浙江大学

Object-Oriented Programming

期末 project 报告

学生姓名 ： 张童童　　学号： 3160101315

学生姓名 ： 江如蓝　　学号： 3160103777

学生姓名 ： 王钟毓　　学号： 3170105709

学生姓名 ： 陈宇威　　学号： 3170105706

学生姓名 ： 彭子帆　　学号： 3170105860

组长：彭子帆　　电话：17342017609

2018~2019 春夏学期　2019 年 6 月

期末 project 报告评分标准

鼓励原创，提倡自己思考!!

1） 图文并茂。文字通顺，语言流畅，无错别字。字数不少于5000。20 分

2） 内容不完整，马虎潦草，或内容有明显错误；<12 分

3） 没有排版；-2 分

4） 没有体现团队协作：-3 分

5） 存在拼凑、剽窃等现象一律认定为抄袭；0 分

1、需求分析

在科技日益发展的今天，人们对于图像的处理要求越来越高；从人们自拍时使用的美颜滤镜，再到计算机图形学领域的蓬勃发展，图像处理一直时计算机领域的热点。在此背景下上，我们小组结合实验选题的推荐，选择了图像处理软件作为本次 OOP 大作业的内容。

图像处理软件的面向对象主要是软件面向的用户。图像处理软件不仅应满足用户日常对图片处理的需求（如裁剪、旋转等），也应该具有良好交互性，界面对使用对象友好。

在此基础上，根据实验选题的要求，基本的图像处理器需要具有基本的编辑处理图像功能，实现图像的大小调整、平移旋转、缩放分割；当然我们小组在这一基础上，我们希望实现一些拓展功能，如对比度、灰度的调整，并且实现图形界面，使得我们的软件使用更加友好。

1.1 基本需求

（1） 具有图像大小调整、平移、缩放、旋转、分割功能；

对于图像的基本简单操作，也是图像处理工具软件最基础的基本功能。括大小的缩放和分割等编辑功能和平移、旋转等对于位置的控制。

（2） 能够显示直方图；

图像直方图由于其计算代价较小，且具有图像平移、旋转、缩放不变等众多优点，广泛地应用于图像处理的各个领域，特别是灰度图像的阈值分割、基于颜色的图像检索以及图像分类。

（3） 具有图像（圆、三角形、长方形）编辑功能；

（4） 可将图像文件保存、打开；

图像处理工具软件需要导入要编辑的图像并输出处理完毕的图像，这就需要软件具有文件的保存和打开的功能

1.2 拓展需求

（1） 图片对比度、亮度的调整

很多时候，一张图像被过度曝光显得很白，或者光线不足显得很暗，有时候背景跟图像人物也观察不清楚，这个时候可以通过调节图像的两个基本属性-亮度与对比度来获得整体效果的提升，从而得到质量更高的图片。

对比度、亮度调整示例：

（2） 图片饱和度的调整

饱和度是指色彩的鲜艳程度，也称色彩的纯度。饱和度取决于该色中含成分和消色成分（灰色）的比例。含色成分越大，饱和度越大；消色成分越大，饱和度越小。纯的颜色都是高度饱和的，如鲜红，鲜绿。混杂上白色，灰色或其他色调的颜色，是不饱和的颜色，如绛紫，粉红，黄褐等。完全不饱和的颜色根本没有色调，如黑白之间的各种灰色。

饱和度调整后的对比：

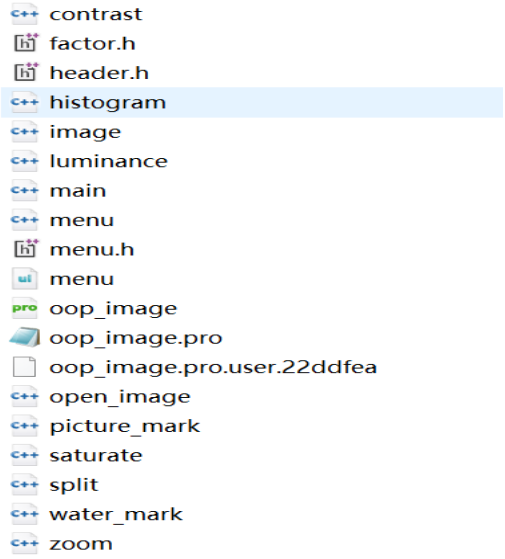
（3） 为图片增加水印

为图片增加水印，要求水印半透明不能将背后图片完全盖住。初步要求是输入图片后可在已有水印模板中选择水印添加，水印位置可自选，高级要求提供输入界面，用户输入文字产生相应水印并添加，提供图片导入界面，用户选择要导入的图片作为水印添加

2、关键设计思路或方法

本次程序设计主要采用 C++中分模块设计的的基本原则，即将所要实现的功能进行分割，C++中每一个类实现一部分模块的功能最后进行总体的实现。

根据需求的分析，我们明确了程序的功能要求，并且根据其功能对程序进行了“分块”

