Xi'an Jiaotong-Liverpool University
西交利物浦大學

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE330 lab report

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Engineering

Student Name : Lingxuan Kong
Student ID : 1405867

Contents

1 Introduction **2**

2 Task 1 **3**

2.1 Imread and Image 3

2.2 colormap 3

2.3 turesieze display 4

2.4 cool hot gray 5

3 Task 2 **8**

3.1 Crop the face of the image 8

3.2 Sampling with nearest neighbor interpolation 8

3.2.1 Down-sampling 8

3.2.2 Up-sampling 10

3.3 Sampling with bilinear interpolation 11

3.3.1 Down-sampling 11

3.3.2 Up-sampling 12

3.3.3 Finding 13

3.4 PSNR 13

3.5 Quantization 14

4 Conclusion **16**

1 Introduction

Based on the previous lectures, we have already learned about how to load and read imagines in matlab, and operate some basic functions about sample and quantities. In report, I will provide some basic functions, which concerned about how to load and read the images in matlab, it will also include all the codes and results. Additional, this report will also discuss the influence about some basic operations, for example, up-sampling, down-sampling and quantization, and analysis the advantages and disadvantages about these operations.

2 Task 1

2.1 Imread and Image

Using the following codes to show the photo:

```
A=imread('lenna512.bmp');%write the information matrix to A
image(A)%display image of A
```

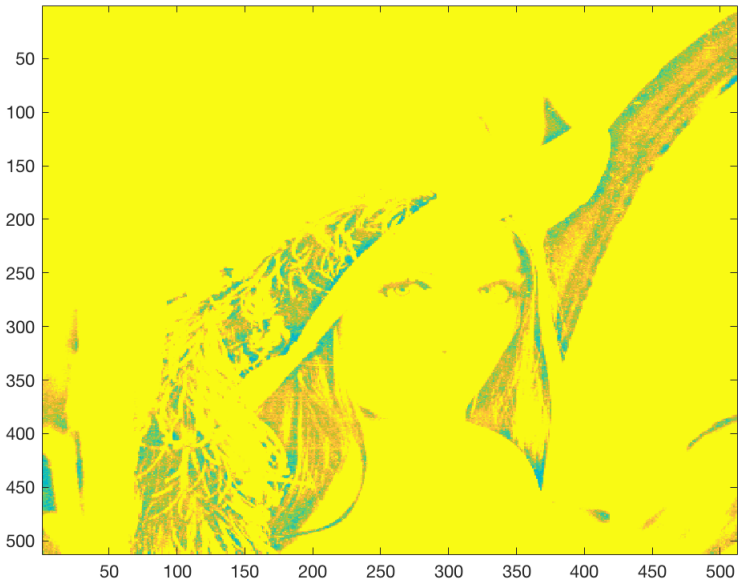


Figure 1: The photo showed by image

The original photo is a grey image. However, this photo is a yellow photo. This problem will be solved in next section.

2.2 colormap

From the previous section, we could find that the color of this photo is incorrect, thus, we should find another method to display this grey photo.

```
A=imread('lenna512.bmp');
image(A);
```

```
colormap(gray(256))%define the colormap, gray of 256 bits
axis on;%turn axis
colorbar;%turn color bar
```

Figure 2: The photo showed with colormap function

2.3 turesieze display

The turesize function could adjust the size of the figure window to arbitrary dimension. Here show the codes:

```
A=imread('lenna512.bmp');
image(A);
truesize([200,400])%redefine the size of this image
```

