ECE 484W: Computer Engineering Design I

Project #1

Author: Victor Olabiyi

UIN: 01210388

Group Member: Enoch Ampong

Due Date: 09/12/2023

**Honor Pledge**:

"I pledge to support the Honor System of Old Dominion University. I will refrain from any form of

academic dishonesty or deception, such as cheating or plagiarism. I am aware that as a member of the

academic community, it is my responsibility to turn in all suspected violators of the Honor System. I will

report to an Honor Council hearing if summoned."

Signed: Victor Olabiyi

Abstract

The purpose of this assignment is to introduce the theory of image processing, as well as the

required techniques to design execute the given tasks, which focuses on processing given image. The

first task of this assignment is to create a graphical interface to generate an image that consist of another

laid on it. The second task would be a continuation of the first, but additional elements such as

brightness and contrast slider are added to affect the image. This would be the first time using the Qt

software, with lead to countless of problems, while attempting to navigate through the software.

However, the team was able to come up with a solution, completing the given tasks. The majority of this

assignment consists of just programming graphical user interfaces using the program Qt creator.

Table of Contents

Table of Contents

[1 Introduction: 4](#_Toc115473045)

[1.1 Objective: 4](#_Toc115473046)

[1.2 Overlaying user-selected image: 4](#_Toc115473047)

[1.3 Histogram equalization: 4](#_Toc115473048)

[2 Methodology: 4](#_Toc115473049)

[3 Results and Analysis: 6](#_Toc115473050)

[4 Alternative Design: 9](#_Toc115473051)

[5 Broader Impact 10](#_Toc115473052)

[6.1 Societal Impact 10](#_Toc115473053)

[6.2 Environmental Impact 10](#_Toc115473054)

[6.3 Societal Impact 10](#_Toc115473055)

[6.4 Global Impact 10](#_Toc115473056)

[6 Individual contributions 11](#_Toc115473057)

[References 12](#_Toc115473058)

[Appendix 12](#_Toc115473059)

[Task 1: 12](#_Toc115473060)

[Task 2: 14](#_Toc115473061)

1. Introduction:

This assignment consists of two tasks. The first task was to overlay two user-selected images. The second

task was to use some elements in the Qt creator software to affect the image brightness and contrast.

The tasks introduced to us, image processing. Both tasks were done using the Qt creator software.

* 1. Objective:

The objective of this assignment is to create a graphical user interface to complete two tasks. The first

tasks required that two images, a user-select 8-bit bitmap original and overlay image, be loaded, and

displayed onto the GUI, graphical user interface. Along with that, an output should be able to be

generated by overlaying the overlay image over the original image. The second tasks extend the first but

requires that only a user-select 8-bit bitmap original image is to be loaded, and the functionalities of

brightness and contrast adjustments should be added. In detail, a slider should be added which would be

allow for the adjustment to be made. With brightness, the slider sets a factor by which the brightness is

multiplied. For the contrast, the slider configures a transfer function that either compresses the

brightness histogram for lower contrast or stretches it for higher contrast.

* 1. Overlaying user-selected image:

The technique of overlaying user selected images is a very useful technique since it’s present in many

applications nowadays. For example, subtitles can be seen in many movies or Tv shows. Another would

be logos, and other various types of information. The idea of this technique is quite straightforward as

it’s basically just having a base image and overlaying another unto it.

1. Methodology:

The first task utilized the clicked functionality of Qt creator which created a void function in the

mainWindow.cpp file that allow for us to include the function to load the user-selected file. selected file.

The second task extends the idea of loading a user-select 8-bitbitmaporiginal image and instead include

some UI controls which allow for its brightness and contrast to be altered.

2.1 Task 1:

There was not much of complication doing the first task. There were three main functions used to

complete this task. The first two functions **on\_pushButton\_clicked ()** and **on\_pushButton\_2\_clicked ()**

acts as a trigger whenever either of the two push buttons are clicked. Furthermore, on the functions,

they get the file name that a use would like to use, it then checks if the file name is not null, it is null if no

file were selected. It then proceeds to load the image using the file name. If the image is valid, it shall be

displayed to the window. It allows you to choose the window to upload the image to, so it is important

to choose the right one to avoid wrongly placed images.

The third function, **on\_pushButton\_3\_clicked ()**, works a differently than the first two. First, the base

and overlay images are loaded as **QPixmap**. **QPixmap** is used because it allows for loaded images to be

displayed onto a display widget. The overlay was performed by creating a transparent QPixmap image,

and a **QPainter** object was created using the QPixmap as a foundation, draws the base, then the overlay

on top. In all, the display widget resulting image was displayed onto the window; the display widget.