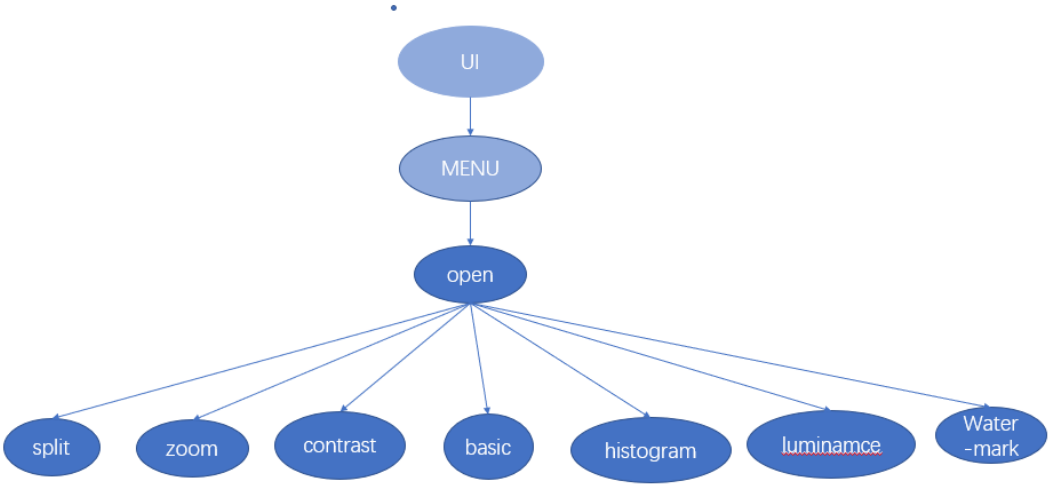


factor.h 以及 header.h 里面定义了各个变量、函数的声明以及各种常量

（eg.最大放大倍数（max_zoom）、初始化亮度（initial_bright）），供剩下的模块调用。contrast、luminance、saturate 模块分别实现了图片对比度、亮度、饱和度的调整；water_mark 模块实现了给图片添加水印的功能；zoom 模块实现了图片的缩放；open_image 模块实现了图片的打开。其中比较重要的就是 menu 模块，

menu 模块包含了本软件所有的基本操作，实现了图形界面以及对于各个操作的响应以及相应过后不同模块的调用。其中一些函数的实现调用了 Qt 或 openCV 库中的函数。

程序的主要结构如图所示：（basic 为基本图像编辑功能）



3、详细设计

其详细设计我们分模块进行介绍：

Factor.h

Factor.h 主要定义了一些界值以及一些变量初始化的初值,均为其他模块调用的常量，在此不一一赘述。

Header.h

Header.h 模块中包含了本程序所需使用函数大的库，大部分为 Qt 自带的库，以及 C++编程所需要大的库，在此也不一一赘述。

Histogram.cpp

Histogram 模块主要实现的是图像直方图的显示功能。灰度直方图表示图象中具有每种灰度级的象素的个数，可以用来评价图像的一些性质。若要绘制图像的灰度直方图，首先就要从定义入手，需要获得所输入的二维图像的每个灰度级的像素个数。因此我们遍历图像，统计具有相同灰度级的像素个数。所有的像素点由于在未统计前像素个数无法确定，因此我们用 vector 的形式来保存。

```
void Menu::get_histogram_function()
{
    histgram = new QVector<int>(300);
    unsigned char* graydata = image->bits();
    for (int i=0;i!=image->width();i++)
    {
        for (int j=0;j!=image->height();j++)
        {
            int index = int(*graydata);
            (*histgram)[index] = (*histgram)[index]+1;
            graydata+=3;
        }
    }
    graydata = NULL;
}
```

接着我们用 Qt 自带的 QPainter 函数绘制直方图。首先要获得像素个数最多的那一个灰度级，将这个数量与窗口的高度进行一个映射，其他灰度级在绘制的时候都要按照这个比例来绘制，这样得到的结果就是在数量最多的灰度级是最高的。值得注意的是每个方向都预留点空间绘制坐标轴

由于 QPainter 的坐标系是从左上角开始的，这里对坐标系进行了一个简单的转换，使原点为左下角，x 轴向右，y 轴向上。这里将 QPainter 绘制的结果以 QImage 的形式保存，是方便后续将直方图输出保存到硬盘中。

```
void Menu::paint_histogram_function()
{
    histgram_picture = new QImage(800,600,QImage::Format_RGB888);
    QPainter p(histgram_picture);
    p.setBrush(QBrush(QColor(121,121,121)));
    p.drawRect(0,0,this->width(),this->height());

    p.setBrush(QBrush(QColor(255,255,255)));
    p.drawRect(0,0,this->width(),this->height());
    QVector<int> sortcount(300);
    sortcount = *histgram;
    std::sort(sortcount.begin(),sortcount.end());
    int maxcount = sortcount[sortcount.size()-1];


    QImage* hist = new QImage(this->width(),this->height(),QImage::Format_RGB888);
    hist->fill(qRgb(255,255,255));
    p.translate(0,hist->height());


    p.drawLine(0,0,100,100);

    int wid=hist->width();
    int hei=hist->height();

    p.drawLine(10,-10,wid-10,-40);// 横轴
    p.drawLine(10,-10,10,-hei+10);//纵轴

    float xstep = float(wid-20) / 256;
    float ystep = float(hei-20) / maxcount;
```

Luminance.cpp

亮度调节的原理十分简单，图片颜色由 red、green、blue 三种颜色通道的变化以及他们之间的相互叠加所形成的。亮度的调节也无非是对于这三种颜色通道数值的调节。在该模块，我们通过获取图片像素点 RGB 分量值，直接加上亮度值来进行亮度的调节。(pixel 代表图片上的像素点，对像素点进行遍历添加亮度值)

```
void Menu::luminance_funtion()
{
    // red, green,blue refer to Mitaka colored passage
    int red,green,blue;

    // calculate the whole number of pixels
    int pixels = image->width() * image->height();

    unsigned int *image_data = (unsigned int *)image->bits();

    for (int i = 0; i < pixels; ++i)
    {
        // the value of rgb should within 0-255
        // change the unproper number
        red= qRed(image_data[i])+ bright_value;
        red = (red < 0x00) ? 0x00 : (red > 0xff) ? 0xff : red;

        green= qGreen(image_data[i]) + bright_value;
        green = (green < 0x00) ? 0x00 : (green > 0xff) ? 0xff : green;

        blue= qBlue(image_data[i]) + bright_value;
        blue =  (blue  < 0x00) ? 0x00 : (blue  > 0xff) ? 0xff : blue ;

        image_data[i] = qRgba(red, green, blue, qAlpha(image_data[i]));
    }
```

