

# HKUST Independent Project CSIT 6910

**Report:**

**Recovery of Blurred Portrait Photograph**

**Student:** Zhou Zilin

**Supervisor:** Prof. Rossiter

# Contents

1 Introduction.....	3
1.1 Overview.....	3
1.2 Project Objective .....	3
2 Design .....	4
2.1 Architecture .....	4
2.2 Function Design .....	4
2.2.1 Choose a photo.....	4
2.2.2 Apply effect .....	4
2.2.3 Check details.....	5
2.2.4 Save the result photo .....	5
2.3 UI Design.....	5
3 Algorithms Implementation .....	6
3.1 Traditional sharpen .....	6
3.2 Sharpen similar with Gauss Blur.....	7
3.3 Origin photo plus masking .....	9
3.4 Unsharp Masking .....	9
4 Project Demonstration .....	10
5 Conclusion.....	12
6 Appendix.....	13

# 1 Introduction

## 1.1 Overview

Nowadays image is undisputedly the predominant medium for conveying information. The popularization of electronic devices and social media have greatly stimulated people to take photos anywhere and anytime with great ease. And among all types of photograph, no doubt that portrait is the most popular one. However, due to lacking expertise in photography, limit of devices and other inevitable causes, it is common that we took photos with blurred faces.

Obviously that there is a significant need in recovering those blurred portrait photos. In addition to normal usage, there are also many applications. For example, the police needs to identify criminals' faces from camera records, but usually those records provide very low resolution hence it is difficult to see things clearly. Though there exists some software which can be used for recovering blurred images, like Photoshop, it requires the user to be very familiar with the software and the recovered result varies from user to user. What's worst, those software lacks automatic functionality to process photos that heavy labor working is needed to handle each photo.

## 1.2 Project Objective

In this project I focus on recovering blurred portrait photographs. The cause of the blurred image varies, thus, it's necessary to implement different

algorithms to recover the blurred image due to its specific cause. This project aims to create a user-friendly and easy-operating system to help user recover blurred images.

## **2 Design**

### **2.1 Architecture**

This is a Microsoft Foundation Classes (MFC) program.

### **2.2 Function Design**

This program is mainly used to recover blurred image. There are four main functions in this program.

#### **2.2.1 Choose a photo**

You could choose a blurred image in different format like jpg, png, etc.

#### **2.2.2 Apply effect**

You could apply multiple effects on the image. Different effect has its own variables. There are five effects provided by this program which are Traditional Sharpen, Sharpen similar with Gauss Blur, origin plus masking and Unsharp Masking.

## 2.2.3 Check details

You could check the details of the result photo, using zoom in or zoom out.

## 2.2.4 Save the result photo

You could save the result image to specific path.

## 2.3 UI Design

The UI design is based on the function design above and shown in Fig. 1.

