Home Work I

CHENG XINHAO*

Dalian University of Technology
cxh@mail.dlut.edu.cn

March 14, 2019

I. Functions of Lecture I

i. imread function

Returns a three-digit array representing the intensity values of the three colors R(red) G(green) B(blue) of the image strip.

Matlab documentation gives the following calling method:

```
A = imread(filename)
A = imread(filename,fmt)
A = imread(___,idx)
A = imread(___,Name,Value)
[A,map] = imread(___)
[A,map,transparency] = imread(___)
```

ii. imshow

In homework, I read two mammogram and ultrasound diagrams into the workspace. And use the imshow function to display the four images read.

```
im1 = imread("mammogram1.jpg");
im2 = imread("mammogram2.jpg");
im3 = imread("ultrasound1.jpg");
im4 = imread("ultrasound2.jpg");
figure;
subplot(2,2,1);
imshow(im1);
subplot(2,2,2);
imshow(im2);
subplot(2,2,3);
imshow(im3);
subplot(2,2,4);
imshow(im4);%show the four imagine ...
     in figure 1\ref{f1}
```

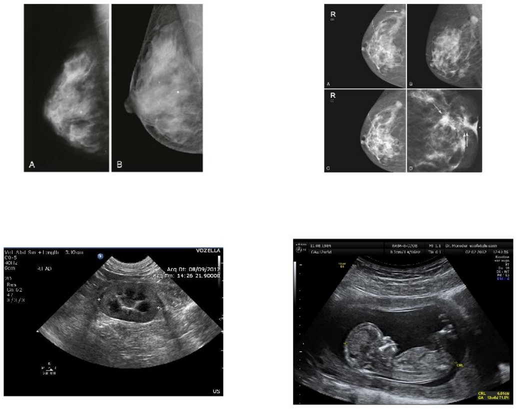


Figure 1

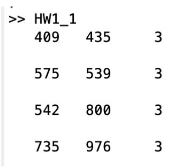
iii. size function

The Matlab documentation gives the following call methods:

- `sz = size(A)`: returns a row vector whose value is the length of each dimension.
- `szdim = size(A,dim)`:returns the length of the dim dimension.
- `[m,n] = size(A)`: m is the number of matrix rows, n is the number of matrix columns.
- `[sz1,...,szN] = size(A)`: when A is multi-dimensional, return the length of each dimension.

In the homework, use the size function to return the size of the four image matrices and print them on the screen.

*



```
>> HW1_1
   409   435     3

   575   539     3

   542   800     3

   735   976     3
```

Figure 2

```
s1 = size(im1);
s2 = size(im2);
s3 = size(im3);
s4 = size(im4);
disp(s1);
disp(s2);
disp(s3);
disp(s4);
```

iv. imadjust function

Matlab documentation gives the following calling method

```
J = imadjust(I)
j = imadjust(I,[low_in high_in])
j = imadjust(I,[low_in ...
    high_in],[low_out high_out])
j = imadjust(I,[low_in ...
    high_in],[low_out high_out],gamma)
j = imadjust(RGB,[low_in high_in],___)
Newmap = imadjust(cmap,[low_in ...
    high_in],___)
```

The pixel value of the original image f whose gray value is lower than `low_in` is assigned the value of `low_out` in the new image f1. Similarly, when a pixel whose gray value is higher than `high_in` is converted to f1, its gray value is also assigned `high_out`.

For the parameter gamma, when gamma<1, the gray value of the gray image is close to the low value of `low_in`, and the gray value of the pixel becomes higher, and the grayscale variation range is stretched, and the grayscale value is close to the grayscale variation range of the `high_in` end. Compressed, the image as a whole becomes brighter. Similarly, when gamma>1, the gray value of the gray image near `low_in` is lower, and the gray value of the pixel is lower, and the grayscale variation range is compressed, and the gray value is close to the grayscale variation range of one end of `high_in`. When stretched, the image will be darkened.

