1. ЦЕЛЬ РАБОТЫ

Целью работы данный практики является изучение способов сегментации изображений. В этой работе будут рассматривать цветные трехканальные (**RGB**) изображения. Для изучения технологии сегментации изображения были выбраны случайные фотографии формата .ipg.

2. ИСХОДНЫЕ ДАННЫЕ

Сбор собственного высококачественного датасета изображений для сегментации очень трудоёмкое занятие. По этой причине было отобрано ограниченное количество изображений разной насыщенности, контрастности в формате .jpg для проверки написанного алгоритма сегментации изображения.

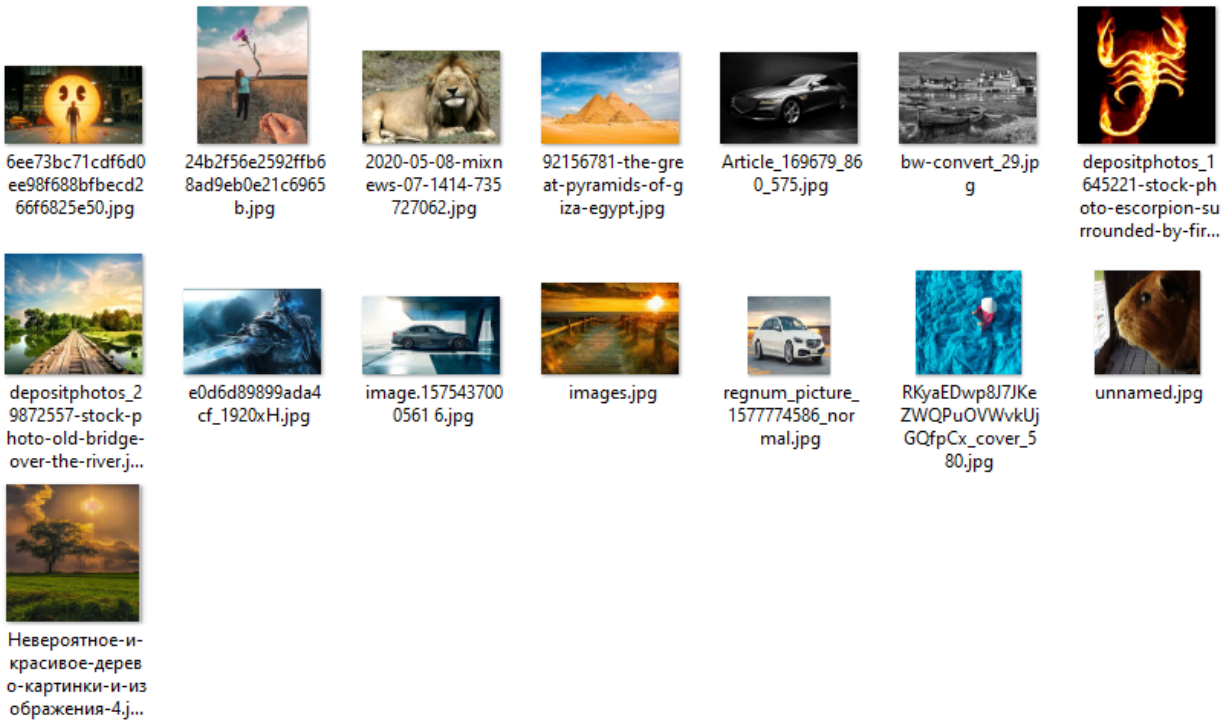

Рис. 1: Пример изображений

3. ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ

3.1. Сегментация изображения

В компьютерном зрении, сегментация — это процесс разделения цифрового изображения на несколько сегментов (множество пикселей, также называемых суперпикселями). Цель сегментации заключается в упрощении и/или изменении представления изображения, чтобы его было проще и легче анализировать. Сегментация изображений обычно используется для того, чтобы выделить объекты и границы (линии, кривые, и т. д.) на изображениях. Более точно, сегментация изображений — это процесс присвоения таких меток каждому пикселю изображения, что пиксели с одинаковыми метками имеют общие визуальные характеристики.

