المستوي الرابع- شعبة الحاسب الالي
نظام الساعات المعتمدة
كلية العلوم
الفصل الدراسى الاول 2015- 2016 م
تاريخ الامتحان: 28 / 12/ 2015

نموذج اجابة –ورقة كاملة
المـــــادة: معالجة الصور-366 رس
اسم استاذ المادة: الدكتور/ عبدالحميد محمد عبدالحميد –
جامعة بنها – كلية العلوم – قسم الرياضيات

Department of Mathematics | Final Exam | Time: 2 Hours

Fac. of Science, Benha Univ. | IMAGE PROCESSING (366 MC) | 28 December 2015

Please answer all the following questions. Total Marks = 48 points:-

---

Question 1:

1) What are the types of digital images? Explain your answer?

2) What is Histogram Equalization?

3) The following table gives the number of pixels at each of the grey levels 0-15 in an image those values only:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
20	40	60	75	80	75	65	55	50	45	40	35	30	25	20	30

draw the histogram corresponding to these grey levels and then perform a histogram equalization and draw the resulting histogram.

Question 2:

1) What is meant by filter and linear filter for an image?

2) What are the steps to implement the spatial filtering to an image?

3) What happens at the the edge of the image, where the mask partly falls outside the image? Explain how do you deal with such case?

4) The array below represents a small grey scale image

$$\begin{bmatrix} 170 & 240 & 10 & 80 & 150 \\ 230 & 50 & 70 & 140 & 160 \\ 40 & 60 & 130 & 200 & 220 \\ 100 & 120 & 190 & 210 & 30 \\ 110 & 180 & 250 & 20 & 90 \end{bmatrix},$$

compute the image values that result when the image is convolved with the following mask:

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Question 3:

1) State **three** types of noise to the image and explain them.

2) Explain Pratt's approach for cleaning the salt and pepper noise for an image.

3) Write the `MATLAB` syntaxes to perform the following operations on the given images:

Figure 1: 64×64 resolution.

Figure 2: Average filtering.

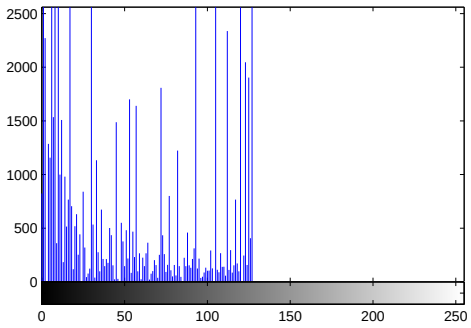

Figure 3: The image trees.tif and its histogram.

4) What the following functions `imadjust`, `imadd`, `im2unit8`, `fspecial`, `filter2` do?

With My Best Wishes

Dr. A. M. Nagy

Model Answer

Question 1:

1- Types of digital images:

i) **Binary**: Each pixel is just black or white. Since there are only two possible v alues for each pixel, we only need one bit per pixel. Such images can therefore be very efficient in terms of storage. Images for which a binary representation may be suitable include text (printed or handwriting), fingerprints, or architectural plans.

ii) **Greyscale**: Each pixel is a shade of grey , normally from 0 (black) to 255 (white). This range means that each pixel can be represented by eight bits, or exactly one byte. This is a very natural range for image file handling. Other greyscale ranges are used, but generally they are a power of 2. Such images arise in medicine (X-rays), images of printed works, and indeed 256 different grey levels is sufficient for the recognition of most natural objects.

iii) **True colour**, or **RGB**: Here each pixel has a particular colour; that colour being described by the amount of red, green and blue in it. If each of these components has a range 0-255, this gives a total of $255^3 = 16581375$ different possible colours in the image. This is enough colours for any image. Since the total number of bits required for each pixel is 24, such images are also called 24-bit colour images.

v) **Indexed**: Most colour images only have a small subset of the more than sixteen million possible colours. For convenience of storage and file handling, the image has an associated colour map, or colour palette, which is simply a list of all the colours used in that image. Each pixel has a value which does not give its colour (as for an RGB image), but an index to the colour in the map.