Contrast.cpp

Contrast.cpp 模块主要实现的是对于图像对比度的处理。图像对比度的处理与图像亮度的处理类似。在该模块，我们通过获取图片像素点 RGB 分量值，乘以饱和度系数（param）来进行饱和度的调节。(pixel 代表图片上的像素点，对像素点进行遍历调节饱和度，并且对红、绿、蓝三种颜色通道同时进行调节)

```
void Menu::contrast_funtion()
{
    int pixels = image->width() * image->height();
    unsigned int *data = (unsigned int *)image->bits();

    int red, green, blue, nRed, nGreen, nBlue;


    float param = 1 / (1 - contrast_value) - 1;

    for (int i = 0; i < pixels; ++i)
    {
        nRed = qRed(data[i]);
        nGreen = qGreen(data[i]);
        nBlue = qBlue(data[i]);

        red = nRed + (nRed - 127) * param;
        red = (red < 0x00) ? 0x00 : (red > 0xff) ? 0xff : red;
        green = nGreen + (nGreen - 127) * param;
        green = (green < 0x00) ? 0x00 : (green > 0xff) ? 0xff : green;
        blue = nBlue + (nBlue - 127) * param;
        blue = (blue < 0x00) ? 0x00 : (blue > 0xff) ? 0xff : blue;

        data[i] = qRgba(red, green, blue, qAlpha(data[i]));
    }
}
```

Saturate.cpp

Sacurate 模块主要实现的是对于图片饱和度的调节，其主要原理为：

① 计算每个像素点三基色最小值和最大值

② delta 为两值之差 / 255，如果两值之差为 0 则不做操作

value 为两值之和 / 255

③ 有 RGB 图像空间转化成 HSL（H 色调，S 饱和度，L 亮度）

L = value / 2

如果 L < 0.5 则 S = delta / value;

否则 S = delta / (2 - value);

④ Increment / 100 为饱和度，正值为提升饱和度，负值为降低饱和度。根据不同公式得到新的 rgb 值。（代码较长）

水印的添加：

水印主要分为图片水印和文字水印，因此我们也设置了两个模块对水印的添加进行分析：

Water_mark.cpp:

该模块主要实现的是文字水印的添加。其具体实现方法如下：首先我们先初始化一张新的图片，并且将已初始化得图片作为背景。之后在图片中添加文字，通过一定得算法，按照一定得公式将新建图片得位点与已知图片位点相加。同时判断相加位点是否超过 255，小于 255 则对位点值重新写入，大于 255 则将位点得像素值置为 0.

```
// first fill into the picture as the ground
painter.fillRect(start_x,start_y,start_x + 150,start_y + 45,Qt::transparent);
// set the word written into the picture
QFont ft = painter.font();
ft.setPixelSize(40);
painter.setFont(ft);
painter.drawText(start_x,start_y,start_x + 360,start_y + 45,Qt::AlignLeft,"CYW&PZFDL TQ

QRgb rgbSrc,rgbMark;
int r,g,b;
float alpha = 0.6, beta = 1- alpha;

// write into the picture pixel by pixel
// simliar to contrast\luminance\saturate function
for(int x = 0; x < image->width(); x++)
{
   for(int y = 0; y < image->height(); y++)
   {
        rgbSrc = image->pixel(x,y);
        rgbMark = image->pixel(x,y);

        // calcate the red\green\blue value of each pixel
        r = int(qRed(rgbSrc) * alpha + qRed(rgbMark) * beta);
        g = int(qGreen(rgbSrc) * alpha + qGreen(rgbMark) * beta);
        b = int(qBlue(rgbSrc) * alpha + qBlue(rgbMark) * beta);

        // to prevent overflow or underflow
        // each range from 0 to 255
        r = (0 <= r && r <= 255) ? r : 0;
        g = (0 <= g && g <= 255) ? g : 0;
        b = (0 <= b && b <= 255) ? b : 0;
        // write into the pixel
        image->setPixel(x,y,qRgb(r,g,b));
```

Picture_mark.cpp

该模块主要实现的是图片水印的添加。当两幅图像尺寸相同时，给图像加上水印效果其实很简单，就是简单的将一幅图像加到另一幅图像上（两幅图像乘以一定的系数就可以控制相加的效果，相加指的是像素点位值得增加）。当图像的尺寸不同时，需要在较大的图像上定义一个与较小的图像尺寸相同的感兴趣的区域 ROI，把较小的图像加到感兴趣区域上。若出现 ROI 区域的像素点有可能超过 255，导致饱和而显示出白色时可通过掩码在感兴趣区域点，只有较小图像的内容，而没有较大图像的内容。

```
void Menu::picture_mark_function()
{
    QColor color(255,255,255);  //白色
    QPainter imagepainter(image);  //新建画板
    imagepainter.setCompositionMode(QPainter::CompositionMode_SoftLight);   //设置重叠效果
    *mark_picture = mark_picture->scaledToWidth(50,Qt::SmoothTransformation);
    *mark_picture = mark_picture->scaledToWidth(50,Qt::SmoothTransformation);
    imagepainter.drawImage(0,0, *mark_picture);
    imagepainter.end();
}
```

