CSC3002

Introduction to Computer Science: Programming Paradigms

Project Proposal
A Multifunctional Image Editor: Unipicture

Ran Hu 116010078

Shaoyu Wang 117010262

Jiayi Qiu 116010182

Zhixian Hu 116010082

Xuening Zhang 116010302

Jinjing Yang 115020157

Course Coordinator: Prof. Rui Huang

The Chinese University of Hong Kong, Shenzhen

Spring 2019

Background

Many people enjoy taking photos, edit them, and share to others. They like taking beautiful selfies to save sweet memories; they want to share funny scenes with friends through WeChat; they create compelling and interesting photos for pleasure. In many cases, people need to do some adjustments to the original photo. If the weather is not good, they may increase the brightness or add filters to the photo; if there are many people and stuff in the background, they may need to blur them; if they want to make a meme, they may add some funny words on the picture. Such image editing tools with multiple functions are widely used in our daily life.

Motivation

For mobile phones, there are already many kinds of image editor. Some Apps like Fashion Self-image and B612 mainly support users to take photos using these tools with predetermined filters and effects. Others like MeiTu and Play Photos give users choices to mainly make adjustment to already-taken photos on size, filters, brightness, contrast, saturation and etc. However, for PC users, choices are limited. The most common image editor may be Photoshop. However, many users are not comfortable with this tool because it is for professional use, and it may be a little difficult for beginners to deal with a photo quickly. And if we want to make simple adjustments to a photo received on PC, we need to transfer the image to our mobile phones and after editing, transfer it back to PC, which is not efficient. Considering this situation, we want to develop a simple image editor to allow users to make multiple adjustments to images on PC.

Solution

To solve the problem identified above, we propose a multifunctional image editor: Unipicture. As for image processing functions, besides the common functions, some advance algorithms will also be used in our project. Not only will it contain the useful image processing functions, it also has an elegant interface which is friendly for users. Details are listed below.

1. GUI design

The initial user interface is a welcome interface (Figure 1), which includes a button "START" that allows users to enter the main interface. On the main interface (Figure 2), there is an image processing canvas in the middle; some function buttons (tool, filer, sticker, text, bokeh, mosaic, frame etc.) on the left; a strip of display frame on the top; a picture gallery that allows users to load pictures on the lower left corner; icons for delete, save, return, and temporary storage functions on the lower right corner. The display frame on the top will display details of each function when the corresponding function button is pushed. To make the interface more special, a different dictum will be shown every time the user enter the main interface.

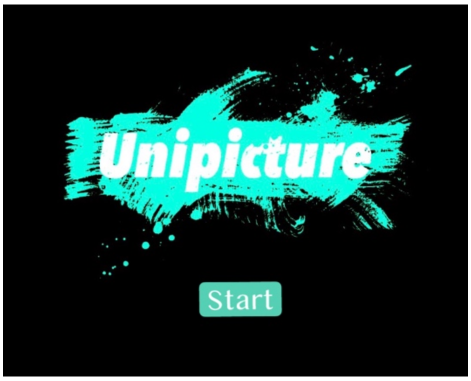
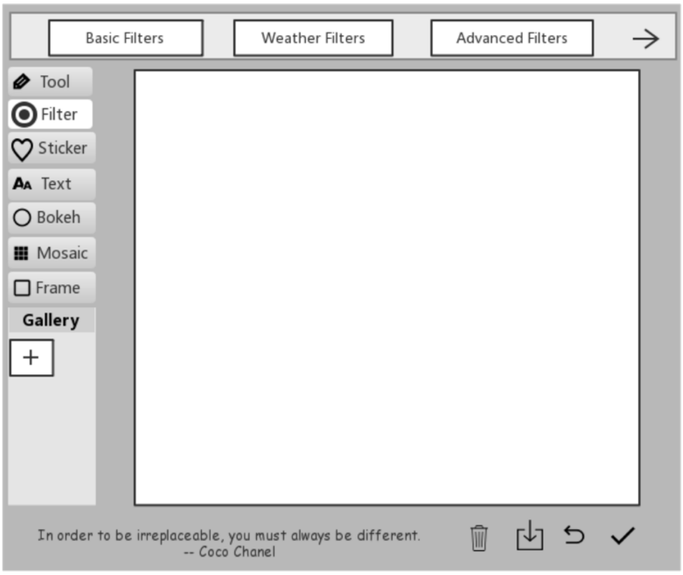