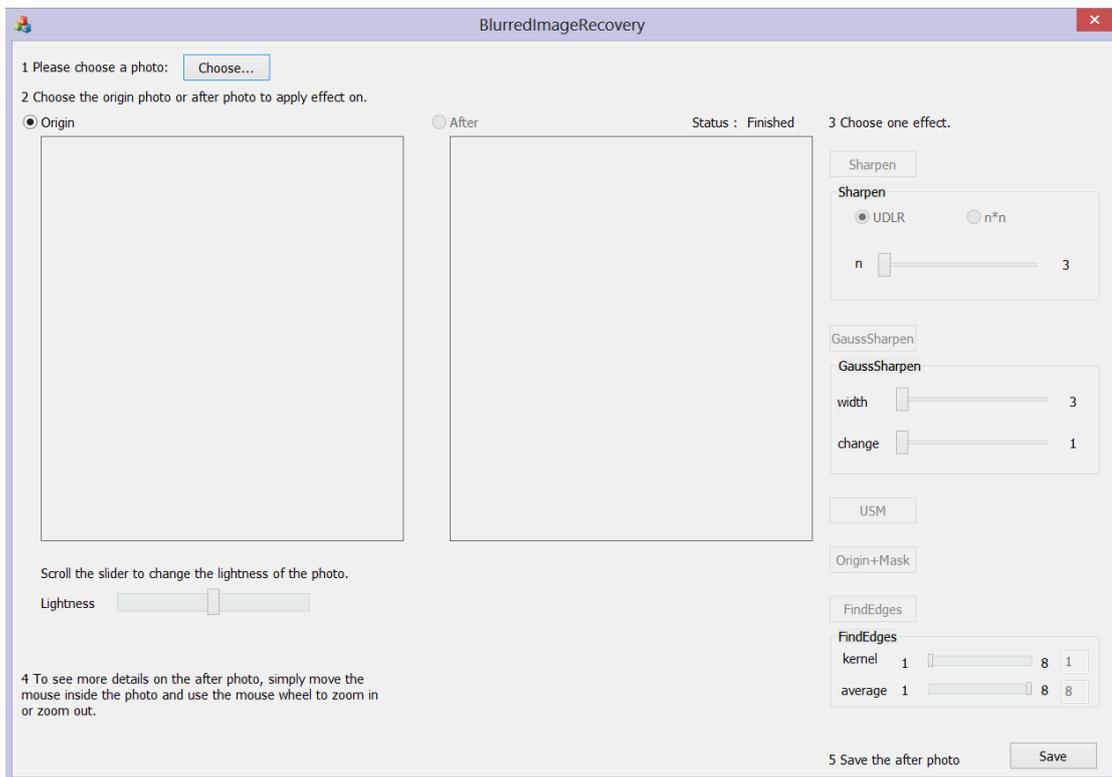


Fig. 1 UI Design

# 3 Algorithms Implementation

## 3.1 Traditional sharpen

In the traditional sharpen algorithm, I implement the traditional sharpen method that for each pixels, I recalculate its value by combine its own value with the pixels around it. The weights of the pixels around the origin pixel are the same.

There are two types of traditional sharpen modes, one is Up-Down-Left-Right (UDLR), that the pixel we want to recalculate is only affected by its up pixel, down pixel, left pixel, right pixel.

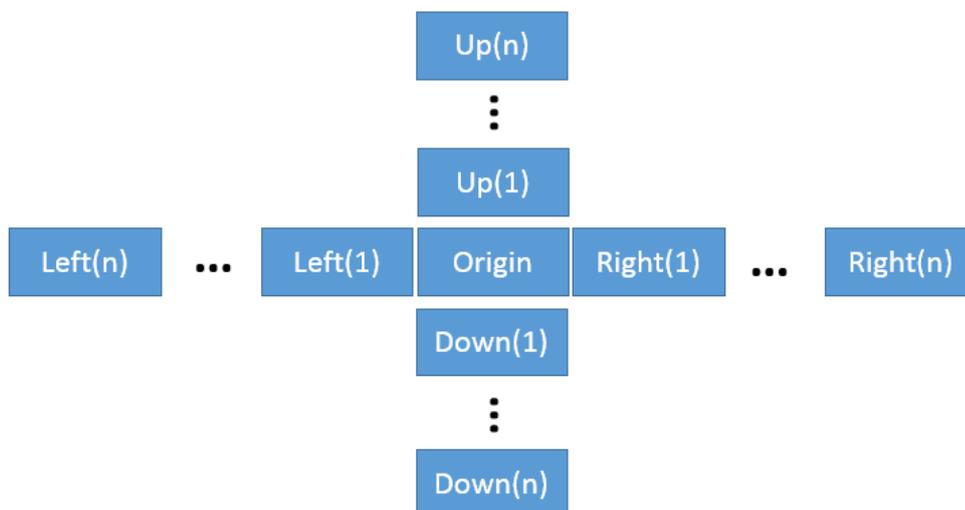


Fig. 2 UDLR

Figure 2 is a demonstration of UDLR method. There is one variable in this method, width. From Fig.2, width =  $n * 2 + 1$ .

The new value of the new origin pixel =  $-1 * \text{left}(1) + \dots + -1 * \text{left}(n) + -1 *$

$\text{right}(1) + \dots + -1 * \text{right}(n) + -1 * \text{up}(1) + \dots + -1 * \text{up}(n) + -1 * \text{down}(1) + \dots + -1 * \text{down}(n) + [(\text{width} - 1) * 2 + 1] * \text{origin}.$

The second mode is  $n*n$  mode. This mode is similar with UDLR mode, the difference is the origin pixel is affected by the pixels all around it. There is also one variable that can be changed, width. That width =  $n*2 + 1$ , similar with the UDLR mode. Figure 3 is an example of the  $n*n$  mode.