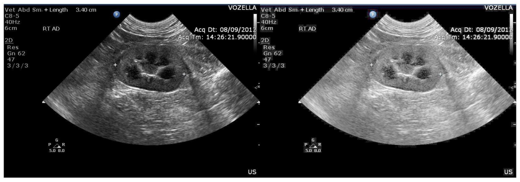


Figure 3

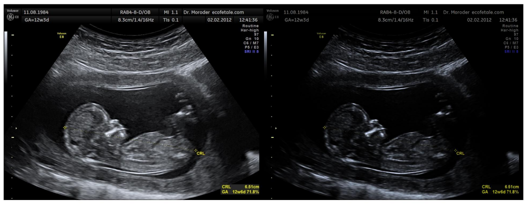


Figure 4

In homework, I used gamma=0.45 and gamma=2.2 as parameters to adjust two ultrasound images and compare them with the original image.

```
im3_1 = imadjust(im3,[],[],0.45);
figure
imshowpair(im3,im3_1,"montage");%compare ...
    the initial imagine and the ...
    transformed imagine (gamma:0.45)
im4_1 = imadjust(im4,[],[],2.2);
figure
imshowpair(im4,im4_1,"montage");%compare ...
    the initial imagine and the ...
    transformed imagine (gamma:2.2)
```

Next, we assign the values of `low_in` and `high_in` to `high_out` and `low_out` respectively to achieve the effect of negation (negative transformation).

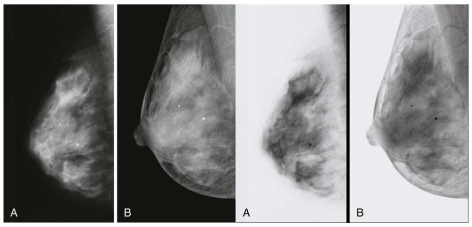


Figure 5

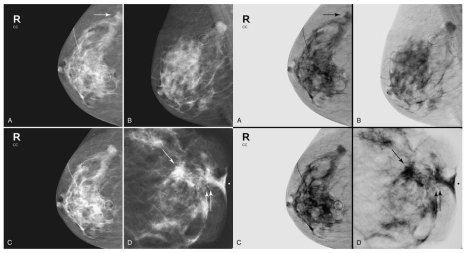


Figure 6

```
im1_1 = imadjust(im1,[0,1],[1,0]);
figure
imshowpair(im1,im1_1,"montage");%compare ...
    the initial imagine and the ...
    transformed imagine
im2_1 = imadjust(im2,[0,1],[1,0]);
figure
imshowpair(im2,im2_1,"montage");%compare ...
    the initial imagine and the ...
    transformed imagine
```

v. imcomplement function

It can also implement negative slice transformation.

```
im1_2 = imcomplement(im1);
figure
imshowpair(im1_1,im1_2,"montage");%compare ...
    the transformed ...
    imagine(imadjust) and the ...
    transformed imagine(imcomplement)
```

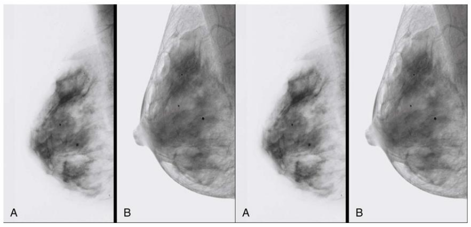


Figure 7: *Is identical to the image generated by the imadjust function*

vi. mat2gray

The format of the call given in the Matlab documentation is

```
I = mat2gray(A,[amin amax])
I = mat2gray(A)
```

The parameter meaning of the function is to normalize the data between amin and amax in the image matrix. The element smaller than amin becomes 0, and the element larger than amax becomes 1.

If you use the amin and amax parameter function formats, directly normalize to [0,1]

In homework, I read in a picture downloaded from the network and converted the value of the matrix element with mat2gray.

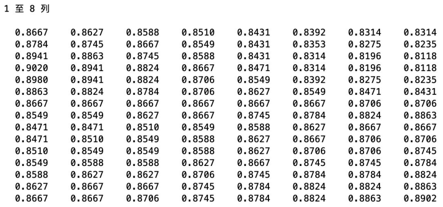
```
Im1 = imread("experiment.jpg");
im1_1 = mat2gray(im1,[200,300]);
```

Similar to the use of mat2gray, there is also the `im2double` function, which maps the value of the element between [0, 255] to [0, 1], and converts the data type into double.

```
im1_2 = im2double(im1);
```