2.2 Task 2:

Understanding the second task is rather complicated. The first function extends the loading of image

from the first task. The difference now is now that the loaded image is displayed twice. The reason for

this is that the first window is used as an original image while the other would constantly change.

Furthermore, the are two sliders within the UI, one for brightness and the other for contrast. For the

brightness, if the value of the slider is changed (moving up or down), the function

**on\_vertical\_slider\_brightness\_valueChanged ()** would be triggered and will pass the value on which the

slider is on. Inside the function, the original image is loaded, then **adjust\_Brightness ()** is called and

passed the slider value onto the image. Inside the **adjust\_Brightness ()** function are declared, some

required variables. It gets the total amount of pixels of the passed image and sets a pointer to the first

pixel data. The pixels are then iterated through, the rgb values of the image are then configured by

adding the factor, slider value, then checking of the either the red, green, or blue values are less than

0x00 or greater than 0xff. After, the rgb of that specific pixel is then changed and the iteration continues

throughout the rest. Lastly, the altered image is returned to the previous function. Once returned, a

**Pixmap** is created of the image, and it is then displayed onto the window. The contrast slider works

similar, but instead of adding the factor to the rgb values, it is added to a created contrast variable,

which is (100+factor)/100.0, Then inside the loop, the rgb values are equated to the contrast times the

existing pixel data for that color minus 128. The rest is similar to the brightness’s function. For the

**on\_reset\_button\_clicked ()** function, it acts as one would expect. Replace the changed image with the

original image and reset the slider values to zero. For the **on\_save\_button\_clicked ()** function the altered

image is saved using the save function using the filename of the altered image.

1. Results and Analysis:

For the result, a video was recorded showing the expected outcome of this assignment. The

expected result for the first task is to show the loaded images and the generated overlayed image. This

shows how the user would select the 8-bit bitmap image.

A screenshot of a computer

Description automatically generated

The below image shows the base and the overlay images.

A screenshot of a computer

Description automatically generated

The below image shows the result as it generates the overlaying images.

A screenshot of a computer

Description automatically generated

Below is the altered image with the brightness at max.

A screenshot of a computer

Description automatically generated

Image below is the altered image with the contrast at max.

A screenshot of a computer

Description automatically generated

Image below is how one would save the altered file.

A computer screen shot of a computer screen

Description automatically generated

Image below is the saved altered image.

Close-up of a pile of round objects

Description automatically generated

1. Alternative Design:

For this assignment, since not much experience is present coming up with alternative designs proves

to be challenging.

For the first task, an alternate design would include maybe iterating through the first image

pixels and replacing it with that of the second. Basically, taking the pixels of the overlay image and

iterating through the base image and replacing the pixels in the specified positions to that of the

overlay’s pixel.

For the second task, one could work on allowing for the image brightness and contrast changes

to be synchronized. At the current moment, image can only be altered one way, so an idea would

be to include some variables in the header files which could save the already altered image, via increase

in brightness or contrast and use that when calling either the contrast or brightness changing functions.

1. Additional Considerations:

I do not have other considerations aside of the ones mentioned in the alternative design.

1. Broder Impact:

This assignment could have many impacts on economic, environmental, global, and societal

contexts. Mainly with image enhancement, the impact could be large. Image is intangible the way

something looks and is perceived, how people view an image could vary per individual as they will be

drawn and focused on different aspects of the image in this case, the city [1]. Image enhancement in a

city aspect is giving it a new outlook, revamping, and regenerating what it already must show off and

adding new aspects to help the city reach its full potential and generate a boost in the economy,

environment, infrastructure, and tourism [1]. With the boost in tourism, one could say economic impact

would include foreign exchange earnings. For environmental impacts there would include transportation

problems. Following, a societal impact would include preservation of local culture. Lastly, a global impact

would be the negative effect on the natural resource of the world.

6.1 Societal Impact

With the impacts of economic the details could be endless. For one, Tourism expenditures

generate income to the host economy [2]. The money that the country makes from tourism can then be

reinvested in the economy [2].

6.2 Environmental Impact

Along with the impacts of economic, with environmental impacts, the details could be endless.

With environmental, 55% of global traffic movement occurs for tourism. If we didn’t go on holidays,

especially across the world, that would split the need for transportation in approximately half [3].

6.3 Societal Impact

Tourism impact on societal would include preserving the local culture and heritage;

strengthening communities; provision of social services; commercialization of culture and art;

revitalization of customs and art forms and the preservation of heritage [4].

6.4 Global Impact

With global impacts, it’s details could be really described. With global impact, due to the

booming success of the tourism industry, there is an increase of recreational facilities and an increase of

construction for the tourism sector [5]. This increases the pressure towards the natural resources that

the world has to offer directly [5].

1. Individual Contributions:

The tasks were completed with my partner, Enoch Ampong. We collaborated in getting the work

done.