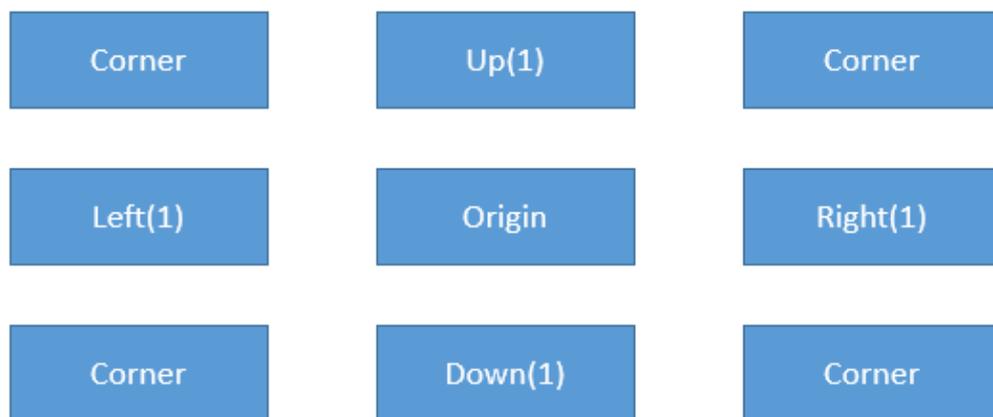


Fig. 3  $n*n$

### 3.2 Sharpen similar with Gauss Blur

This method is similar with the  $n*n$  mode of traditional sharpen. The difference is that in the traditional sharpen, all the pixels affect the origin pixel of the same weight. This method give different weights to different pixels according to its distance with the origin pixel. The further the distance is, the less weight it can get.

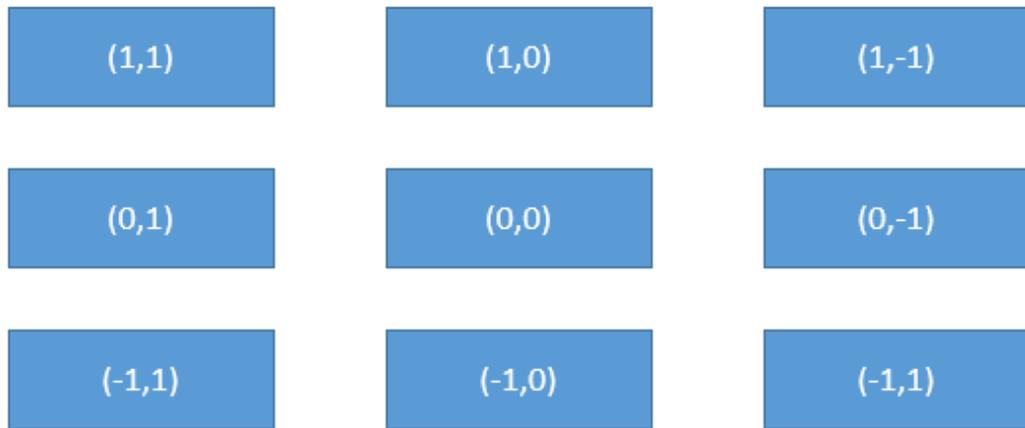


Fig. 4 Sharpen similar with Gauss Blur

From Fig.4 we can see that the origin pixel is in the position(0,0) and there are 8 pixels around it that combined together can result in the new value of the origin pixel. The distance of position(1,0) to (0,0) is  $(1-0)^2 + (0-0)^2$ , and the weight equals to  $1/\text{distance}$  as the distance grows, it should affect less on the origin pixel. In figure 5 we can see the weight of the pixel around the origin pixel according to different distances and changes.