Результатом сегментации изображения является множество сегментов, которые вместе покрывают всё изображение, или множество контуров, выделенных из изображения. Все пиксели в сегменте похожи по некоторой характеристике или вычисленному свойству, например, по цвету, яркости или текстуре. Соседние сегменты значительно отличаются по этой характеристике.

3.2. Методы сегментации изображения

3.2.1. Методы, основанные на кластеризации

Метод k-средних — это итеративный метод, который используется, чтобы разделить изображение на K кластеров. Базовый алгоритм приведён ниже:

1. Выбрать *K* центров кластеров, случайно или на основании некоторой эвристики;
2. Поместить каждый пиксель изображения в кластер, центр которого ближе всего к этому пикселю;
3. Заново вычислить центры кластеров, усредняя все пиксели в кластере;
4. Повторять шаги 2 и 3 до сходимости (например, когда пиксели будут оставаться в том же кластере).

Здесь в качестве расстояния обычно берётся сумма квадратов или абсолютных значений разностей между пикселем и центром кластера.

Разность обычно основана на цвете, яркости, текстуре и местоположении пикселя, или на взвешенной сумме этих факторов. K может быть выбрано вручную, случайно или эвристически.

Этот алгоритм гарантированно сходится, но он может не привести к оптимальному решению. Качество решения зависит от начального множества кластеров и значения K.

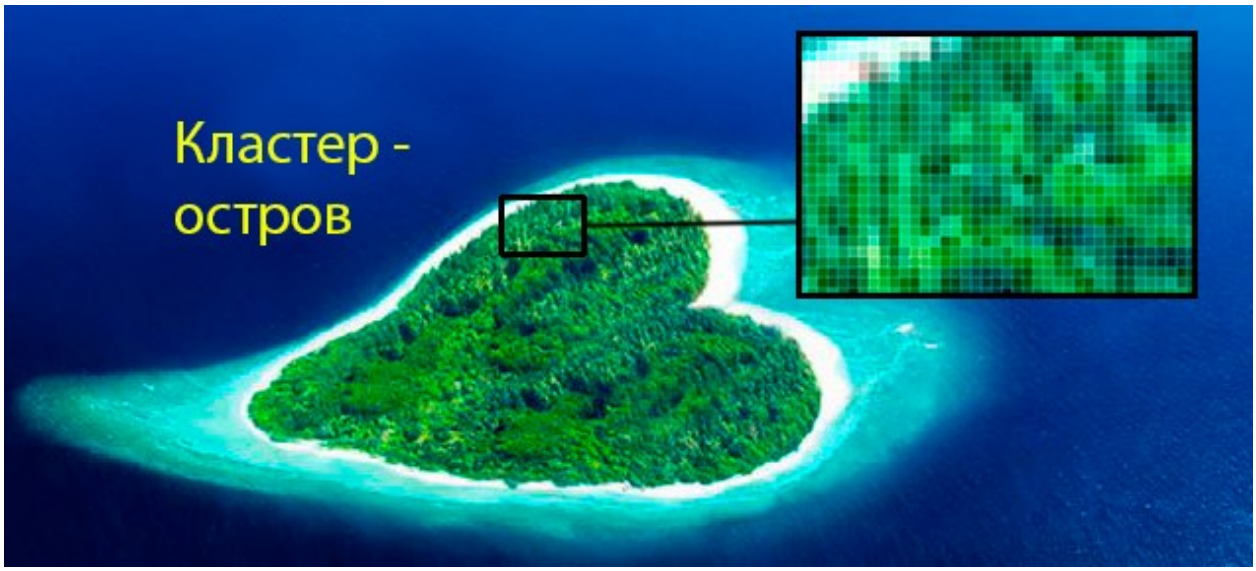

Рис. 2: Пример кластеризации изображения

3.2.2. Методы с использованием гистограммы

Методы с использованием гистограммы очень эффективны, когда сравниваются с другими методами сегментации изображений, потому что они требуют только один проход по пикселям. В этом методе гистограмма вычисляется по всем пикселям изображения и её минимумы и максимумы используются, чтобы найти кластеры на изображении. Цвет или яркость могут быть использованы при сравнении.

