Report of project3 for the course of Digital Image Processing

Xu Kaiyuan
Student ID:5130309113

I. Introduction

I choose the second topic of Image Super-Resolution. In the conventional way, we recover a high resolution image from several low resolution images by aligning with sub-pixel accuracy. This is an intuitive way to realize the topic we want but has a constraint that it needs huge amount of low resolution images for the same scene, which indicates that if we don't have enough images we want to use as the so-called input, then we can't do the reconstruction.

To minimize the influence of the limitation mentioned before, in the paper 'Image Super-Resolution as Sparse Representation of Raw Image Patch', the way of sparse representation to do the reconstruction is mentioned. In this special method, the only necessary thing is a single low resolution image with two patches sampled from training images as dictionaries. This indicates that we should only do preprocess to get the dictionaries for the process and get only one low resolution image, which seems to be much simpler than the conventional way mentioned before.

A. Some notation and definition

First of all, I am going to place some definition which are necessary for the algorithm. In the paper which give the main idea of the Super resolution method, Y denotes the low-resolution image and X denotes the high-resolution image, and the relation between X and Y is given as

$$Y = DHX$$

Here, we consider that the original image has a relative low resolution due to the blur and the downsampling, so the function of H represents the blurring function and D is the downsampling operator. So our aim is to get the image which is the closest to the X. And we resolve the problem via units of small patches of an image.

In order to get the small patches x of the X, the notation of Dictionaries are given in the research of the paper. For instance, D_h represents the dictionary of high resolution patches which is sampled from some training images. Hence, the wanted patches x can be derived as:

$$x \approx D_h\alpha$$

Here, α is a somewhat important parameter which will be mentioned later, and the equation above can be called as sparse representation as well.

B. Some models for the solution

First of all, the model of the sparse representation is introduced. In order to derive the representation, the paper give the basic expression:

$$min||\alpha||_0 \quad s.t.||FD_l\alpha - Fy||_2^2 \leq \varepsilon$$

Due to that the expression is NP-hard, in this paper, it is simplified as:

$$min||\alpha||_1 \quad s.t.||FD_l\alpha - Fy||_2^2 \leq \varepsilon_1 \quad ||PD_h\alpha - w||_2^2 \leq \varepsilon_2$$

Here, F is to provide a perceptually meaningful constraint on the approximation and w contains the overlap values. Finally, after a serious steps of rewriting, the model can be expressed as:

$$min\lambda||\alpha||_1 + \frac{1}{2}||\widetilde{D}\alpha - \tilde{y}||_2^2$$

After that, the model of the reconstruction constraint is given:

$$X^* = argmin||X - X_0|| \quad s.t.DHX = Y$$

To solve this optimal problem, the method of back-projection is mentioned which is somewhat like the iterative method:

$$X_{t+1} = X_t + ((Y - DHX_T) \uparrow s) * p$$

Here, X_t denotes the estimate of the high resolution image after t-th iteration and p is the back-projection filter, $\uparrow s$ means downsample by factor of s.

C. Introduction of the algorithm

In this section, I am going to simply summarize the algorithm mentioned in this paper.

Step 1, the input of the program should be a single low resolution image and the Dictionaries for both high and low resolution by using the training images.

Step 2, for the predefined range of patches y of Y, solve the optimal problem below:

$$min\lambda||\alpha||_1 + \frac{1}{2}||\widetilde{D}\alpha - \tilde{y}||_2^2$$

Then, generate the high resolution patches x by $x = D_h\alpha^*$. After that, just put x into the high resolution image X and do the loop of step 2 until travel all the patches of Y.

Step 3, use back projection to find out the super resolution image X^*:

$$X^* = argmin||X - X_0|| \quad s.t.DHX = Y$$

II. Realization of the gain of dictionaries

In this section, I am going to introduce the way to get dictionaries.

Here, I have to claim that I found the source code from the writer's website and do the imitation for the program.

```
dict_size    = 512;
lambda       = 0.15;
patch_size   = 5;
nSmp         = 100000;
upscale      = 2;
```

figure 1 some pre-definition

Figure 1 shows some definition, which are the size of the dictionary, the sparsity regularization, the patch size, the number of patches to sample and the downsampling factor.