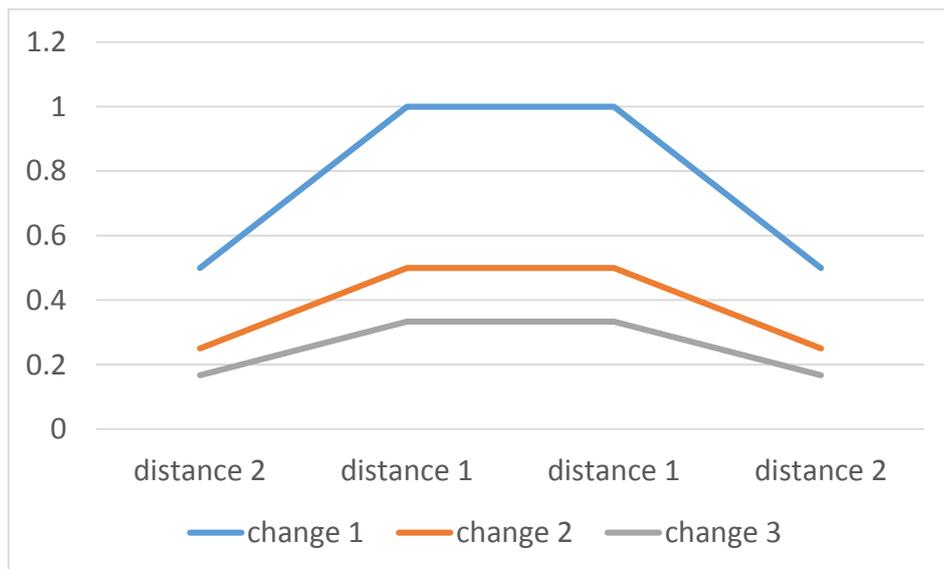


Fig. 5 Weight with different change value

### 3.3 Origin photo plus masking

In this method, first I find the edges of the origin image, then add the value of the edges on the origin image, to enhance the edges and therefore get a clearer result.

### 3.4 Unsharp Masking

The procedure of this method is shown in Figure 6.

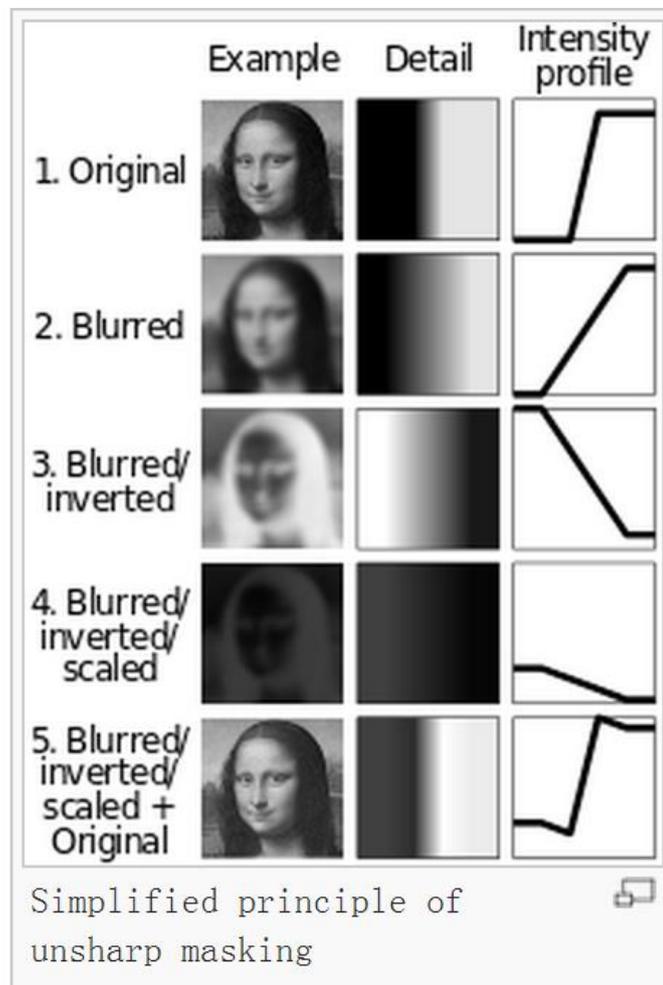


Fig. 6 Unsharp masking (from Wikipedia)

## 4 Project Demonstration

In this part, I would just simply apply different results of the same photo.