Улучшение этого метода — рекурсивно применять его к кластерам на изображении для того, чтобы поделить их на более мелкие кластеры. Процесс повторяется со всё меньшими и меньшими кластерами до тех пор, когда перестанут появляться новые кластеры.

Один недостаток этого метода — то, что ему может быть трудно найти значительные минимумы и максимумы на изображении. В этом методе

классификации изображений похожи метрика расстояний и сопоставление интегрированных регионов.

Подходы, основанные на использовании гистограмм, можно также быстро адаптировать для нескольких кадров, сохраняя их преимущество в скорости за счёт одного прохода. Гистограмма может быть построена несколькими способами, когда рассматриваются несколько кадров. Тот же подход, который используется для одного кадра, может быть применён для нескольких, и после того, как результаты объединены, минимумы и максимумы, которые было сложно выделить, становятся более заметны. Гистограмма также может быть применена для каждого пикселя, где информация используется для определения наиболее частого цвета для данного положения пикселя. Этот подход использует сегментацию, основанную на движущихся объектах и неподвижном окружении, что даёт другой вид сегментации, полезный в видеотрекинге.

Рис. 3: Пример гистограммного метода

3.2.3. Метод выделения краёв

Выделение краёв — это хорошо изученная область в обработке изображений. Границы и края областей сильно связаны, так как часто существует сильный перепад яркости на границах областей. Поэтому методы выделения краёв используются как основа для другого метода сегментации.

Обнаруженные края часто бывают разорванными. Но чтобы выделить объект на изображении, нужны замкнутые границы области.

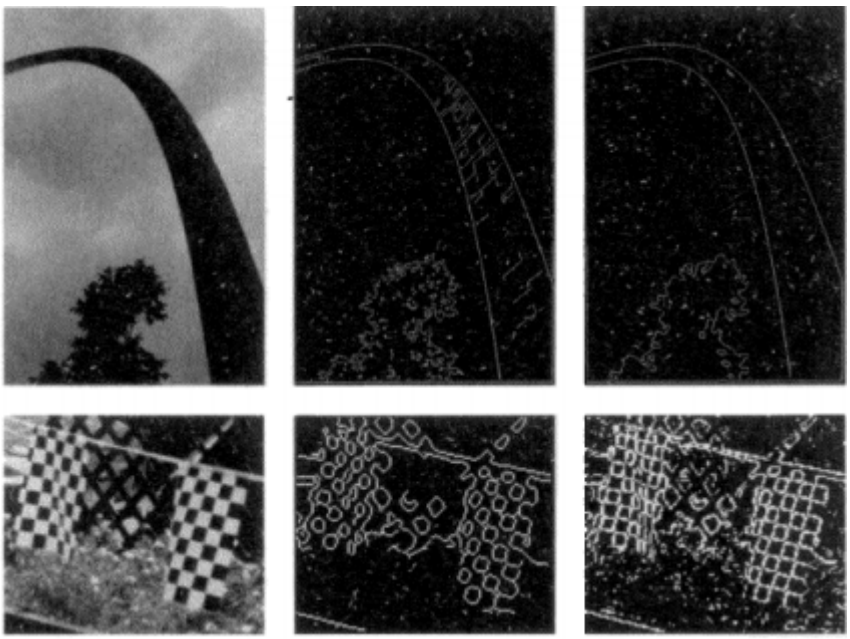

Рис. 4: Пример выделения краёв

3.2.4. Методы разрастания областей

Первым был метод разрастания областей из семян. В качестве входных данных этот метод принимает изображения и набор семян. Семена отмечают объекты, которые нужно выделить. Области постепенно разрастаются, сравнивая все незанятые соседние пиксели с областью. Разность δ между яркостью пикселя и средней яркостью области используется как мера схожести. Пиксель с наименьшей такой разностью добавляется в соответствующую область. Процесс продолжается, пока все пиксели не будут добавлены в один из регионов.

