CENTRO UNIVERSITARIO DE CIENCIAS EXÁCTAS E INGENIERÍAS

Tarea 02

Ecualización de imagen

Aldo Alexandro Vargas Meza

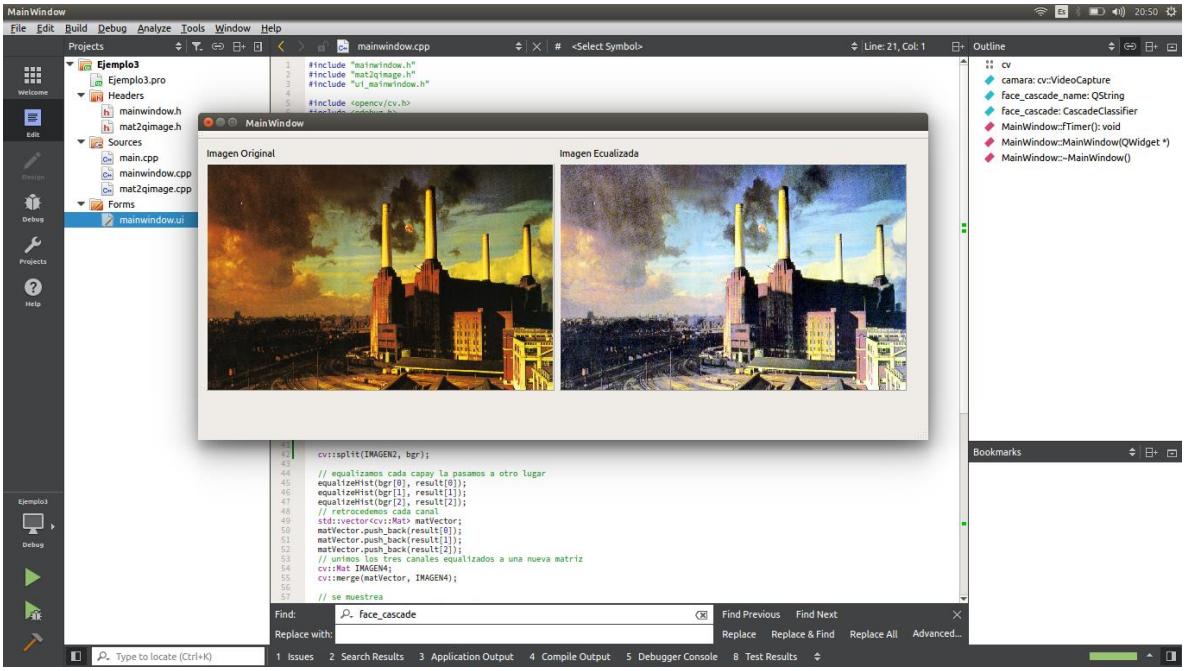
13/10/2017

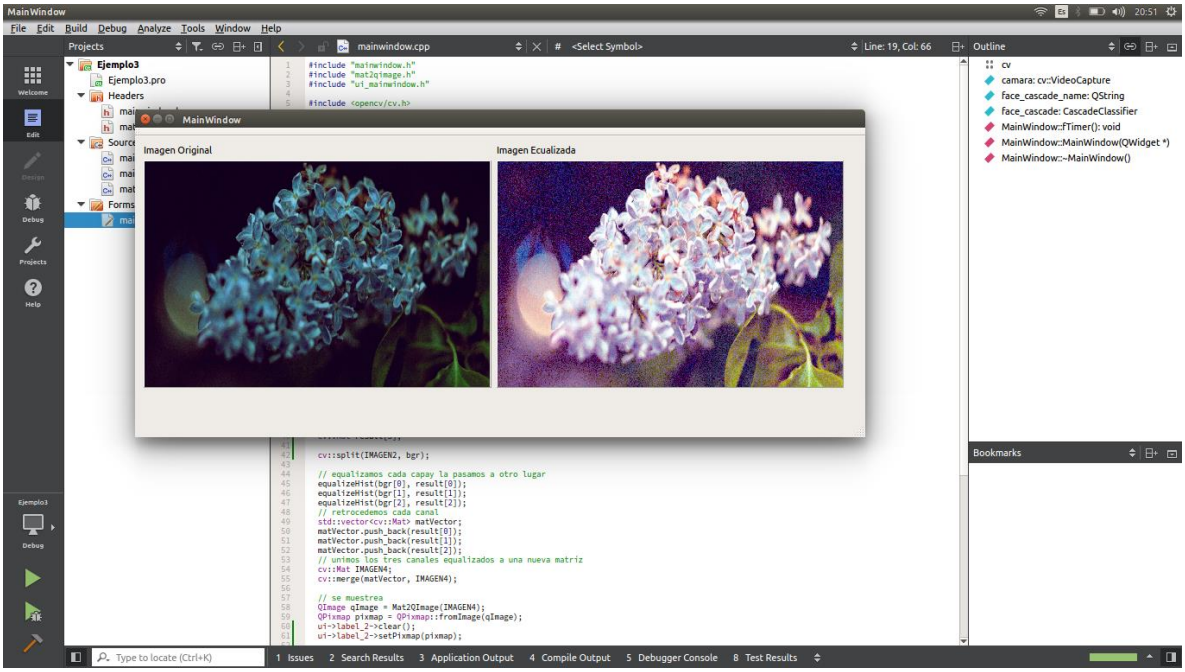
Resumen

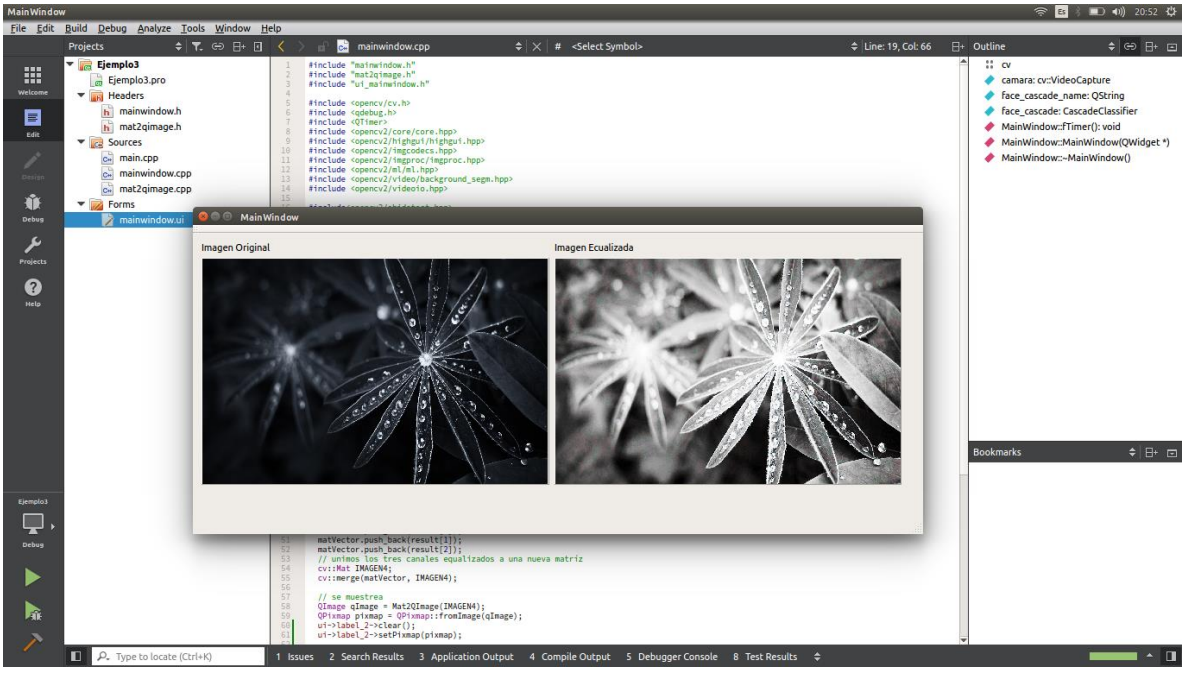
La tarea consiste en tratar las capas que componenen a una imagen, para poder resaltar los bordes mas sutiles. Con esto, se busca encontrar diferentes patrones que en la imagen original es mas complicado.

Cabe mencionar que la utilización de este técnica puede ser parte de un desarrollo o análisis mas profundo y profesional.