2- Histogram equalization is a technique for adjusting image intensities to enhance contrast.

Suppose our image has L different grey levels 0, 1, 2, $\ldots, L-1$ and that grey level i occurs n_i times in the image. Suppose also that the total number of pixels in the image is n (so that $n_0 + n_1 + n_2 + \cdots + n_{L-1} = n$. To transform the grey levels to obtain a better contrasted image, we change grey level i to

$$\left(\frac{n_0 + n_1 + \cdots + n_i}{n}\right)(L-1),$$

and this number is rounded to the nearest integer.

3- The histogram corresponding to the grey levels given is:

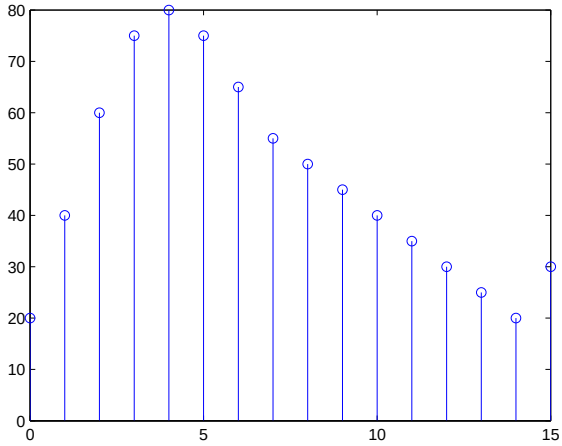

Figure 4: The histogram before equalization.

We now have the following transformation of grey values, obtained by reading off the first and last columns in the above table:

Original grey level i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Final grey level i	0	1	2	4	6	7	8	9	10	11	12	13	13	14	14	15

The histogram corresponding to the final grey levels given is:

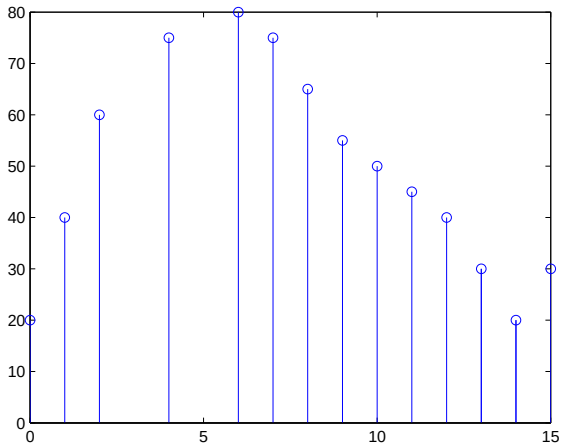

Figure 5: The histogram after equalization.

Question 2:

Grey level i	n_i	$\sum n_i$	$(15/745)\sum n_i$	Rounded value
0	20	20	0.4027	0
1	40	60	1.2081	1
2	60	120	2.4161	2
3	75	195	3.9262	4
4	80	275	5.5369	6
5	75	350	7.0470	7
6	65	415	8.3557	8
7	55	470	9.4631	9
8	50	520	10.4698	10
9	45	565	11.375	11
10	40	605	12.1812	12
11	35	640	12.8859	13
12	30	670	13.4899	13
13	25	695	13.9933	14
14	20	715	14.3960	14
15	30	745	15	15

1-

The combination of mask and function is called a *filter*.

If the function by which the new grey value is calculated is a linear function of all the grey values in the mask, then the filter is called a *linear filter*.

2- The spatial filtering requires three steps:

1. position the mask over the current pixel,

2. form all products of filter elements with the corresponding elements of the neighbourhood,

3. add up all the products.

This must be repeated for every pixel in the image.

3- When the mask partly falls outside the image, there will be a lack of grey values to use in the filter function.

There are a number of different approaches to dealing with this problem:

1- **Ignore the edges**: That is, we only apply the mask to those pixels in the image for with the mask will lie fully within the image. This means all pixels except for the edges, and results in an output image which is smaller than the original. If the mask is very large, we may lose a significant amount of information by this method.