Метод разрастания областей из семян требует дополнительного ввода. Результат сегментации зависит от выбора семян. Шум на изображении может вызвать то, что семена плохо размещены. Метод разрастания областей без использования семян — это изменённый алгоритм, который не требует явных семян. Он начинает с одной области A_1 — пиксель, выбранный здесь, незначительно влияет на конечную сегментацию. На каждой итерации он рассматривает соседние пиксели так же, как метод разрастания областей с использованием семян. Но он отличается тем, что если минимальная δ меньше, чем заданный порог *T*, то он добавляется в соответствующую область A_j. В

противном случае пиксель считается сильно отличающимся от всех текущих областей A_i и создаётся новая область A_{n+1}, содержащая этот пиксель.

Один из вариантов этого метода, предложенный Хараликом и Шапиро области, и яркость пикселя-кандидата используется для построения тестовой статистики. Если тестовая статистика достаточно мала, то пиксель добавляется к области, и среднее и дисперсия области пересчитываются. Иначе пиксель игнорируется и используется для создания новой области.

Рис. 5: Разрастания областей

3.2.5. Метод водораздела

В сегментации методом водораздела рассматривается абсолютная величина градиента изображения как топографической поверхности. Пиксели, имеющие наибольшую абсолютную величину градиента яркости, соответствуют линиям водораздела, которые представляют границы областей. Вода, помещённая на любой пиксель внутри общей линии водораздела, течёт вниз к общему локальному минимуму яркости. Пиксели, от которых вода стекается к общему минимуму, образуют водосбор, который представляет сегмент.

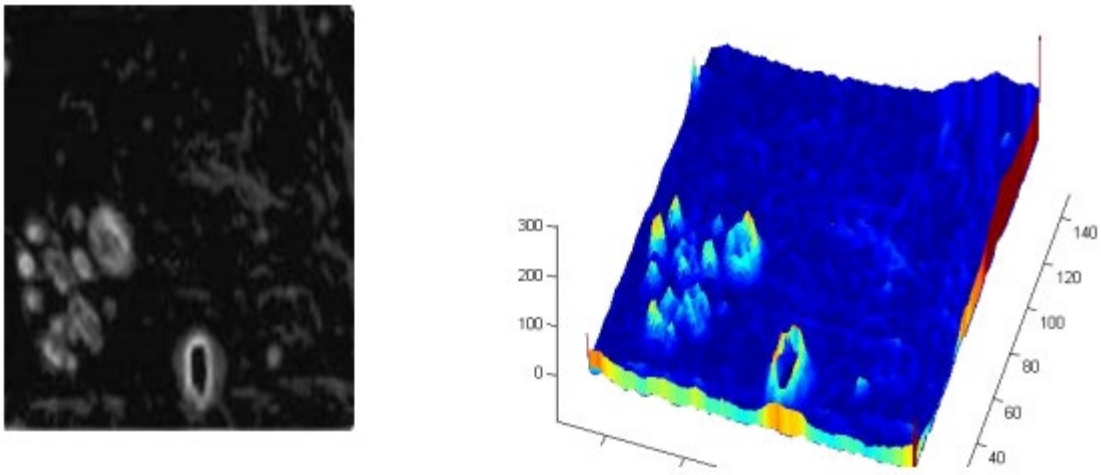

Рис. 6: Пример метода водораздела

3.2.6. Сегментация с помощью модели

Основное предположение этого подхода — то, что интересующие структуры или органы имеют повторяющиеся геометрические формы. Следовательно, можно найти вероятностную модель для объяснения изменений формы органа и затем, сегментируя изображение, накладывать ограничения, используя эту модель как априорную. Такое задание включает в себя (i) приведение тренировочных примеров к общей позе, (ii) вероятностное представление изменений, приведённых образцов и (iii) статистический вывод для модели и изображения. Современные методы в литературе для сегментации, основанной на знании, содержат активные модели формы и внешности, активные контуры, деформируемые шаблоны и методы установления уровня.