Split.cpp

Splie.cpp 主要实现的是图像的分割。我们采用的是基于区域的图像分割技术，函数识别需要进行分割的图片的位置。我们在这里采用的是（左下、左上、右下、右上）四个不同的位置区域；可选择不同的区域按照一定的比例对图片进行切割。

```
void Menu::split_picture()
{
    switch (split_mode)
    {
        case left_up_corner:
            *image = image->copy(0,0,image->width() / 2,image->height() / 2);
            break;
        case right_up_corner:
            *image = image->copy(image->width() / 2,0,image->width(),image->height() / 2);
            break;
        case left_down_corner:
            *image = image->copy(0,image->height() / 2,image->width() / 2,image->height());
            break;
        case right_down_corner:
            *image = image->copy(image->width() / 2,image->height() / 2,image->width(),image-
            break;
    }
}
```

Zoom.cpp

Zoom.cpp 模块主要实现的是图片的缩放功能。我们首先在 factor.h 中定义了缩放的最大值与最小值（避免图片因过度放大或过分缩小而失真）。我们在函数中将放大或缩小倍数与最大放大缩小倍数进行比较，若没有超过临界值，则按倍数对图片进行放大（原有倍数值加上需放大的倍数）/缩小。反之则按临界值进行放大或缩小

```
void Menu::zoom_funtion()
{
    // to prevent over large/small zoom ratio
    if(zoom_in_or_out > 0)
        zoom_value = (zoom_value < max_zoom) ? (zoom_value + del_zoom) : max_zoom;
    else
        zoom_value = (zoom_value > min_zoom) ? (zoom_value - del_zoom) : min_zoom;
}
```

Image.cpp

Image.cpp 主要实现的是本次大作业所要求的基础功能，即绘制长方形和椭圆以及图像的重置。

长方形和椭圆的绘制：采用了 Qt 中自带的 Qpainter 函数进行绘制。椭圆的绘制需确定椭圆所在坐标位置（x,y）以及椭圆的长段半轴；长方形的绘制则需确定长方形所在的坐标位置以及其宽度与高度

```
void Menu::add_image_function()
{
    QPainter painter(image);
    painter.setPen(QPen(Qt::blue,4,Qt::SolidLine));
    switch (image_mode)
    {
        case ecllipse_mode:
            painter.drawEllipse(current_pos_x,current_pos_y,current_width,current_height);;
            break;
        case rectangular_mode:
            painter.drawRoundRect(current_pos_x,current_pos_y,current_width,current_height,50
            break;
    }
}
```

图形的重置：删除原有图形；new 一个新的图形，图形中的值按照 factor.h 中定义的初始值进行初始化。

```
void Menu::reset()
{
    if(image->isNull())     return;
    delete image;
    image = new QImage(initial_image);
    update_picture();
}
```

图片的打开与保存：

图片的打开与保存分别在 menu.cpp 以及 open_picture.cpp 中实现

Open_picture.cpp:

实现了图像的打开并且将图像显示在图形界面上，通过调用 Qt 自带的函数实现

```
void Menu::open_image_funtion()
{
    QStringList file_name_list;
    QString file_name;

    // Acquire the file name of the image
    file_name = QFileDialog::getOpenFileName(
                this, tr("open image file"),
                "./", tr("Image files(*.bmp *.jpg *.pbm *.pgm *.png *.ppm *.xbm *.xpm);;A

    // return when user cancel
    if(file_name.isEmpty())
        return;

    // case 2: open success
    image = new QImage(file_name);
    initial_image = *image;
    if(image->isNull())
        return;

    // create the scene and paint
    scene = new QGraphicsScene;
    scene->addPixmap(QPixmap::fromImage(*image));

}
```

图像的保存功能则在 menu.cpp 上进行详细阐述。

Menu

Menu 作为与图形界面交互的函数，具有并发相应以及操作保存功能，具体功能可参照 menu.h

Menu.h

在 menu.h 中定义了图形窗口界面，以及响应（鼠标点击或者鼠标上下滑动的函数）

```
class Menu : public QMainWindow//图形窗口界面
{
    Q_OBJECT

public://定义了关于鼠标点击或者鼠标上下滑动的函数
    explicit Menu(QWidget *parent = nullptr);
    void wheelEvent(QWheelEvent *event);
    void keyPressEvent(QKeyEvent *event);
    ~Menu();
```

与此同时，定义了操作响应函数，如打开图片点击、图片重置点击。其操作响应函数包含了本程序所提供的所有功能，定义了所有功能的触发函数。(即所有功能的触发均可通过鼠标点击相应 button 进行触发)，同时 menu 中还定义了 void Menu::keyPressEvent(QKeyEvent *event)函数,，相关操作也可通过按键进行触发。

```
void on_open_image_clicked();
void on_luminance_image_valueChanged(int arg1);
void on_saturation_image_valueChanged(double arg1);
void on_contrast_image_valueChanged(int arg1);
void on_Reset_clicked();
void on_save_image_clicked();
void on_rotate_clicked();
void on_zoom_in_image_clicked();
void on_zoom_out_image_clicked();
void on_split_image_clicked();
void on_Water_mark_image_clicked();
void on_Histogram_clicked();
void on_ellipse_clicked();
void on_rectangular_clicked();
void on_picturemark_clicked();
```

定义了各种操作所需的操作变量以及储存的临时变量

```
Ui::Menu* ui;
QImage* image;
QImage initial_image;
QImage* histgram_picture;
QImage* mark_picture;
QGraphicsScene* scene;
QGraphicsScene* hist_scene;
qreal zoom_value = 1;
int   bright_value;
int   last_bright_value;
double saturation_value;
float contrast_value;
qreal zoom_in_or_out;
int   split_mode;
int   split_mode_button;
QVector<int>* histgram;
int image_mode;
int current_pos_x;
int current_pos_y;
int current_width;
int current_height;
int paint_size;
int image_mode_button;
```