Reference Page

## References

1. “Impact of Image Enhancement on Event Regeneration and Housing Displacement,” www.ukessays.com. https://www.ukessays.com/essays/tourism/impact-of-image-enhancement-on-event-regeneration-and-housing-displacement.php#:~:text=Image%20enhancement%20in%20a%20city%20aspect%20is%20giving (accessed Sep. 20, 2022).
2. H. Stainton, “Economic impacts of tourism,” Tourism Teacher, Jun. 27, 2019. https://tourismteacher.com/economic-impacts-of-tourism/
3. ‌admin, “NEGATIVE IMPACTS OF TOURISM YOU SHOULD KNOW,” LAURA IN WATERLAND, Jul. 25, 2020. https://laurainwaterland.com/negative-impacts-of-tourism/#:~:text=NEGATIVE%20IMPACTS%20OF%20TOURISM%20YOU%20SHOULD%20KNOW%201 (accessed Sep. 20, 2022).
4. ‌H. Stainton, “Social impacts of tourism,” Tourism Teacher, Jul. 19, 2019. https://tourismteacher.com/social-impacts-of-tourism/
5. ‌“Global Tourism Negative Impacts Of Tourism | Blablawriting.com,” Bla Bla Writing, Sep. 23, 2016. https://blablawriting.net/global-tourism-negative-impacts-of-tourism-essay (accessed Sep. 20, 2022).

Appendix

Task1:

mainwindow.h

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <QFileDialog>

QT\_BEGIN\_NAMESPACE

namespace Ui { class MainWindow; }

QT\_END\_NAMESPACE

class MainWindow : public QMainWindow

{

Q\_OBJECT

public:

MainWindow(QWidget \*parent = nullptr);

~MainWindow();

private slots:

void on\_pushButton\_clicked();

void on\_pushButton\_2\_clicked();

void on\_pushButton\_3\_clicked();

private:

Ui::MainWindow \*ui;

QImage image1, image2;

};

#endif // MAINWINDOW\_H

**Main.cpp**

#include "mainwindow.h"

#include <QApplication>

int main(int argc, char \*argv[])

{

QApplication a(argc, argv);

MainWindow w;

w.show();

return a.exec();

}

**mainwindow.cpp**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <QApplication>

#include <QFileDialog>

#include <QPixmap>

#include <QLabel>

#include <QPainter>

MainWindow::MainWindow(QWidget \*parent)

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

MainWindow::~MainWindow()

{

delete ui;

}

void MainWindow::on\_pushButton\_clicked()

{

QString fileImage = QFileDialog::getOpenFileName(this, tr("Choose"), "", tr("Images (\*.png, \*.jpg \*.jpeg \*.bmp)")); // gets existing file selected by user, null if cancel is selected

if(QString::compare(fileImage, QString()) != 0) { // checks for null

bool valid = image1.load(fileImage); // loads image from file name

if(valid){ // checks if image is valid

ui->image\_label->setPixmap(QPixmap::fromImage(image1)); // displays image to window

}

}

}

void MainWindow::on\_pushButton\_2\_clicked()

{

QString fileImage = QFileDialog::getOpenFileName(this, tr("Choose"), "", tr("Images (\*.png, \*.jpg \*.jpeg \*.bmp)")); // gets existing file selected by user, null if cancel is selected

if(QString::compare(fileImage, QString()) != 0) { // checks for null

bool valid = image2.load(fileImage); // loads image from file name

if(valid){ // checks if image is valid

ui->image\_label\_2->setPixmap(QPixmap::fromImage(image2)); // displays image to window

}

}

}

void MainWindow::on\_pushButton\_3\_clicked()

{

QPixmap base = QPixmap::fromImage(image1); // creates pixmap from loaded image

QPixmap overlay = QPixmap::fromImage(image2.scaled(200, 200)); // creates pixmap from loaded image scaled to fixed size

QPixmap result((base.width()-150), (base.height()-150)); // creates pixmap with lenght and height from base image

result.fill(Qt::transparent); // force alpha channel

{

QPainter painter(&result); // creates a painter object

painter.drawPixmap(0, 0, base); // draws base image onto the result pixmap

painter.drawPixmap(0, 0, overlay); // draws overlay image onto the result pixmap

}

ui->image\_label\_3->setPixmap(QPixmap(result)); // displays image to window

}

**mainwindow.ui**

**A screenshot of a computer

Description automatically generated**

**Task 2:**

**mainwindow.h**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

QT\_BEGIN\_NAMESPACE

namespace Ui { class MainWindow; }

QT\_END\_NAMESPACE

class MainWindow : public QMainWindow

{

Q\_OBJECT

public:

MainWindow(QWidget \*parent = nullptr);