4. ПРАКТИЧЕСКИЙ РАЗДЕЛ

4.1 Средства разработки

Для разработки алгоритма сегментации изображения был использован IDE для разработки на C, C++ и QML. Разработана Trolltech (Digia) для работы с фреймворком Qt. Включает в себя графический интерфейс отладчика и визуальные средства разработки интерфейса как с использованием QtWidgets, так и QML.

Так как QtCreator поддерживает язык C++, алгоритм сегментации изображения был написан именно на нём.

7. ПРИЛОЖЕНИЕ

7.1 Код программы

Файл mainwindow.h

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QImage>
#include <QGraphicsScene>
#include <QPixmap>

namespace Ui {
class MainWindow;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT

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

private slots:

    void on_btLoad_clicked();

    void on_btProcess_clicked();

private:
    Ui::MainWindow *ui;

    // исходное изображение и сцена для размещения
    QImage *image;
    QGraphicsScene *scene;

    // результат (обработанное изображение)
    QImage *result_image;
    QGraphicsScene *result_scene;
};
```

#endif // MAINWINDOW_H

Файл main.cpp

```
#include "mainwindow.h"
#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();

    return a.exec();
}
```

Файл mainwindow.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

#include <QFileDialog>
#include <QDebug>
#include <QtMath>

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

    // создаем сцену и настраиваем просмотрщик (визуальное отображение сцены)
    scene = new QGraphicsScene(this);
    ui->viewer->setScene(scene);

    result_scene = new QGraphicsScene(this);
    ui->result_viewer->setScene(result_scene);
}

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


void MainWindow::on_btLoad_clicked()
{
    // Диалог выбора изображения для обработки
    QString file_path = QFileDialog::getOpenFileName(nullptr, "Select image",
"", "*.jpeg *.jpg *.png *.bmp");

    // Загрузка изображения
    image = new QImage();
    image->load(file_path);

    // Добавление изображения на сцену
    scene->clear();
    scene->addPixmap(QPixmap::fromImage(*image));
}

void MainWindow::on_btProcess_clicked()
{
    // создаем новое изображение
    result_image = new QImage(*image);

    // im - указатель для работы с изображением (для удобства)
    QImage *im = result_image;

    // определяем минимальную и максимальную интенсивность цвета пикселей
    qreal min_e = 255, max_e = 0;
    for(int x = 0; x<im->width(); x++)
    {
        for(int y=0;y<im->height();y++)
        {
            QColor color = im->pixelColor(x, y);
            int r = color.red();
            int g = color.green();
            int b = color.blue();

            qreal e = qSqrt((r*r + g*g + b*b)/3);
```

```
            if(e<min_e)
                min_e = e;
            if(e>max_e)
                max_e = e;
        }
    }

    // разбиваем изображение по интенсивности цвета на 3 группы,
    // которые закрашиваем красным (темные участки), зеленым (средние) и
синим цветом (яркие)
    for(int x = 0; x<im->width(); x++)
    {
        for(int y=0;y<im->height();y++)
        {
            // определяем цвет пикселя с координатами (x,y)
            QColor color = im->pixelColor(x, y);

            // получаем все три компоненты цвета
            int r = color.red();
            int g = color.green();
            int b = color.blue();

            qreal e = qSqrt((r*r + g*g + b*b)/3);
            qreal p1 = (min_e + max_e)/3;
            qreal p2 = 2*(min_e + max_e)/3;

            // перекрашиваем в зависимости от яркости пиксела
            if(e<p1) {
                im->setPixelColor(x,y, QColor(100, 0, 0));
            }
            else if(e>p2) {
                im->setPixelColor(x,y, QColor(0, 100, 0));
            }
            else {
                im->setPixelColor(x,y, QColor(0, 0, 100));
            }
        }
    }

    // Добавление изображения на сцену
    result_scene->clear();
    result_scene->addPixmap(QPixmap::fromImage(*result_image));
}
```