而 menu.cpp 里则详细实现了里面所包含的所有函数，由于相应操作实现的原理大同小异，我们就选择每一种功能选择其中一个函数进行详细分析

1. 鼠标点击相应对应操作

```
void Menu::on_zoom_in_image_clicked()
{
    if(image->isNull())     return;
    ui->graphics_view->scale(1 / zoom_value, 1 / zoom_value);
    zoom_in_or_out = 1;
    zoom_funtion();
    ui->graphics_view->scale(zoom_value,zoom_value);
}
```

Ui 类管理整个 UI 界面，当在方框内输入数值时，鼠标点击触发，调用已写好的缩放函数，实现交互和响应

2. 鼠标滑动实现对应操作

```
void Menu::wheelEvent(QWheelEvent *event)
{
    if(image->isNull())     return;
    ui->graphics_view->scale(1 / zoom_value, 1 / zoom_value);
    zoom_in_or_out = event->delta();
    zoom_funtion();
    ui->graphics_view->scale(zoom_value,zoom_value);
}
```

Qt 中自带的鼠标滑轮函数，当程序检查测到鼠标滑动的事件时，调用图片缩放函数，将缩放的比例与鼠标滑动方向、多少绑定

3. 图片点击保存

```
void Menu::on_save_image_clicked()
{
    if(image->isNull())     return;
    // use FileDialog to capture the saving path
    QString save_filepath = QFileDialog::getSaveFileName(this,tr("Save Image"),"",tr("Imag
    // similar to capature the screen
    QScreen *screen = QGuiApplication::primaryScreen();
    screen->grabWindow(image->save(save_filepath));
    QMessageBox::information(this,"Success","Success to save the picture");
}
```

4. 图片更新

新的得到的图片的位点信息覆盖旧的信息

```
void Menu::update_picture()
{
    if(image->isNull())     return;
    delete scene;
    scene = new QGraphicsScene;
    scene->addPixmap(QPixmap::fromImage(*image));
    ui->graphics_view->setScene(scene);
    ui->graphics_view->resize(400, 400);
    ui->graphics_view->show();
}
```

4、开发体会&小结

开发体会：

其实在刚刚确定实现的要求时觉得还是比较困难的。因为 Qt 大家都不熟悉，而且 Qt 语言的语法规则和其包含的库函数都是我们之前从未涉及到的。着手和入门的过程还是比较困难的。我们刚刚开始的时候设想了很多的高级功能，比如图片像素化、图片池化、降低噪点等。但是最后由于工程开发的时间不足，再加上这些功能的算法实现过于复杂，我们没有实施。但我们还是通过积极地搜集资料实现了饱和度、亮度、对比度地调整，并且额外增加实现了图片直方图地显示。算是基本上完成了我们之前设置地功能目标。我们再图形界面地设计上也出现了一下问题，本想图形界面比较没化，但最后由于操作难度地问题，仅实现了较为简洁地图形界面，但是这也达到了我们之前形成图形界面地要求，并且这并不影响我们这个图片编辑器功能地强大。

在开发地过程中，我们也对 oop 程序设计原则有了更加明确地认识，同时增加了对 QT 软件的运用与了解。在整个过程中大家合作的也非常愉快，合作过程中展现了团队的魅力

小结：

本次 OOP 大程的设计算是基本完成了中期报告中所提出的功能需求。并且由于大家齐心协力完成地较早，我们也有更多地时间对程序进行优化以及报告地撰写。希望我们能够加强对图想编辑方面算法地学习，以后若遇到相同地开发图像设计软件地机会时可以实现更多地高级功能