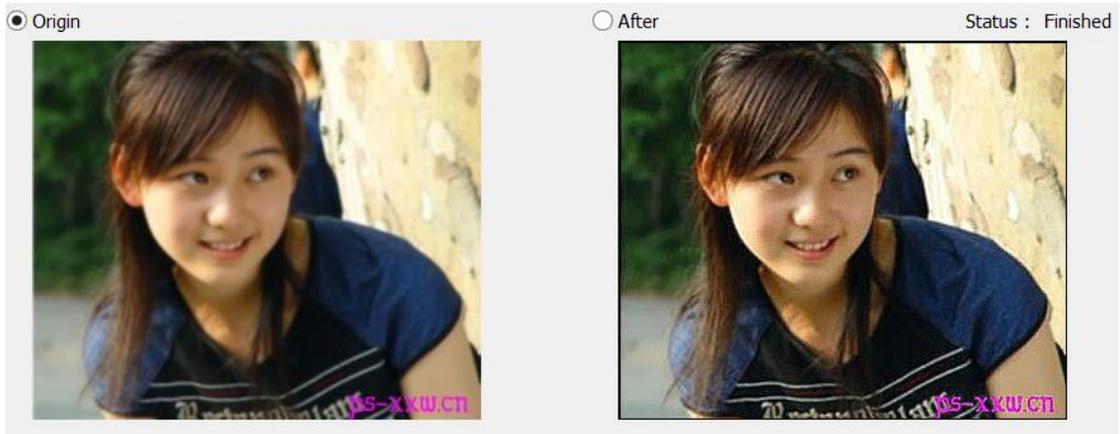


Fig. 7 Sharpen – UDLR with  $n = 3$

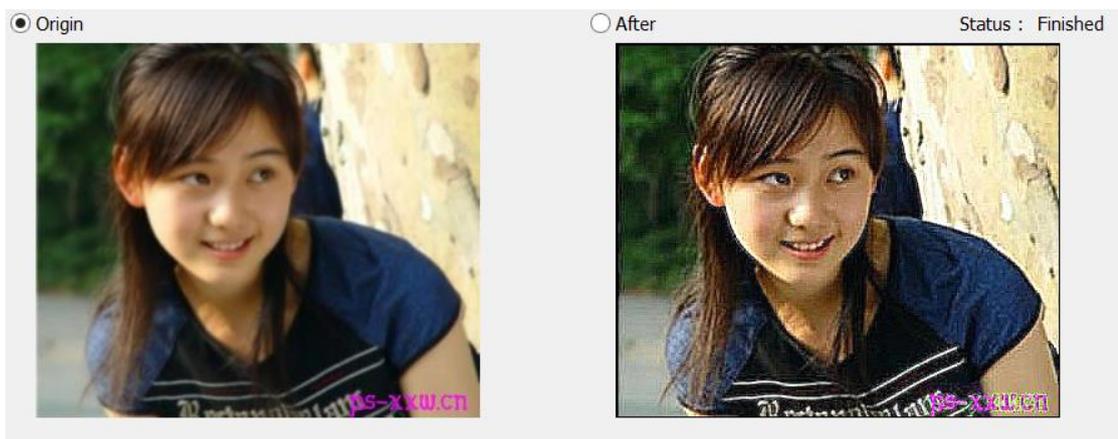


Fig. 8 Sharpen –  $n * n$  with  $n = 3$

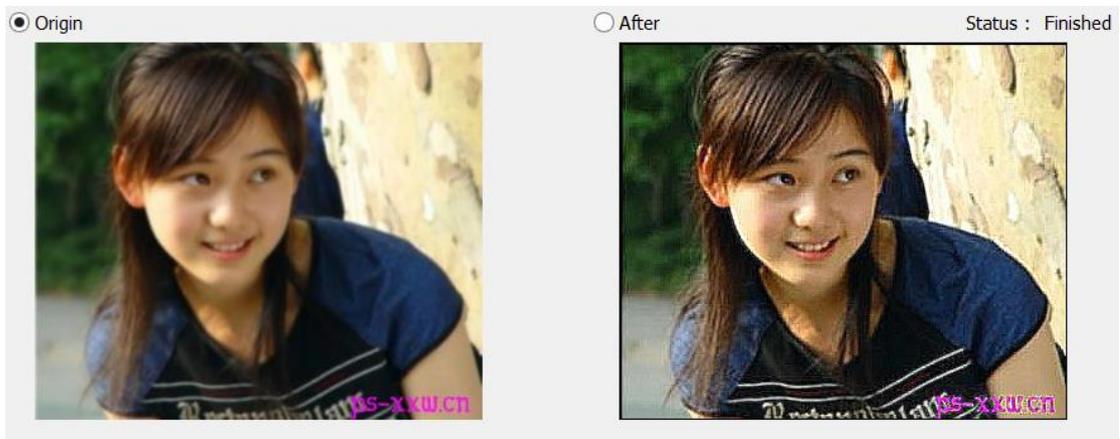