7.2 Результаты сегментации изображения

Рис. 7: Результат сегментации

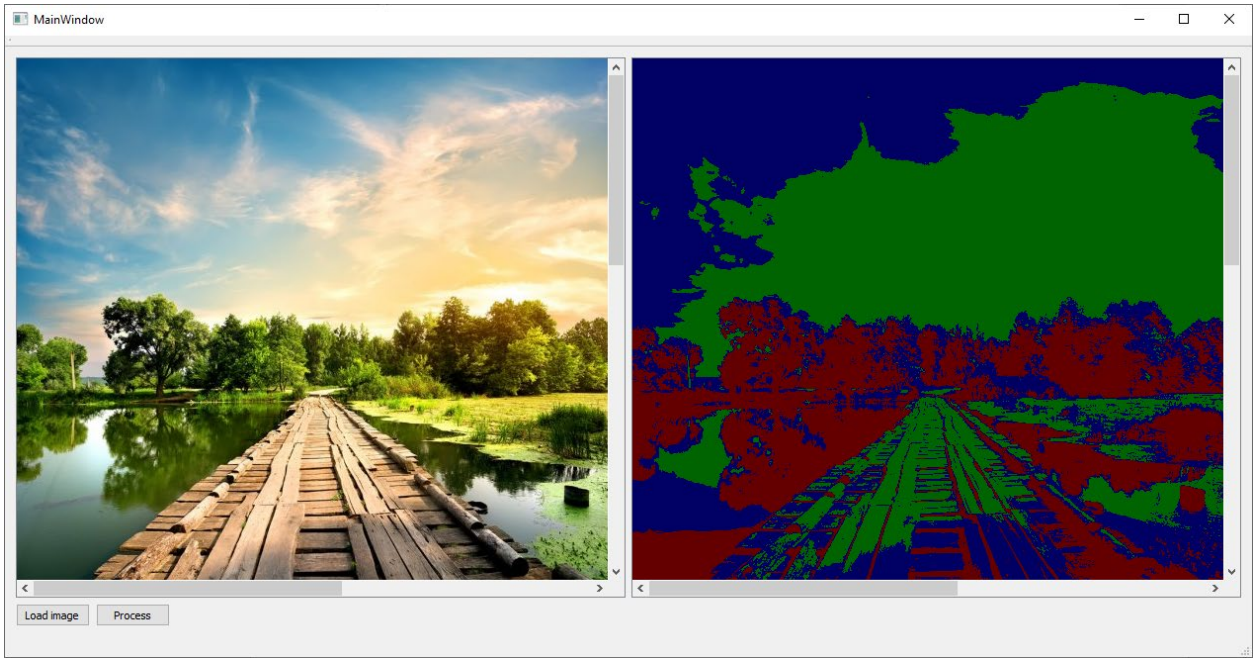

Рис. 8: Результат сегментации