5、测试报告

5.1 基础界面

如图 5-1 所示即是基础的图形界面。可将图像文件保存、打开，具有图像（圆、长方形）编辑功能。多个按键分别对应了不同的功能。

点击了“Rectangle”和“Ellipse”后，就会进入图形编辑模式，分别可以在图像中画出椭圆和长方形。如图 5-2 和图 5-3 所示。

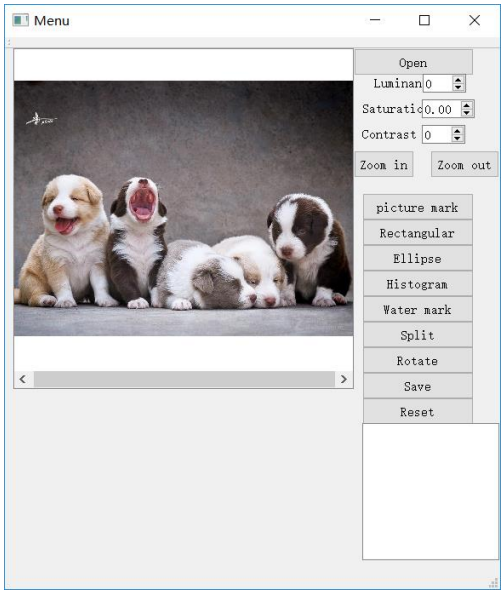


图 5-1 基础界面图

图 5-2 椭圆功能效果图

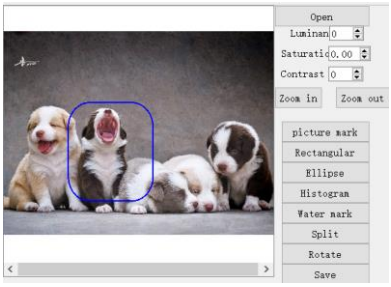


图 5-3 长方形编辑效果图

5.2 图像的尺寸调整功能

点击“zoom in”或者“zoon out”，具有图像大小调整功能。点击不同的按钮分别有平移、缩放、旋转、分割功能。

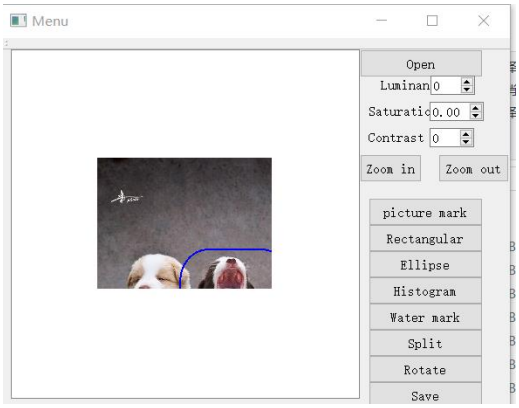


图 5-4 图像分割效果图

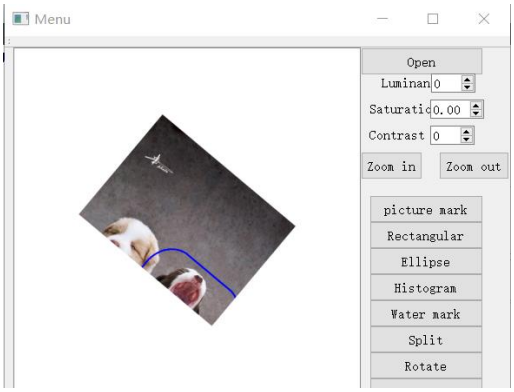


图 5-5 图像旋转效果图

5.3 显示直方图

点击“Histogram”按钮后，会显示当前图像的直方图，如图 5-6 所示。

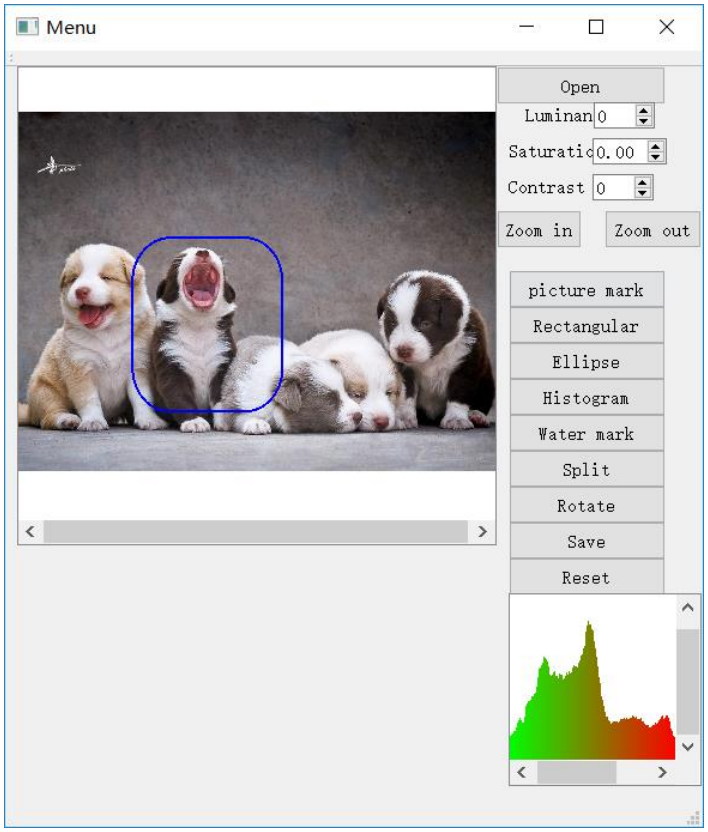


图 5-6 图像直方图效果图

5.4 亮度、对比度、饱和度的调节

如图 5-7 所示是测试所用图像的原图，是为了用来对照各种效果的对照图。

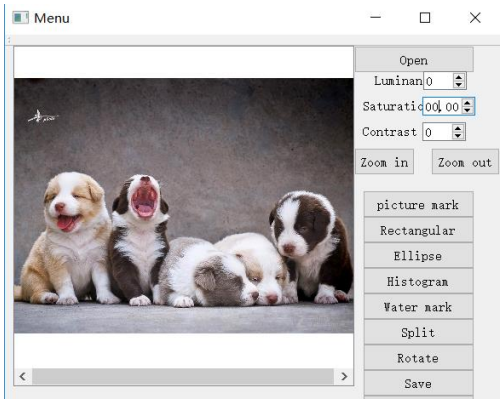


图 5-7 对照图

调整图像的饱和度，在 Saturation 栏内输入 10，显示效果如图 5-8 所示。图像色彩明显变得更加显眼。经过多次测试之后，确认饱和度功能已经实现。

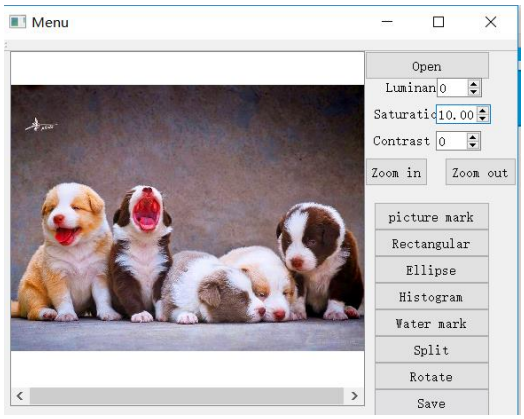


图 5-8 饱和度调整效果图

调整图像的饱和度，在 Luminance 栏内输入 50，显示效果如图 5-9 所示。图像变得更加亮。经过多次测试之后，确认亮度的调整功能已经实现。

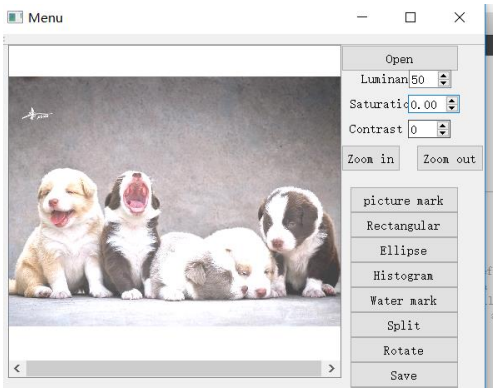


图 5-9 亮度调整效果图

调整图像的对比度，在 Contrast 栏内输入 10，显示效果如图 5-10 所示。从结果看出，图像中明亮的部分和暗的部分之间的差别更加明显，实现了对比度的增大。经过多次测试之后，确认对比度的调整功能已经实现。