vii. im2uint8 function

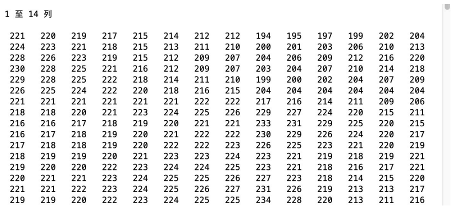
This function converts an image data type to an unsigned eight-bit integer. If the input im-



```
1 至 8 列

  0.8667    0.8627    0.8588    0.8510    0.8431    0.8392    0.8314    0.8314
  0.8784    0.8745    0.8667    0.8549    0.8431    0.8353    0.8275    0.8235
  0.8941    0.8863    0.8745    0.8588    0.8431    0.8314    0.8196    0.8118
  0.9020    0.8941    0.8824    0.8667    0.8471    0.8314    0.8196    0.8118
  0.8980    0.8941    0.8824    0.8706    0.8549    0.8392    0.8275    0.8235
  0.8863    0.8824    0.8784    0.8706    0.8627    0.8549    0.8471    0.8431
  0.8667    0.8667    0.8667    0.8667    0.8667    0.8667    0.8706    0.8706
  0.8549    0.8549    0.8627    0.8667    0.8745    0.8784    0.8824    0.8863
  0.8471    0.8471    0.8510    0.8549    0.8588    0.8627    0.8667    0.8667
  0.8471    0.8510    0.8549    0.8588    0.8627    0.8667    0.8706    0.8706
  0.8510    0.8549    0.8549    0.8588    0.8627    0.8706    0.8706    0.8745
  0.8549    0.8588    0.8588    0.8627    0.8667    0.8745    0.8745    0.8784
  0.8588    0.8627    0.8627    0.8706    0.8745    0.8784    0.8784    0.8824
  0.8627    0.8667    0.8667    0.8745    0.8784    0.8824    0.8824    0.8863
  0.8667    0.8667    0.8706    0.8745    0.8784    0.8824    0.8863    0.8902
```

Figure 8



```
1 至 14 列

  221   220   219   217   215   214   212   212   194   195   197   199   202   204
  224   223   221   218   215   213   211   210   200   201   203   206   210   213
  228   226   223   219   215   212   209   207   204   206   209   212   216   220
  230   228   225   221   216   212   209   207   203   204   207   210   214   218
  229   228   225   222   218   214   211   210   199   200   202   204   207   209
  226   225   224   222   220   218   216   215   204   204   204   204   204   204
  221   221   221   221   221   221   222   222   217   216   214   211   209   206
  218   218   220   221   223   224   225   226   229   227   224   220   215   211
  216   216   217   218   219   220   221   221   233   231   229   225   220   215
  216   217   218   219   220   221   222   222   230   229   226   224   220   217
  217   218   218   219   220   222   222   223   226   225   223   221   220   219
  218   219   219   220   221   223   223   224   223   221   219   218   219   221
  219   220   220   222   223   224   224   225   223   221   218   216   217   221
  220   221   221   223   224   225   225   226   227   223   218   214   215   220
  221   221   222   223   224   225   226   227   231   226   219   213   213   217
  219   219   220   222   223   224   225   225   234   228   220   213   211   216
```

Figure 9

age is an unsigned eight-bit integer, the returned image is the same as the source image. If the source image is not an unsigned eight-bit integer, the function returns the same image as the source image but with a data type of uint8 (adjust the image if necessary) Call method.

J = im2uint8(I) J = im2uint8(I,indexed)

In homework, we used the im2uint8 function to transform

```
Im1_3 = im2uint8(im1);
```

Logarithmic transformation G = c*log(1+double(f)) c is a constant, and the transformation range can be reduced by logarithmic transformation Gs = im2uint8(mat2gray(g)) Contrast stretch transformation function, the most accepted value type is unit8

viii. Imresize

It is is used to scale the image This is the calling method given in the Matlab documentation.

```
This is the calling method given in ...
    the Matlab documentation.
```

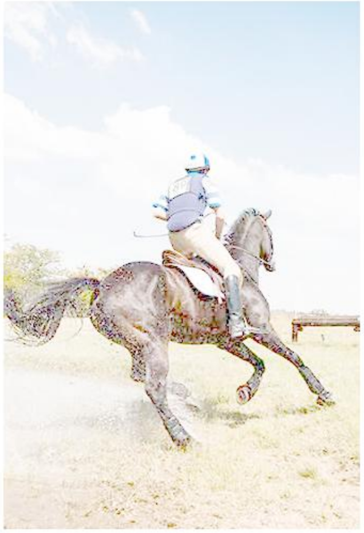


Figure 10

```
B = imresize(A,scale)
B = imresize(A,[numrows numcols])
[Y,newmap] = imresize(X,map,___)
___ = imresize(___,method)
___ = imresize(___,Name,Value)
```

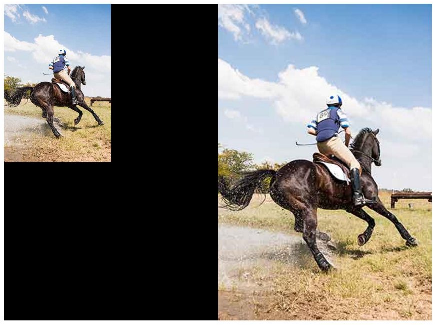


Figure 11