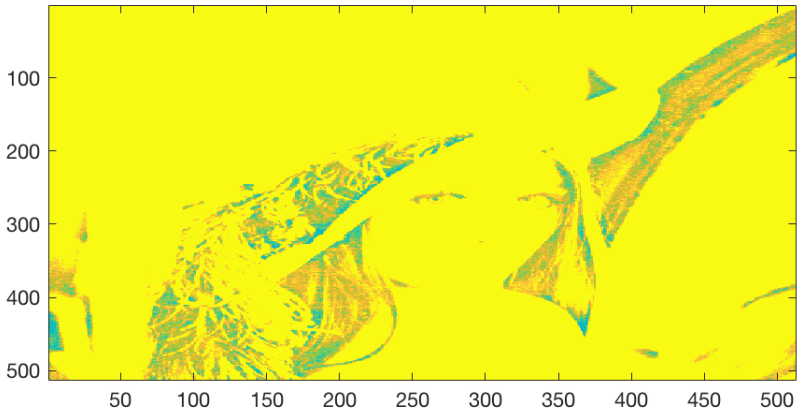


Figure 3: The truesieze photo

2.4 cool hot gray

cool(256):

```
A=imread('lenna512.bmp');
image(A);
colormap(cool(256))%define the colormap as cool of 256 bits
axis on;
colorbar;
```

hot(255):

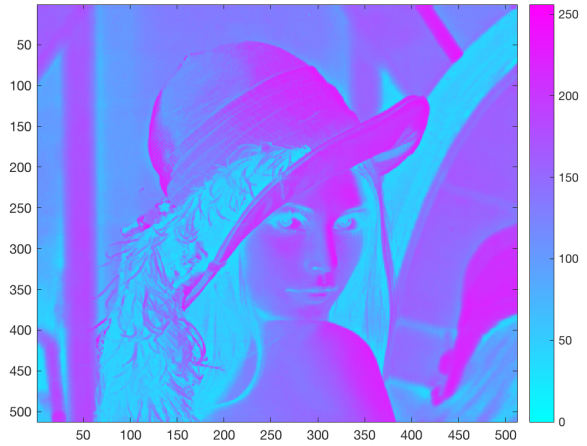
```
A=imread('lenna512.bmp');
image(A);
colormap(hot(255))%colormap as 255 bits
axis on;
colorbar;
```

gray(128):

```
A=imread('lenna512.bmp');
image(A);
colormap(gray(128))%colormap of 128 bits
axis on;
colorbar;
```

gray(64):

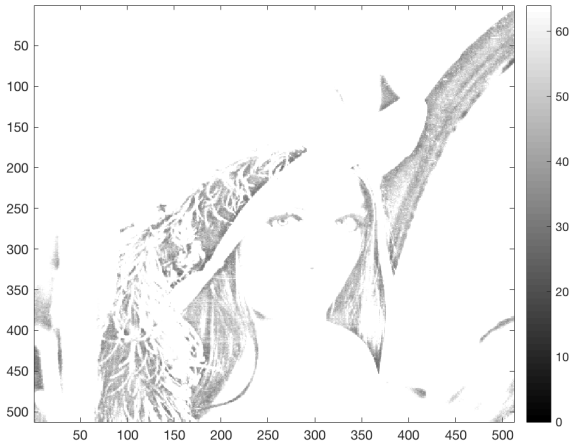
```
A=imread('lenna512.bmp');
image(A);
colormap(gray(64))%colormap of 64 bits
axis on;
colorbar;
```



(a) cool(256)

(b) hot(255)

(c) gray(128)



(d) gray(64)

Figure 4: colormap with different color type

According to the previous photos, we could draw some significant conclusions:

- Different color type will change the color tone of the color. Foe example, when we use the hot(255), the color of this photo could be considered as the warm tone, like red and orange. However, if we use cool(256), this photo will display the cool tone, like ice blue.
- The original photo is a gray photo. Therefore, when we change the colormap of gray, the gray level of this photo will also change.
- When we use gray(128), we could easily find that the gray level of this photo is much deeper. For example, the color of this girl's hair is black, not gray. However, if we use gray(64). It could be found that the gray level of this photo become much shallower. The color of this girl's hair is white and gray. In another word, this photo seems to be overexposed.
- When we compare the last two photos, we can find that the first photo looks a little bit nature and comfortable, because the gray level is enough high. The gray level of the second photo is low, which could make this photo looks white.