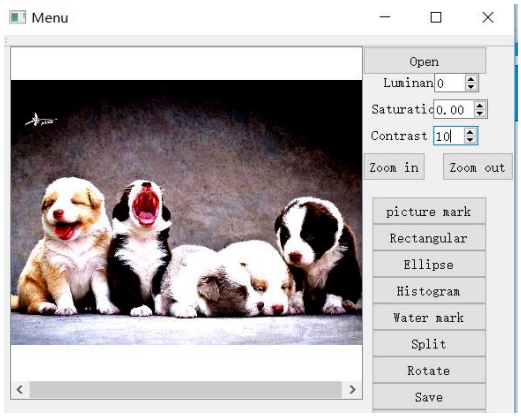


图 5-10 对比度调整效果图

5.5 文字水印、图片水印

点击“water mark”按钮之后，在图像中输入文字，就可以在图片上添加文字水印，如图 5-11 所示。

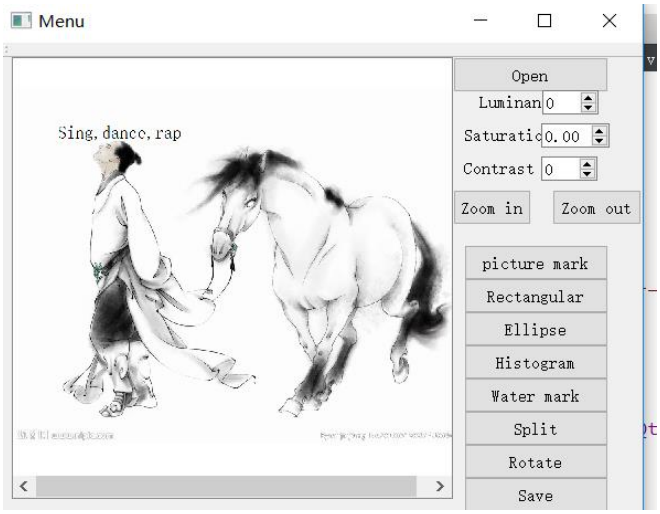


图 5-11 文字水印效果图

点击“picture mark”按钮之后，选择图片文件，就可以在图像的左上角添加图片类型的水印，如图 5-12 所示。

图 5-12 图片水印效果图

6、组内成员分工及互评

在 report 中，给出每个组员完成情况。组内互评，给每个组员打分(3 分制)，如共 5 个成员小组：

张童童（姓名） 3160101315（学号） 3 （分数）

江如蓝（姓名） 3160103777（学号） 3 （分数）

王钟毓（姓名） 3170105709（学号） 3 （分数）

陈宇威（姓名） 3170105706（学号） 3 （分数）

彭子帆（姓名） 3170105860（学号） 3 （分数）

7、附录

（体现团队合作方面的证据，如小组会议记录等体现良好的沟通能力，如邮件关键内容屏幕截图）

小组线上会议记录 1：

∞(937427981) 10:23:55
我们等会
∞(937427981) 10:23:59
讨论一下做什么
∞(937427981) 10:24:01
分工吧
∞(937427981) 10:24:07
10：45
王钟毓<wzy0173@qq.com> 10:25:44
收到
王钟毓<wzy0173@qq.com> 12:18:37
那要不我们晚上9点叭
王钟毓<wzy0173@qq.com> 12:18:40
@ 277 @奥拉夫
江如蓝(805034982) 12:18:51
ok
张童童(290410607) 12:23:53
ok
王钟毓<wzy0173@qq.com> 21:03:52
大家好了嘛w
王钟毓<wzy0173@qq.com> 21:04:33
要不我们语音说一下吧
∞(937427981) 21:04:56
我好了
王钟毓<wzy0173@qq.com> 21:06:26
@ 277 @奥拉夫
陈宇威(1051260524) 21:09:08
三个人面对面语音
张童童(290410607) 21:11:01

张童童(290410607) 21:11:15

sorry啊大家我大概不能语音

王钟毓<wzy0173@qq.com> 21:11:22

emm那我们打字吧

张童童(290410607) 21:11:29

可以的

∞(937427981) 21:12:08

那我们先来决定选题吧

∞(937427981) 21:12:35

- 题目 1：企业工资管理系统

  目标：

  1、企业最起码有管理人员、技术人员、工人三类人员，每一类人员有不同的工资计算方法（具体计算方法自行设计）；

  2、可以存储、显示、修改和删除企业人员信息。

  3、可以查询、统计工资信息。

- 题目 2：文本编辑器

∞(937427981) 21:12:43

- 题目 3：学生成绩管理系统

  目标：

  1、有不同类别的学生（本科生、硕士生和博士生）；

  2、实现学生成绩的录入、显示、查询、修改和删除等功能。

- 题目 4：图形编辑器

  目标：

  1、图形文件的读取和保存

  2、各种基本线条，图形（圆、长方形、三角形）的绘制。

- 题目 5：邮件系统

  目标：

  1、支持基本的用户注册和管理。

江如蓝(805034982) 21:19:11

(想做图形)

∞(937427981) 21:19:19

那就做吧真的

∞(937427981) 21:19:21

我们都行

张童童(290410607) 21:19:42

我也都行

张童童(290410607) 21:19:44

江如蓝(805034982) 21:19:52

图形可以吗?