Introducción




MainWindow
Imagen Original
Imagen Ecualizada


MainWindow
Imagen Original
Imagen Ecualizada

Desarrollo:

Conclusión:

Códigos

```
/* LIbrerías y archivos incluidos */
#include "mainwindow.h"
#include "mat2qimage.h"
#include "ui_mainwindow.h"

#include <opencv/cv.h>
#include <qdebug.h>
#include <QTimer>
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgcodecs.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/ml/ml.hpp>
#include <opencv2/video/background_segm.hpp>
#include <opencv2/videoio.hpp>

#include <opencv2/objdetect.hpp>

/* MACROS */
#define IP 0
#define IMG_ADD "/home/aldo/Imágenes/Wallpapers/1505520401699.jpg"
#define CASCADE "../haarcascade_frontalface_alt2.xml"

using namespace cv;

QString face_cascade_name = QString(CASCADE);
CascadeClassifier face_cascade;

MainWindow::MainWindow(QWidget *parent)
   : QMainWindow(parent), ui(new Ui::MainWindow) {
 ui->setupUi(this);

 if (!face_cascade.load(face_cascade_name.toUtf8().constData())) {
  qDebug() << "Error al cargar el detector de rostros";
 }

 QTimer *cronometro = new QTimer(this);
 connect(cronometro, SIGNAL(timeout()), this, SLOT(fTimer()));
 cronometro->start(30);

 if (!camara.isOpened()) camara.open(IP);
```

```
}

MainWindow::~MainWindow() { delete ui; }

void MainWindow::fTimer() {
  /* Matrices para Imagen */
  Mat IMAGEN;
  Mat IMAGEN2;
  Mat IMAGEN3;
  Mat IMAGEN4;

  Mat GRIS;
  Mat rgb[3];
  Mat result[3];

  /* Lectura de Imagen */
  IMAGEN = cv::imread(IMG_ADD);
  /* Ajuste de dimension */
  cv::resize(IMAGEN, IMAGEN2, Size(400, 300), 0, 0, 0);

  cv::split(IMAGEN2, rgb);

  /* Ecualizacion y guardado por capas */
  equalizeHist(rgb[0], result[0]);
  equalizeHist(rgb[1], result[1]);
  equalizeHist(rgb[2], result[2]);

  std::vector<cv::Mat> matVector;
  matVector.push_back(result[0]);
  matVector.push_back(result[1]);
  matVector.push_back(result[2]);

  /* Union de 3 canales en la matriz */
  cv::merge(matVector, IMAGEN4);

  /* Impresion en Etiquetas */
  QImage qImage = Mat2QImage(IMAGEN4);
  QPixmap pixmap = QPixmap::fromImage(qImage);
  ui->label_2->clear();
  ui->label_2->setPixmap(pixmap);

  QImage qImage1 = Mat2QImage(IMAGEN2);
  QPixmap pixmap1 = QPixmap::fromImage(qImage1);
  ui->label->clear();
  ui->label->setPixmap(pixmap1);
}

#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
```

```
namespace Ui {
class MainWindow;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit MainWindow(QWidget *parent = 0);
    ~MainWindow();

public slots:

    void fTimer();
private slots:

private:
    Ui::MainWindow *ui;
};

#endif // MAINWINDOW_H

#-------------------------------------------------
#
# Project created by QtCreator 2017-08-18T16:20:48
#
#-------------------------------------------------

QT       += core gui

greaterThan(QT_MAJOR_VERSION, 4): QT += widgets

TARGET = pencv_videoio
TEMPLATE = app

# The following define makes your compiler emit warnings if you use
# any feature of Qt which as been marked as deprecated (the exact warnings
# depend on your compiler). Please consult the documentation of the
# deprecated API in order to know how to port your code away from it.
DEFINES += QT_DEPRECATED_WARNINGS

# You can also make your code fail to compile if you use deprecated APIs.
# In order to do so, uncomment the following line.
# You can also select to disable deprecated APIs only up to a certain version of Qt.
#DEFINES += QT_DISABLE_DEPRECATED_BEFORE=0x060000    # disables all the APIs deprecated before Qt
6.0.0

SOURCES += \
        main.cpp \
        mainwindow.cpp \
```

```
    mat2qimage.cpp

HEADERS += \
        mainwindow.h \
    mat2qimage.h

FORMS += \
        mainwindow.ui

INCLUDEPATH += /usr/local/include/opencv2

LIBS += -L/usr/local/lib -lopencv_core -lopencv_imgcodecs -lopencv_highgui -opencv_videoio

CONFIG += link_pkgconfig
PKGCONFIG += opencv
```