Figure 1. Welcome interface



In order to be irreplaceable, you must always be different.
-- Coco Chanel

Figure 2. Main interface

2. Basic operations

- **Crop:** user could choose to cut picture into a specific length-width ratio (i.e., 1:1, 4:3, 16:9) given by the program or cut picture freely with square shape.
- **Rotate:** user could choose to rotate 90, 180, 270 degrees clockwise.
- **Flip:** user could choose to flip up and down or left and right.
- **Brightness and contrast:** the user will need to input an integer between 0 and 100 to change the degrees. The brightness and contrast of original picture both are set as 50 as default.
- **Black and white mode:** user could only choose to change colorful picture into black-and-white; while the reverse operation is not permitted (Figure 4).
- **Filtering a specific color:** user could choose one of the following colors: red, green and blue and leave it on the picture, and other colors will change into black-and-white.

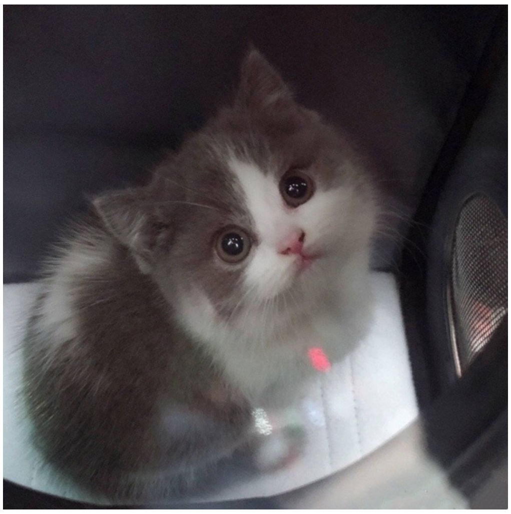


Figure 3. Original photo

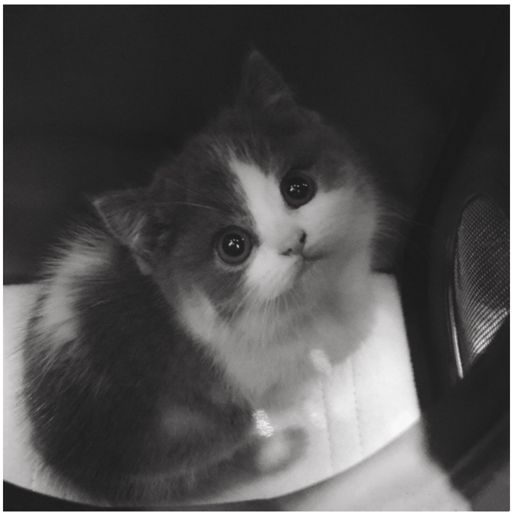


Figure 4. Black and white mode

3. Advanced operations

- **Sticker:** we will collect some images as stickers to build a sticker library. User can choose any stickers and put them on the objective photo (Figure 5).
- **Text:** add texts to the photo.

- **Frame:** add a frame or a border to the photo (Figure 6).
- **Collage:** collage of several photos.
- **Bokeh:** input an image and let the user to blur some places in the photo
- **Background changing:** segment an identification photo and. First segment the face apart with background. Second fill the background with another color.