Then, we sample the patches as below:

```
img_dir = dir(fullfile(img_path, type));

Xh = [];
Xl = [];

img_num = length(img_dir);
nper_img = zeros(1, img_num);

for ii = 1:length(img_dir),
    im = imread(fullfile(img_path, img_dir(ii).name));
    nper_img(ii) = prod(size(im));
end

nper_img = floor(nper_img*num_patch/sum(nper_img));

for ii = 1:img_num,
    patch_num = nper_img(ii);
    im = imread(fullfile(img_path, img_dir(ii).name));
    [H, L] = sample_patches(im, patch_size, patch_num, upscale);
    Xh = [Xh, H];
    Xl = [Xl, L];
end
```

figure 2 sample the patches

Finally, we do the training of the images, the steps can be divided as 3 parts: first normalize the high resolution and low resolution images, respectively; second, do the learning part; the last step, do dictionary training by the results of the learning. Here, I am going to show some results of both the low resolution dictionary and the high resolution dictionary.

	84	85	86	87	88	89	90	91
10	0.0092	-0.1058	-0.0544	-0.1359	0.0904	0.0233	-0.0747	-0.2686
11	0.0292	0.1544	-0.1533	0.1698	-0.1852	0.0763	-0.0652	0.1490
12	0.0192	0.1836	-0.1068	0.0769	-0.1081	0.0823	-0.0571	0.2111
13	0.0188	-0.0737	0.0520	-0.0147	-0.1015	0.0470	-0.0732	0.2123
14	0.0172	-0.1610	0.1050	-0.0776	-0.0963	-0.0271	-0.0880	0.2015
15	0.0103	-0.1041	0.0250	-0.1304	0.0360	-0.1323	0.0753	0.1631
16	0.0191	0.1603	-0.1521	0.1715	-0.1663	0.0519	-0.0598	-0.1053
17	0.0187	0.2012	-0.0502	0.0896	0.1210	0.0242	-0.0812	-0.1751
18	0.0189	-0.0746	0.1757	-0.0102	0.2012	-0.0631	-0.0910	-0.2194
19	0.0163	-0.1612	0.2197	-0.0723	-0.0292	-0.1598	0.0818	-0.2149
20	0.0239	-0.1055	0.1469	-0.1297	-0.0487	-0.2459	0.3500	-0.1973

	52	53	54	55	56	57	58	59
10	-0.0947	0.0231	-0.1229	-0.0759	-0.0020	0.0548	0.1033	0.0043
11	-0.0368	-0.0275	0.0167	0.1133	-0.0031	-0.0260	0.0326	0.0229
12	-0.0494	-0.0258	-0.0174	0.1052	-0.0021	0.0221	-0.0325	0.0485
13	-0.0606	-0.0243	-0.0488	0.0539	-8.9806e-04	0.0588	-0.0225	0.0480
14	-0.0537	-0.0226	-0.0671	-0.0160	0.0011	0.0647	0.1140	0.0292
15	-0.0292	-0.0231	-0.0749	-0.0540	0.0011	0.0493	0.0932	0.0084
16	-0.0541	-0.0472	0.1063	0.0901	-0.0025	-0.0071	0.0360	-0.0476
17	-0.0363	-0.0450	0.0761	0.1076	-0.0027	0.0271	0.0090	-0.0315
18	-0.0190	-0.0449	0.0400	0.0982	2.7939e-05	0.0542	0.0059	-0.0125
19	-0.0058	-0.0456	0.0119	0.0623	0.0034	0.0603	0.0217	-0.0011

figure 3 part of the results of the dictionaries

III. Pre-process

Noticed that the given test-images are all high resolution images and, whereas our requirement of the input image should be a low resolution one. So the very first step is to get the low resolution image by make the test-image small by the inner function in the MATLAB:

```
lowIm = imresize(testIm,1/zooming, 'low');
imwrite(lowIm,'Data/Test/low.bmp','BMP');
```

figure 4 sample the patches

Here, the function used in figure 4 is the zoom function which is needed here to zoom the high resolution image to the low resolution one.

Then, what we need to do is input the low resolution image we get before to the main algorithm to get the super resolution image. And the next section is going to mainly introduce the function.