Fig. 9 Gauss Sharpen with width = 3 change = 1

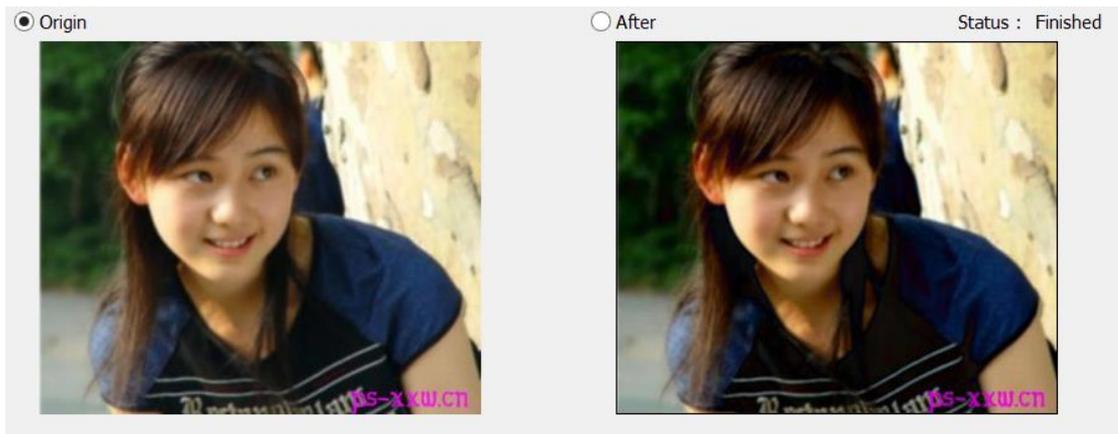


Fig. 10 Unsharp Masking

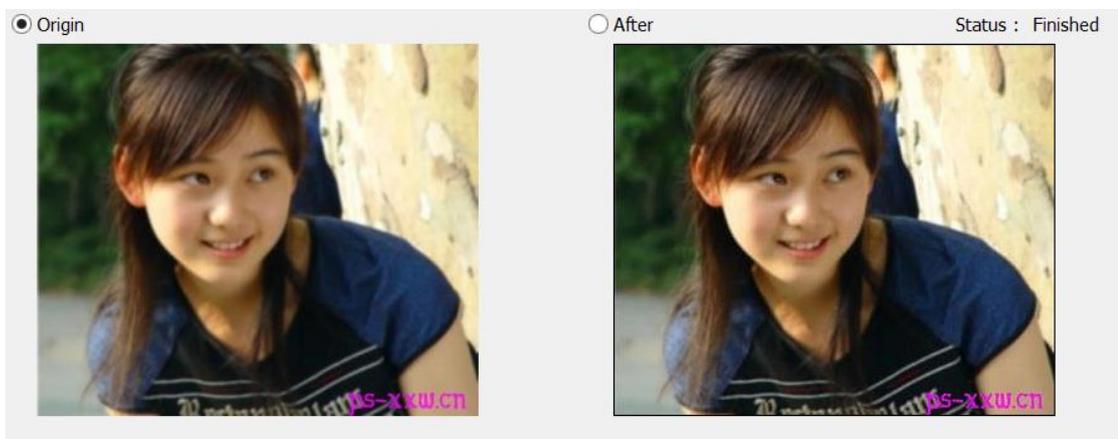


Fig. 11 Origin + Masking

Figure 12 demonstrates the detail of the result image.