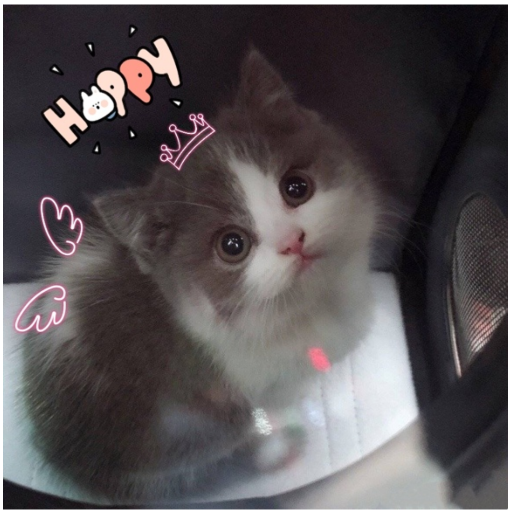


Figure 5. Sticker

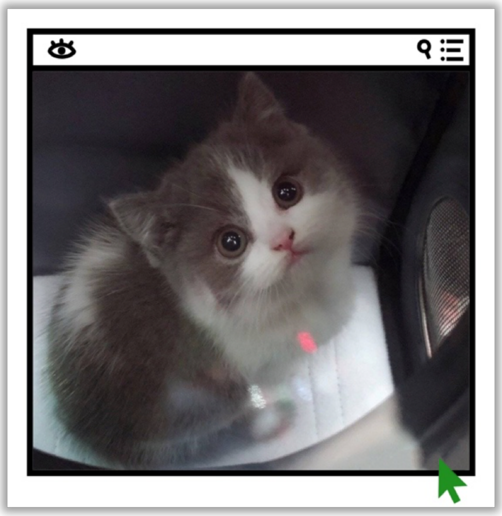


Figure 6. Frame

4. Filters

- **Basic filters** (basic filters included in most Apps, most of them can be fulfilled by adjusting colors or adding masks): Coffee, Pink, Beauty, Silence, Old, Magic.
- **Weather filters** (create static weather effects): Snowy (Figure 7), Foggy, Sunny.
- **Advanced filters** (those advanced filters may include basic algorithms): Embossed, Blur & Sharpen, Sketching.

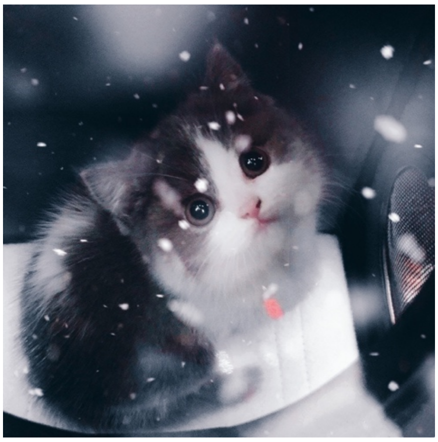


Figure 7. Filter Snowy

Implementation and Resources

There are six main parts of the implementation of this image editor.

1. GUI design

As described in the Solution part, there is a main interface for users to develop their pictures. We will use Qt Designer as a tool to implement the GUI design. Meanwhile, we will read *C++ GUI Programming with Qt 4 (2nd-ed)*, which will teach us how to program GUI applications using Qt. Furthermore, we may consider using some other tools and packages such as XML, OpenGL and QImage to beautify our GUI and create a more user-friendly interface.

2. Image input & output, undo & redo operations

Due to the variety of image formats and the limitation on our time and energy, we decide to use OpenCV, an external package to deal with input and output of image. As for undo and redo of operations, with the consideration of space-saving, we plan to store at most three operations in stack structure.

3. Basic image processing functions

This image processing library is mainly an extension of OpenCV package. Geometric transformation is realized by matrixing image. The algorithms of color adjustment will adapt main stream algorithms in current popular image editors (e.g., Photo-shop). The related materials are mostly from Google and GitHub.

4. Filters design

For basic filters, we mainly adjust colors or add masks to fulfill the function. For advanced filters, we will search on Google and GitHub to find corresponding algorithms to achieve them.

5. The photo sticker function

Some images are collected to become stickers and a sticker library will be built. For the basic functions, we will read the tutorial about OpenCV and the corresponding blogs.

6. The magic wand function

The algorithm of magic wand is mainly based on related papers and blogs. We will also read *OpenCV 3 Computer Vision Application Programming Cookbook* to realize this function.

Schedule

week	date	activity	plan
7	3/13-3/15	project proposal due	finish the proposal
8	3/20-3/22	proposal presentation	give the presentation
9	3/27-3/29	assignment 2	write codes to implement all the functions
10	4/3-4/5	assignment 2 due	
11	4/10-4/12		
12	4/17-4/19		
13	4/24-4/26		integrate codes
14	5/1-5/3	assignment 3	
15	5/8-5/10	assignment 3 due	write the final report
16	5/14-5/18	project and report due	test and adjustment

Division of work

Name	**Division**
Ran Hu	functions design, loading and saving photo
Shaoyu Wang	photo blur, photo background identification
Jiayi Qiu	GUI design
Zhixian Hu	basic operations on photo
Xuening Zhang	stickers, text and stitching
Jinjing Yang	filters