3 Task 2

3.1 Crop the face of the image

In this task, the student should use the function *datacursormode on* to obtain some useful information on this image, which should be coordinate. Based on these coordinates, we could display another image, includes the face. Then, we use another function, *imcrop* to display the image after cut.

$$imcrop(I, [a\ b\ c\ d])$$

- a b: The pixel coordinate on the upper left corner.
- c: The width of image after cropped.
- d: The height of image after cropped.

The value of previous [a b c d] could obtain from *datacursoemode on* function. Then, we could the following codes:

```
A=imread('lenna512.bmp');
datacursormode on%obtain the coordinate
B=imcrop(A,[200 210 160 170]);%get the image information after cutted
imshow(B)%show the cutted image
```

Figure 5: The image after crop

3.2 Sampling with nearest neighbor interpolation

3.2.1 Down-sampling

In this down-sampling section, I will provide two separate methods, the first on is that changing the value of line and row. The second one is using imresize.

```
A=imread('lenna512.bmp');
[line,row]=size(A);%write the information of line and row to matrix A
A_NEW=A(1:2:row, 1:2:line);%sample the matrix line and row
imwrite(A_NEW,'down_sample.png')%write the new image after sampled.
imshow('down_sample.png')%display the image of sampled.
```

Then, another matlab function will be used to sampled:

$$imresize(A, SCALE, METHOD)$$

METHOD can be a string naming a general interpolation method:

- 'nearest' - nearest-neighbor interpolation
- 'bilinear' - bilinear interpolation
- 'bicubic' - cubic interpolation

```
A=imread('lenna512.bmp');
B = imresize(A,0.5,'nearest');%define the sample type is nearest
imshow(B)%display the image after sampled
axis on;
```

The previous two methods display the same photo, which has been down-sampled.

Figure 6: Original photo

Figure 7: Down-sampling photo

Based on these two photos, we could obtain some important conclusions:

- Down-sampling is the process of changing the resolution of any image. For example, the resolution of the original photo is 256×256, after down-sampling, the resolution will become 128×128.
- After down-sampling, the resolution of this photo is decreased, from 512×512 to 256×256. Additional, the size of this photo is also decreased.
- If we only change the size of line and row of this photo. It is same as using imresize function in matlab.
- After down-sampling, we only change the resolution and size of photo. The gray level and color did not change.

3.2.2 Up-sampling

Using the same method of down-sampling, show the following codes:

```
A=imread('down_sample.png');
B = imresize(A,2,'nearest');%use nearest sampled
imshow(B)
```

Figure 8: Down-sampling photo

Figure 9: Up-sampling photo

Then we compare the original photo and up-sampling photo.

Figure 10: Original photo

Figure 11: Up-sampling photo

- Sample only change the resolution of photos. Up-sampling increase the resolution the photo from 256×256 to 512×512.
- However, if we compare the up-sampling photo and original photo. We could find some differences.
- These two photos have the same resolution and same size. But if we look it carefully, we could find that the up-sampling photo is not clear as the original one. Because after sampling, the photo lose some information and significant pixels. Thus, it looks full with 'mosaic'.

3.3 Sampling with bilinear interpolation

3.3.1 Down-sampling

```
1 A=imread('lenna512.bmp');
2 B = imresize(A,0.5,'bilinear');%use bilinear sample method
3 imshow(B)
```

Figure 12: Original photo

Figure 13: Down-sampling photo

3.3.2 Up-sampling

```
A=imread('down_li.png');
B = imresize(A,2,'bilinear');
imshow(B)
```

Figure 14: Down-sampling photo

Figure 15: Up-sampling photo

3.3.3 Finding

Figure 16: Original photo

Figure 17: Up-sampling photo

- Sampling only influence the resolution or the size of photo, it does not change the gray level and the color.
- In nearest neighbor interpolation, when we compare the up-sampling and original photo, we could find that the up-sampling photo seems fulled with mosaic. It is not clear as the original photo.
- In bilinear interpolation, when we compare figure 15 and figure 16, which are the original photo and up-sampling photo. We could find that the up-sampling photo is a little bit 'virtual'. In another word, the up-sampling photo is not real as the original one.
- No matter what kind of interpolation we use. The photo after down and up-sampling, the photo is not clear as the original photo. In another word, after interpolation, the pixel will be effected when we compare with the original one.