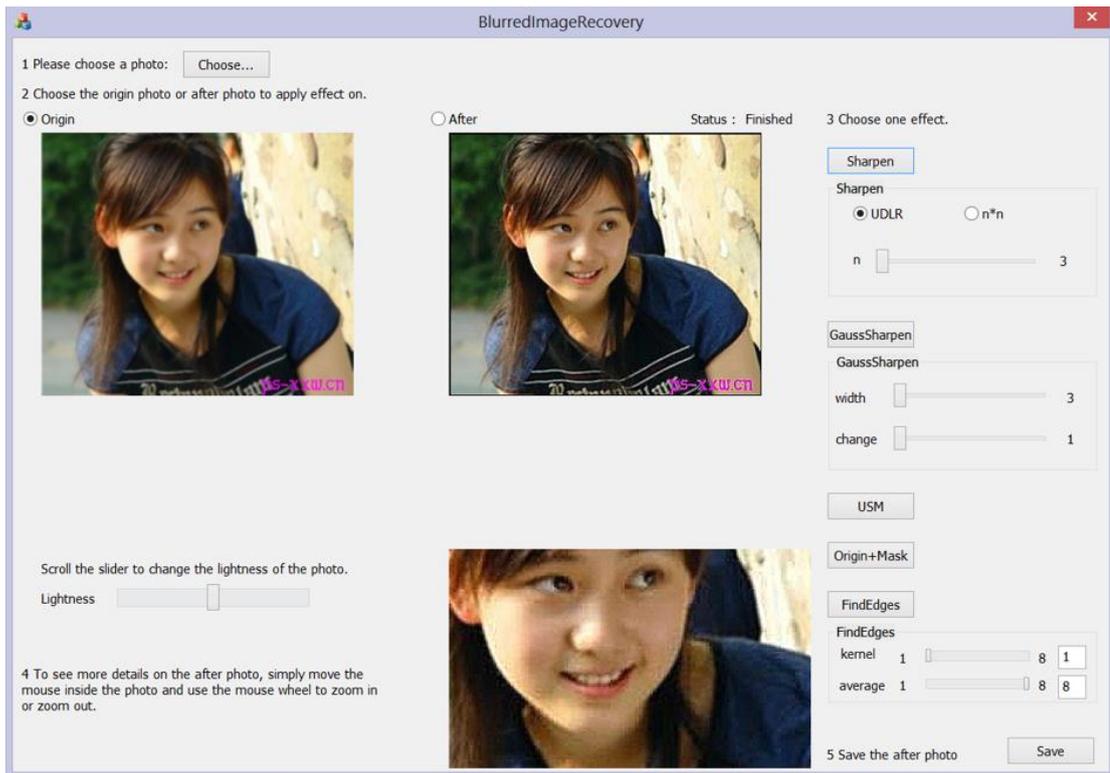


Fig. 12 Detail of the result image

## 5 Conclusion

In this project, I implement several methods to recover the blurred image. In the project demonstration part we can see the results of different methods. We can see that there are some methods that are more suitable for such a blurred image, however, due to the different causes of the blurred image, the results of the methods vary. Thus, in this project, the user could choose different methods he likes or apply multiple methods on the result image to see whether he could get a better result.

There are still many improvements that can be made for this project, like that it would be better that the program can automatically recognize the cause

of the blurred image and recommend a method that suits the image best. Though I cannot accomplish such a function now, I will continue work on that. To sum up, in my opinion, this project can recover blurred image to some extends.

## 6 Appendix

### **Minutes of the 1st meeting**

**Date:** February 21<sup>st</sup> **Time:** 9:50 am

**Place:** Room 3512

**Attending:** Zhou Zilin, Prof. Rossiter

**Absent:** none

**Recorder:** Zhou Zilin

#### **Approval of minutes:**

This is first formal group meeting, so there were no minutes to approve.

#### **Report on Progress:**

Zhou Zilin show the ideas of the algorithm and the way to implement the project.

#### **Discussion Items and Things to Do:**

Prototype of the project

The first blur recovery algorithm to be implemented

**Meeting adjournment:** 10:20 am

## **Minutes of the 2nd meeting**

**Date:** March 24<sup>th</sup> **Time:** 11:50 am

**Place:** Room 3512

**Attending:** Zhou Zilin, Prof. Rossiter

**Absent:** none

**Recorder:** Zhou Zilin

**Approval of minutes:**

The minutes of the last meeting were approved without amendment.

**Report on Progress:**

Zhou Zilin has built up the program and implement a sharpen algorithm that can recovery the image through some extend.

**Discussion Items and Things to Do:**

More blur recovery algorithms

User Interface to be enhance

**Meeting adjournment:** 12:20 pm

## **Minutes of the 3rd meeting**

**Date:** April 24<sup>th</sup> **Time:** 2:00 pm

**Place:** Room 3512

**Attending:** Zhou Zilin, Prof. Rossiter

**Absent:** none

**Recorder:** Zhou Zilin

**Approval of minutes:**

The minutes of the last meeting were approved without amendment.

**Report on Progress:**

More algorithms are implemented, the program is almost finished.

**Discussion Items and Things to Do:**

Finish the project by adjusting some details

**Meeting adjournment:** 2:30 pm

## **Minutes of the 4th meeting**

**Date:** May 14<sup>th</sup> **Time:** 4:00 pm

**Place:** Room 3512

**Attending:** Zhou Zilin, Prof. Rossiter

**Absent:** none

**Recorder:** Zhou Zilin

**Approval of minutes:**

The minutes of the last meeting were approved without amendment.

**Report on Progress:**

The program and the draft report are finished.

**Discussion Items and Things to Do:**

Final report and video.

**Meeting adjournment:** 4:30 pm