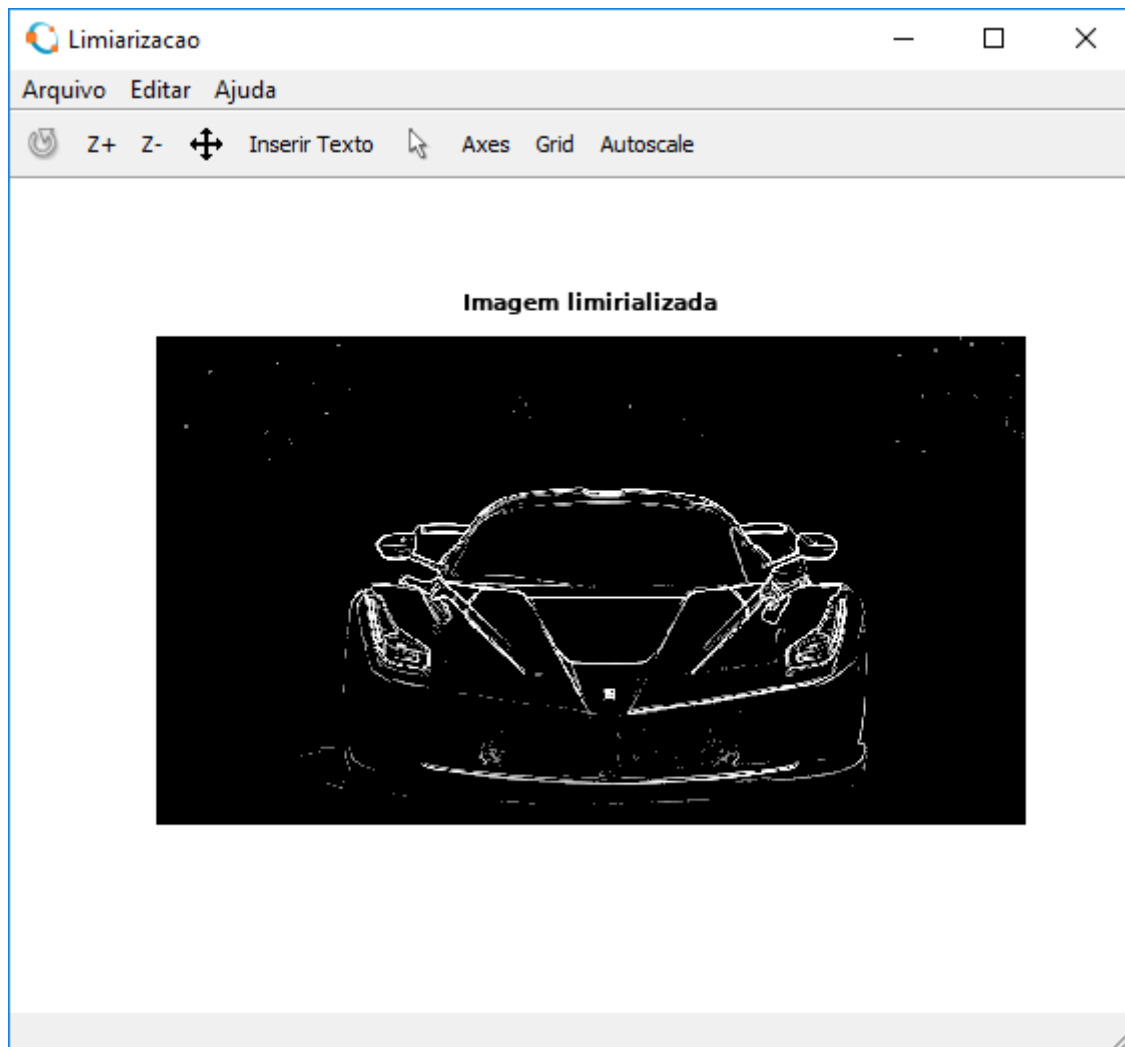
carro = rgb2gray(carro);
ferrari=im2double(ferrari);
subplot(2,2,[3,4]);
imshow(ferrari);
```



```
x=[-1 0 1;-2 0 2;-1 0 1];
y=[-1 -2 -1;0 0 0;1 2 1];
fx=imfilter(ferrari,x);
fy=imfilter(ferrari,y);
grad=abs(fx)+abs(fy);
figure('Name','Grad','NumberTitle','off')
imshow(grad);
title("Imagem com a magnitude do gradiente");
```



```
img = grad;
img(find(grad<0.4))=0;
figure('Name','Limiarizacao','NumberTitle','off')
imshow(img);
title("Imagem limirializada");
```



```

ang=atan(fx./fy);
iang = zeros(size(ang,1),size(ang,2));
iang(find(ang<(1/6)*pi & ang > -(1/6)*pi))=2;
iang(find(ang<(2/6)*pi & ang > (1/6)*pi))=3;
iang(find(ang<-(1/6)*pi & ang > -(2/6)*pi))=4;
iang(find(ang<-(2/6)*pi))=5;
iang(find(ang>(2/6)*pi))=5;
iang(find(grad>(0.4)))=0;
iang2 = iang.*img;
map=[0 0 0;1 1 0;0 0 1; 1 0 0];
figure('Name','Indexada','NumberTitle','off')
imshow(iang2,map);

```

```
title("Imagem Indexada");
```