~MainWindow();

QImage adjust\_Brightness(QImage &picture, int factor);

QImage adjust\_Contrast(QImage &picture, int factor);

private slots:

void on\_pushButton\_1\_clicked();

void on\_reset\_button\_clicked();

void on\_save\_button\_clicked();

void on\_vertical\_slider\_brightness\_valueChanged(int value);

void on\_vertical\_slider\_contrast\_valueChanged(int value);

private:

Ui::MainWindow \*ui;

QImage image1, image2;

};

#endif // MAINWINDOW\_H

**main.cpp**

#include "mainwindow.h"

#include <QApplication>

int main(int argc, char \*argv[])

{

QApplication a(argc, argv);

MainWindow w;

w.show();

return a.exec();

}

**mainwindow.cpp**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <QApplication>

#include <QFileDialog>

#include <QPixmap>

#include <QLabel>

#include <QPainter>

MainWindow::MainWindow(QWidget \*parent)

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

MainWindow::~MainWindow()

{

delete ui;

}

void MainWindow::on\_pushButton\_1\_clicked()

{

{

QString filename = QFileDialog::getOpenFileName(this, tr("Choose"), "", tr("Images (\*.bmp)"));

if(QString::compare(filename, QString()) != 0) {

bool valid = image2.load(filename);

if(valid){

ui->label\_picture1->setPixmap(QPixmap::fromImage(image2));

ui->label\_picture2->setPixmap(QPixmap::fromImage(image2));

}

}

}

}

QImage MainWindow::adjust\_Brightness(QImage &picture, int factor)

{

double red, green, blue;

int pixels;

unsigned int \*data;

pixels = picture.width() \* picture.height();

data = (unsigned int \*)picture.bits();

for (int i = 0; i < pixels; ++i)

{

red= qRed(data[i])+ factor;

red = (red < 0x00) ? 0x00 : (red > 0xff) ? 0xff : red;

green= qGreen(data[i])+factor;

green = (green < 0x00) ? 0x00 : (green > 0xff) ? 0xff : green;

blue= qBlue(data[i])+factor;

blue = (blue < 0x00) ? 0x00 : (blue > 0xff) ? 0xff : blue ;

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

}

return picture;

}

QImage MainWindow::adjust\_Contrast(QImage &picture, int factor)

{

double contrast;

double red, green, blue;

int pixels;

unsigned int \*data;

contrast = (100.0+factor)/100.0;

pixels = picture.width()\*picture.height();

data = (unsigned int \*)picture.bits();

//contrast -> newColor = 128 + contrast \*(oldColor - 128)

for (int i = 0; i < pixels; ++i)

{

red= 128+ contrast\*(qRed(data[i])-128);

red = (red < 0x00) ? 0x00 : (red > 0xff) ? 0xff : red;

green= 128+ contrast\*(qGreen(data[i])-128);

green = (green < 0x00) ? 0x00 : (green > 0xff) ? 0xff : green;

blue= 128+ contrast\*(qBlue(data[i])-128);

blue = (blue < 0x00) ? 0x00 : (blue > 0xff) ? 0xff : blue ;

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

}

return picture;

}

/\*void MainWindow::on\_groupBox\_3\_toggled(bool arg1)

{

}\*/

void MainWindow::on\_reset\_button\_clicked()

{

QImage picture;

QPixmap picturePixmap;

picture=ui->label\_picture1->pixmap().toImage();

picturePixmap=QPixmap::fromImage(picture);

ui->label\_picture2->setPixmap(picturePixmap);

ui->vertical\_slider\_brightness->setValue(0);

ui->vertical\_slider\_contrast->setValue(0);

}

void MainWindow::on\_save\_button\_clicked()

{

QString fullName;

fullName = QFileDialog::getSaveFileName(this, "Save", "modified\_picture.jpg", "\*.\*");

if (fullName=="")

{ return; }

ui->label\_picture2->pixmap().save(fullName);

}

void MainWindow::on\_vertical\_slider\_brightness\_valueChanged(int value)

{

QImage picture;

QPixmap picturePixmap;

picture=ui->label\_picture1->pixmap().toImage();

picture=adjust\_Brightness(picture, value);

picturePixmap=QPixmap::fromImage(picture);

ui->label\_picture2->setPixmap(picturePixmap);

}

void MainWindow::on\_vertical\_slider\_contrast\_valueChanged(int value)

{

QImage picture;

QPixmap picturePixmap;

picture=ui->label\_picture1->pixmap().toImage();

picture=adjust\_Contrast(picture, value);

picturePixmap=QPixmap::fromImage(picture);

ui->label\_picture2->setPixmap(picturePixmap);

}

**mainwindow.ui**

**A screenshot of a computer

Description automatically generated**