IV. Realization of the sparse representation prior

As the input image is the low resolution image, after set the pre-decision parameters as mentioned before, the first step is to get the gradient of the input, and the parameters of function F used here are set the same as the paper used:

$$f_1 = [-1, 0, 1], \qquad f_2 = f_1^T,$$
$$f_3 = [1, 0, -2, 0, 1], \quad f_4 = f_3^T,$$

```
hf1 = [-1,0,1];
vf1 = [-1,0,1]';
hf2 = [1,0,-2,0,1];
vf2 = [1,0,-2,0,1]';
```

figure 5 selection of the function F

After that, just do the convolution and some simple calculation to get the gradient.

Then, there is the important part coming in, which is, to recover the image patch by patch.

In order to achieve this goal, the author applied the function called "SolveLasso" which is used to calculate and solve the optimal problem discussed in section I and give the most closest solution α which is needed for the later calculation of patch x. And then we do the process to handle the overlap parts and the missing parts to complete the images.

V. Back-projection method

After solving the optimal problem, the latest step is to apply the back-projection method to get the final results of super resolution images.

The main idea of the Back-projection method is that, first build a filter with gaussian type and then do the convolution between a properly resized image and the filter operator for several times to enhance the resolution of the processed images in the past section further.

```
function [im_h] = backprojection(im_h, im_l, maxIter)
[row_l, col_l] = size(im_l);
[row_h, col_h] = size(im_h);
p = fspecial('gaussian', 5, 1);
p = p.^2;
p = p./sum(p(:));
im_l = double(im_l);
im_h = double(im_h);
for ii = 1:maxIter,
    im_l_s = imresize(im_h, [row_l, col_l], 'bicubic');
    im_diff = im_l - im_l_s;
    im_diff = imresize(im_diff, [row_h, col_h], 'bicubic');
    im_h = im_h + conv2(im_diff, p, 'same');
end
```

figure 6 the implementation of Back-projection method

VI. Experiment results

Here, I am going to place the results by applying the program to the test images given online.

First of all, the original high resolution images given online are placed below:


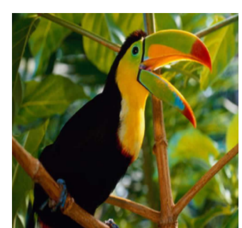
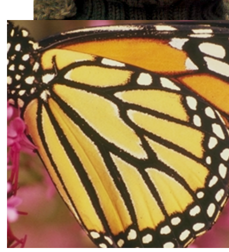
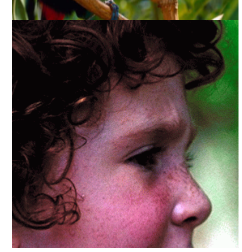

figure 7 the given test images

In this experiment, the zooming factor is set to be 3 which indicates that the low resolution images are obtained by shrinking the original images by three times; and between each patch, we assume the overlap is 1 pixel, which imply that in high resolution images, due to the zooming factor, the overlapped pixels are 3 between two adjacent patches. Then, let's do comparison between the high and low resolution images:

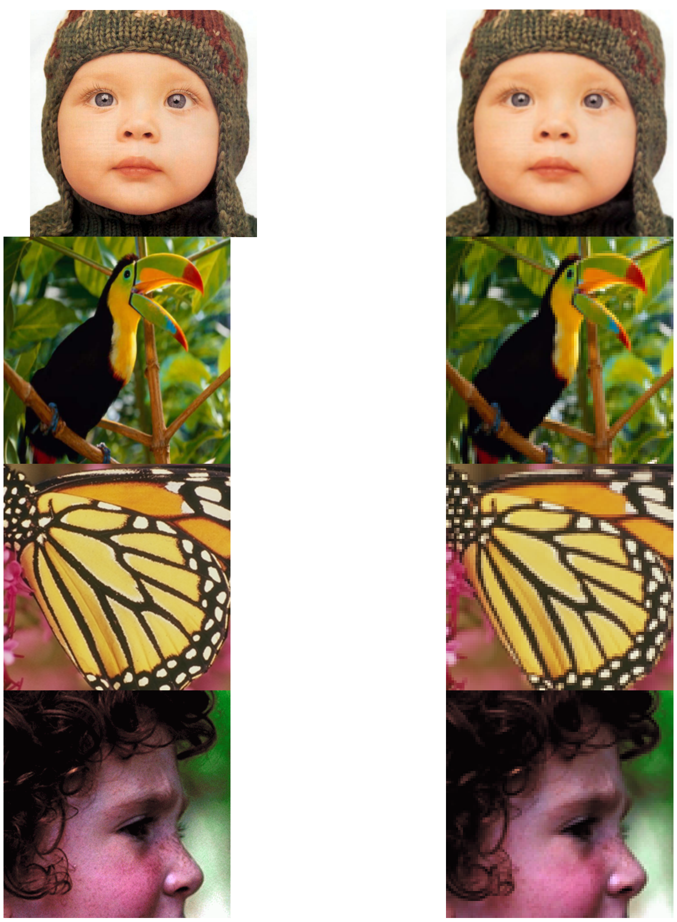

figure 8 comparison of high and low resolution images (left:high right:low)

From the figure 8, we can easily find out that the images do become low resolution and have met the need of our program.

Then, I will show the contract between the low resolution images and the super resolution images get by our program:

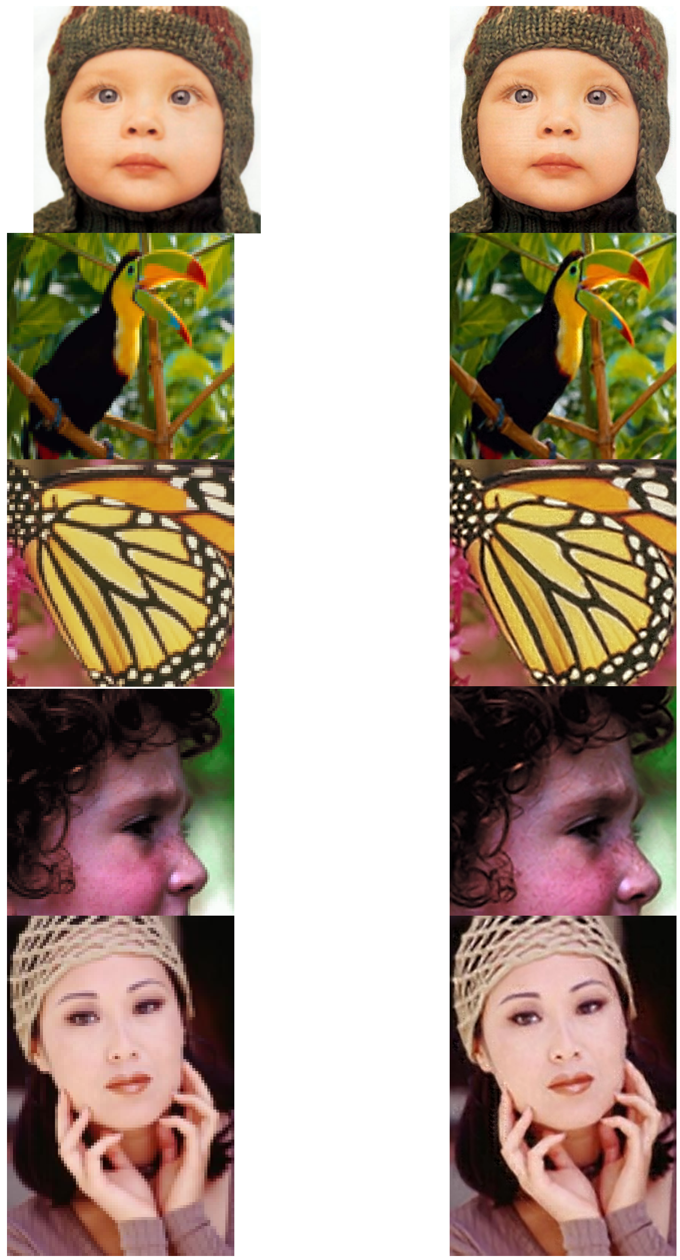

figure 9 comparison of low and super resolution images

(left:low right:super)

From figure 9, we can easily point out that the super resolution program does work, which transfer the low resolution image to relative high one, but when it comes to the original images, the quality still has some flaws which shows that the reconstruction is not perfect enough.

VII. Analysis

From the above results, we can find out that, the results after the process do have relative high resolution than the low resolution ones, which can be inferred from the relatively more details in the result images. In addition, in some parts of the super resolution images, we may find them lighter than the low resolution images for we do the process mainly in the illuminated domain. So we can conclude that the method in this paper does work.

Finally, thank you very much for reading my report!