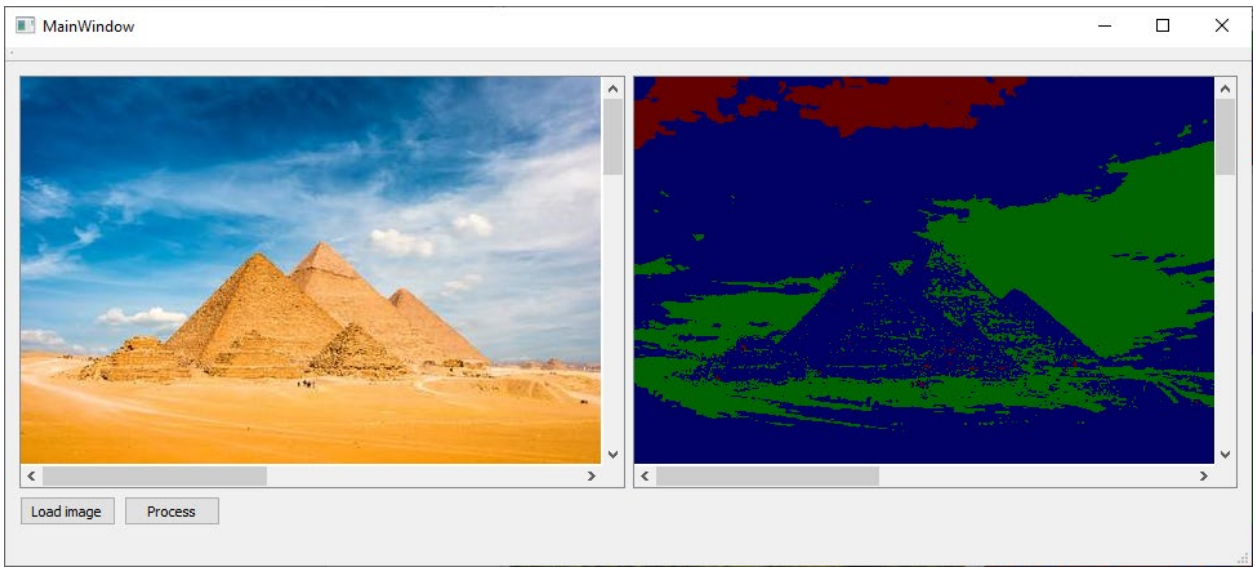

Рис. 9: Результат сегментации

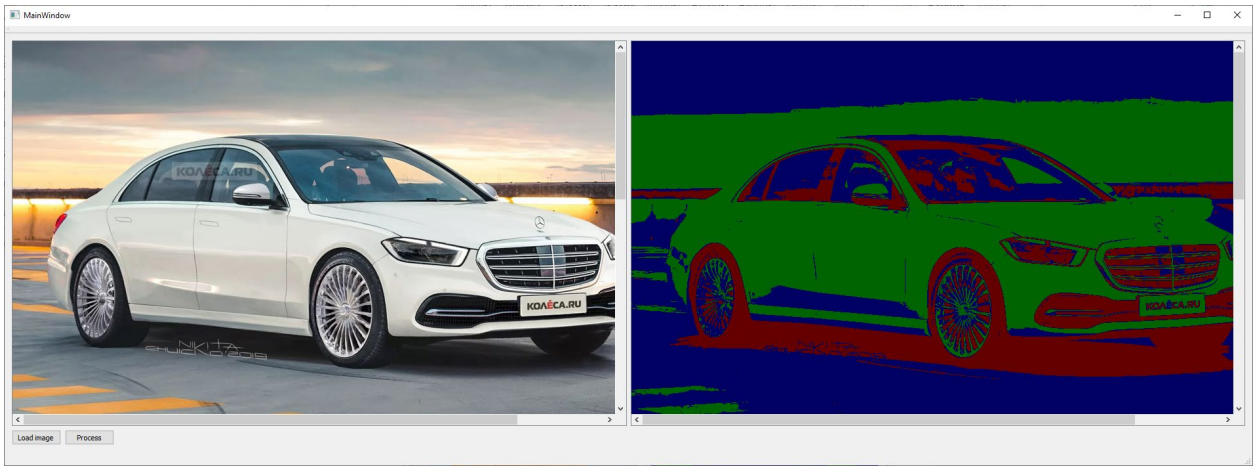

Рис. 10: Результат сегментации