王钟毓<wzy0173@qq.com> 21:19:57

可以挖

∞(937427981) 21:19:57

我们4还是7

王钟毓<wzy0173@qq.com> 21:20:08

奥拉夫学姐 你来决定叭

∞(937427981) 21:20:11

可以的

∞(937427981) 21:20:13

hhh

江如蓝(805034982) 21:20:49

那7?

王钟毓<wzy0173@qq.com> 21:21:00

∞(937427981) 21:24:16

要不就VS怎么样

张童童(290410607) 21:24:32

我也是VS

张童童(290410607) 21:24:34

王钟毓<wzy0173@qq.com> 21:24:41

...xcode 哭了

∞(937427981) 21:25:00

mac走开

张童童(290410607) 21:25:02

其实我也有xcode

∞(937427981) 21:25:03

我们不是一族的了

张童童(290410607) 21:25:04

王钟毓<wzy0173@qq.com> 21:25:34

这个问题不大emm

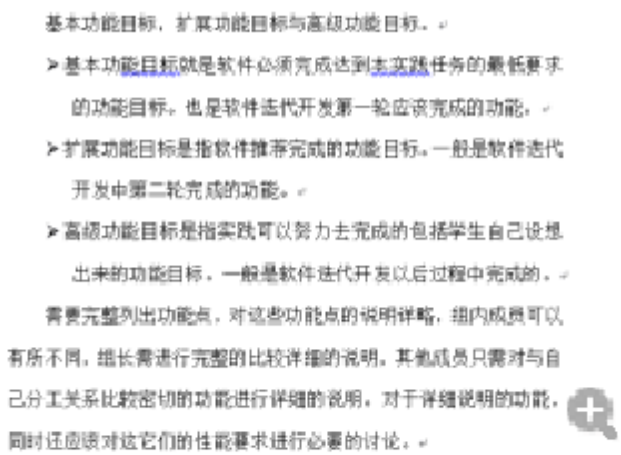
王钟毓<wzy0173@qq.com> 21:25:47

到时候分开写 我和他们住的比较近

王钟毓<wzy0173@qq.com> 21:25:52

0-0可以到他们电脑上

∞(937427981) 21:26:17


基本功能目标，扩展功能目标与高级功能目标。

➢基本功能目标就是软件必须完成达到本实践任务的最低要求的功能目标。也是软件迭代开发第一轮应该完成的功能。

➢扩展功能目标是指软件推荐完成的功能目标。一般是软件迭代开发中第二轮完成的功能。

➢高级功能目标是指实践可以努力去完成的包括学生自己设想出来的功能目标，一般是软件迭代开发以后过程中完成的。

需要完整列出功能点，对这些功能点的说明详略，组内成员可以有所不同，组长需进行完整的比较详细的说明。其他成员只需对与自己分工关系比较密切的功能进行详细的说明，对于详细说明的功能，同时还应该对这它们的性能要求进行必要的讨论。

∞(937427981) 21:26:24

我们有最终的目标的

∞(937427981) 21:26:30

老师提供的是基本功能

∞(937427981) 21:26:49

我们期中报告是需要列出我们的功能目标

∞(937427981) 21:27:29

也不一定要现在订，但是接下来做的时候我们需要不断提出来

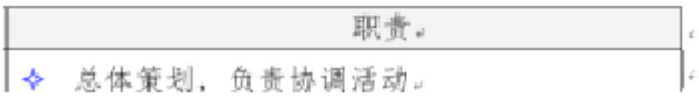
王钟毓<wzy0173@qq.com> 21:28:28

👌

∞(937427981) 21:29:08

这是我们需要进行的分工

∞(937427981) 21:29:08


职责
✧ 总体策划，负责协调活动

小组线上会议记录 2：


一、IDE介绍\Complier\库\人员职责【文档撰写】 PZF
二、项目需要完成的功能目标（参考群文件word）
库的下载、安装【代码部分】【涉及所有功能的调用库 可用库的数量与编程难度成反比】江如蓝、张童童
Project总体架构设计【代码部分】王钟毓
基本功能【文档部分】陈宇威
扩展功能【文档部分】江如蓝
高级功能【文档部分】陈宇威
基本实现思路【文档部分】王钟毓
三、附录【文档撰写】-张童童
标题【宋体 14号 加粗】 正文宋体14号 1.5倍行距
DDL下周五晚上23：59分 - 之后进行中期报告相应的优化
我们这样分工一下吧
王钟毓
收到
2019/4/7 11:06:15
张童童
ok
2019/4/9 21:43:39
王钟毓
我们图形界面用QT 好不好呀 大家w
2019/4/9 21:44:09
王钟毓
求各位可怜一下没有双系统的废物□
2019/4/9 22:54:36
行啊

小组线上会议记录 3:

oop team

聊天 公告 文

王钟毓

https://www.qt.io/offline-installers

王钟毓

Linux Host

- Qt 5.12.3 for Linux 64-bit (1.4 GB) (info)

macOS Host

- Qt 5.12.3 for macOS (3.2 GB) (info)

Windows Host

- Qt 5.12.3 for Windows (3.7 GB) (info)

Source packages & Other releases

The source code is available:

- For Windows users as a single zip file (792 MB) (Info)
- For Linux/macOS users as a tar.xz file (484 MB) (Info)

王钟毓

大嘎下载windows的这个 3.7G

好大

大大大大大

2019/4/18 21:50:34

张童童

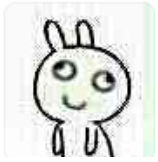
真滴大23333

王钟毓

嘻嘻嘻 安装完更大

我电脑

张童童



小组会议照片记录：