2- **Pad with zeros**: We assume that all necessary values outside the image are zero. This gives us all values to work with, and will return an output image of the same size as the original, but may have the effect of introducing unwanted artifacts (for example, edges) around the image.

4- In order to implement the mask on the values of image, first we Pad the matrix with zeros as follows:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 170 & 240 & 10 & 80 & 150 & 0 \\ 0 & 230 & 50 & 70 & 140 & 160 & 0 \\ 0 & 40 & 60 & 130 & 200 & 220 & 0 \\ 0 & 100 & 120 & 190 & 210 & 30 & 0 \\ 0 & 110 & 180 & 250 & 20 & 90 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

Now, we compute the new values of the image as follows:

$$0*0.1111+0.1111*0+0.1111*0+0.1111*0+0.1111*170+0.1111*240+0.1111*0+...$$

$$0.1111 * 230 + 0.1111 * 50 = 76.6667$$

and so on, we finally obtain the following values:

$$\begin{bmatrix} 76.6667 & 85.5556 & 65.5556 & 67.7778 & 58.8889 \\ 87.7778 & 111.1111 & 108.8889 & 128.8889 & 105.5556 \\ 66.6667 & 110.0000 & 130.0000 & 150.0000 & 106.6667 \\ 67.7778 & 131.1111 & 151.1111 & 148.8889 & 85.5556 \\ 56.6667 & 105.5556 & 107.7778 & 87.7778 & 38.8889 \end{bmatrix}.$$

Question 3:

1- The three types of noise are:

i) **Salt and pepper noise**: Also called *impulse noise*, *shot noise*, or *binary noise*. This degradation can be caused by sharp, sudden disturbances in the image signal; its

appearance is randomly scattered white or black (or both) pixels over the image.
To add noise, we use the Matlab function imnoise, which takes a number of different parameters. To add salt and pepper noise:

```
>> t_sp=imnoise(t,'salt & pepper');
```

ii) **Gaussian noise**: Gaussian noise is an idealized form of white noise, which is caused by random fluctuations in the signal. We can observe white noise by watching a television which is slightly mistuned to a particular channel. Gaussian noise is white noise which is normally distributed. If the image is represented as I, and the Gaussian noise by N, then we can model a noisy image by simply adding the two:

$$I + N.$$

Here we may assume that is I a matrix whose elements are the pixel values of our image, and N is a matrix whose elements are normally distributed. It can be shown that this is an appropriate model for noise. The effect can again be demonstrated by the imnoise function:

```
>> t_ga=inoise(t,'gaussian');
```

3- **Speckle noise**: speckle noise can be modelled by random values multiplied by pixel values, hence it is also called multiplicative noise. Speckle noise is a major problem in some radar applications. As above, imnoise can do speckle:

```
>> t_spk=inoise(t,'speckle');
```

In Matlab, speckle noise is implemented as

$$I(1 + N),$$

where I is the image matrix, and N consists of normally distributed values with mean 0.
2- Pratt has proposed the use of cleaning salt and pepper noise by treating noisy pixels as outliers ; that is, pixels whose grey values are significantly different from those of their neighbours. This leads to the following approach for noise cleaning:
1. Choose a threshold value D.
2. For a given pixel, compare its value p with the mean m of the values of its eight

neighbours.

3. If $|p - m| > D$, then classify the pixel as noisy, otherwise not.

4. If the pixel is noisy, replace its value with m; otherwise leave its value unchanged.

3-

a)

```
>> I=imread('cameraman.tif');
>> I1=imresize(imresize(I,1/8),8);
>> imshow(I1)
```

b)

```
>> c=imread('cameraman.tif');
>> f1=fspecial('average');
>> cf1=filter2(f1,c);
>> imshow(cf1/255)
```

c)

```
>> I=imread('trees.tif');
>> figure; imshow(I);
>> figure, imhist(I);
```

3-

Function	Description
`imadjust`	Adjust image intensity values or colormap
`imadd`	Add two images or add constant to image
`im2unit8`	Brings the values back to the range [0,255]
`fspecial`	Create predefined 2-D filter
`filter2`	2-D digital filter