3.4 PSNR

Peak signal-to-noise ratio, often abbreviated PSNR, is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Typical values for the PSNR in lossy image and video compression are between 30 and 50 dB, provided the bit depth is 8 bits, where higher is better.
We first calculate the psnr for nearest neighbor interpolation:

```
A=imread('lenna512.bmp');
B = imresize(A,0.5,'nearest');
C = imresize(B,2,'nearest');
Upsample_Original_PSNR=psnr(C,A);%calculate the psnr after nearest ...
    sampled
```

$$psnr = 20 \cdot log(MAX) - 10 \cdot log(MSE) = 28.3\ dB$$

We first calculate the psnr for bilinear interpolation:

```
A=imread('lenna512.bmp');
B = imresize(A,0.5,'bilinear');
C = imresize(B,2,'bilinear');
Upsample_Original_PSNR=psnr(C,A);
```

$$psnr = 20 \cdot log(MAX) - 10 \cdot log(MSE) = 31.4\ dB$$

Then we compare the results,

$$PSNR_{bilinear} = 31.4 \ > \ PSNR_{nearest} = 28.3$$

- Higher PSNR generally indicates that the reconstruction is of higher quality. Bilinear interpolation is better than nearest neighbor interpolation, with its higher psnr.
- PSNR is usually expressed in terms of the logarithmic decibel scale.
- When we compare with figure 14 and figure 16, we could find that the photo with high psnr is much clearer than the photo with low psnr. However, compare with original photo, they all have some lost in different range.

3.5 Quantization

```
A=imread('lenna512.bmp');
G=A./16;%The matrix G should be A./ 16, resize the pixel
image(G);
axis on;
colormap(gray(16))
colorbar;
```

Figure 18: gray level 256

Figure 19: gray level 16

According to these two photos, which have different gray levels, we could obtained some important conclusions:

- When photo with high quantization level. It will have a richer image level and details. In addition, it has a higher grey level resolution and a better image quality. However, it has a larger size and data to store.
- When photo with low quantization level. It will have a poorer image level and less details. Additional, it has a lower grey level resolution and bad image quality. The flase contour may appear because its low level quantization. However, it has a smaller size and data to store.
- Obviously, the quality of digital image depends on its resolution and grey level. When it have higher gray level and high resolution, it will display much details and information. The data will also increase.

4 Conclusion

To conclude, after this computer lab, we should have command of some basic operations on matlab, for example, sample and quantization on digital images. According to this operations, we could draw some significant conclusions:

- *imread* function could load the image as a matrix in matlab, and *image* function could display the image that has been read.
- *colormap* function can make a difference on primary hue of the whole picture. For example, cool(256) and hot(255), will make the different color of the pictures. In another word, changing the *colormap* could be considered as changing color filter of this picture.
- We can use some function *datacursormode* and *imcrop* to cut the image as we want. In another word, we could custom the size of the iamge and display it.
- Both sample and quantification are the basic operations for digital images. When the size of images are limited, we should take different operations for processing images. For example, if there are quantified of details in image, we should use sample instead of quantification, in order to avoid mosaic' and unclear. Similarly, if there are less details in the image, we should use quantification, but sample, which could avoid false contour.
- When people want to use less size to save much more information on images, they have to lose the quality of the images. Similarly, the size of image would be large if there are much more details and high quality.