**Digital Image Processing**

**HW2**

**Blind Super Resolution**

Yannay Alon 322227216  
Tomer Gavriel 322230392

Question 1

From definition of convolution using the notations from the third paragraph:

Question 2

From down sampling and convolution definition:

Question 3

By plugging in the results from q1 and q2 we get:  
By change of variables :By another change of variables :  
Since this holds for all , we have:

Question 4

By q3:  
Using Fourier transform:

Substituting we have

Question 5

For we will get:

However, for we have but,   
Therefore, this assumption does not hold.

For .

Where is since the integrand is a distribution probability.

Even-though this is a gaussian, it is not equal to . For example, let , we get:

In real life, the gaussian PSF is more likely, this filter arises naturally in real life scenarios unlike the ideal rectangular filter.

Question 6

The distribution of given the corresponding patch is:  
Moreover,  
The ML for is given by the patches :  
Question 7

Since the convolution with transforms a high-resolution image into a low-resolution image, is some blurring filter (low-pass-filter). One such filter is a gaussian filter (a reasonable filter as we seen in question 5). Therefore, could be a linear operator that normalize to have zero-mean and a desired variance of . The MAP is given according to:

Question 8

Note that the convolution with the down-sampled is a linear operation. We can denote , therefore,

To iteratively solve this problem, we denote .  
We denote the objective function  
We can solve the problem using ADMM with the following steps:  
To solve the first step, we can use gradient-based approach using the gradient we calculated in the beginning of the question:

Question 9

We define , then:  
Using the fact that we get , plugging it in, we get:

Question 10

As we have seen in question 2:   
Using the result from question 1, . Plugging it back:

Question 11

Given the low-resolution image, we can randomly extract patches from it, zoom-in the patches and search for similar patches in the low-resolution image. Suppose we find such a patch whose scaled version is similar to other patches, we can use it to recover the true using the same algorithm as in question 8. According to question 10, using the zoomed-in version patches is equivalent to using the high-resolution patches we used before (as in question 6). All in all, the new algorithm is as follows:

* Initialize
* While
  + Crop a random patch from the low-resolution image.
  + Rescale the patch for a zoomed-in version.
  + Search for similar patches in the low-resolution image.
  + If found many similar ones:
    - Add to the patch set:
* Run algorithm (q8) with the low-resolution image and as the patch set.

Question 12

We can use MAP to recover the high-resolution image:  
Assuming we have:  
Assuming some prior over , :  
This can be seen as a regularized least-squares.