

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий  
Кафедра вычислительной техники

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ О. В. Непомнящий  
подпись    инициалы, фамилия  
« \_\_\_\_ » \_\_\_\_\_ 2019 г.

## БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника

Разработка программного продукта для автоматической обработки  
портретного изображения.

Руководитель	_____	доцент, канд. техн. наук	<u>П. П. Дьячук</u>
	подпись, дата	должность, учёная степень	инициалы, фамилия
Выпускник	_____	_____	<u>А. А. Клюквин</u>
	подпись, дата		инициалы, фамилия
Нормоконтролер	_____	доцент, канд. техн. наук	<u>В. И. Иванов</u>
	подпись, дата	должность, учёная степень	инициалы, фамилия

Красноярск 2019

## РЕФЕРАТ

Выпускная квалификационная работа на тему «Разработка программного продукта для автоматической обработки портретного изображения» содержит 49 страниц текстового документа, 36 иллюстраций, 17 источников и 1 приложение.

ОБРАБОТКА ПОРТРЕТНОГО ИЗОБРАЖЕНИЯ, ПОИСК ПОРТРЕТНОГО ИЗОБРАЖЕНИЯ, БИНАРИЗАЦИЯ ИЗОБРАЖЕНИЯ, КАСКАДЫ ХААРА, СОЗДАНИЕ ПРОГРАММНОГО ПРОДУКТА

Целью выпускной квалификационной работы является разработка программного продукта для автоматической обработки портретного изображения.

В ходе работы были решены следующие задачи:

1. Изучена предметная область.
2. Выполнен анализ предметной области и методов распознавания изображений по теме работы.
3. Изучены существующие программы, приложения и методы автоматической обработки портретного изображения.
4. Разработан программный продукт.
5. Произведена апробация программного продукта.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1. Анализ предметной области и методов распознавания изображений по теме работы.....	5
1.1 Актуальность автоматической обработки портретного изображения .....	5
1.2 Обзор существующих программ и приложений для автоматической обработки портретного изображения .....	8
1.3 Бинаризация изображения .....	10
1.4 Нахождение контуров и операции с ними .....	12
1.5 Метод Виолы-Джонса (каскады Хаара) .....	14
1.6 Метод свертки изображения.....	19
1.7 Оператор Собеля.....	21
2 Разработка программного продукта.....	23
2.1 Обзор программного продукта .....	23
2.2 Поиск портрета на изображении .....	25
2.3 Удаление эффекта «красных глаз» на изображении.....	28
2.4 Удаление морщин на изображении .....	30
2.5 Удаление родинок на лице.....	32
2.6 Изменение цвета кожи на изображении .....	34
3 Апробация программного продукта.....	37
ЗАКЛЮЧЕНИЕ .....	42
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	43
ПРИЛОЖЕНИЕ А .....	46

## ВВЕДЕНИЕ

В настоящее время всё чаще приходится сталкиваться с необходимостью автоматической обработки изображения. Задача обработки изображения, требуется как в профессиональных целях человеческой деятельности, (например, системах мониторинга Земли, распознавания летательных аппаратов, медицинской диагностики, оценки качества продукции), так и в повседневных (например, при редактировании изображения, обработке старых фотографий и т.д).

При проведении анализа обработки изображения в различных технических системах наиболее значимо то, какие математические модели применяются для обработки получаемых данных. По мере усложнения математических моделей необходимо стремиться к повышению вычислительной мощности технических средств. В связи с этим необходимо обеспечивать построение эффективных методов обработки, и передачи больших объемов информации, связанных с изображениями.

При анализе задач, необходимо определить адекватную модель наблюдения. Практика показывает, что в настоящее время не существует универсального способа решения задач для достаточно широкого диапазона характеристик. Поэтому исследователям приходится находить новые модели или исследовать эффективность применения уже существующих.

Изображения могут подвергаться воздействию различных методов обработки. Отметим некоторые из них:

- Подбор контраста изображения.
- Сглаживание по наиболее неоднородной окрестности центральной точки.
- Аппроксимации характеристик изображения. При этом целое изображение разбивается на отдельные части, в каждой из которых проводится аппроксимация полиномом, например, на основе метода наименьших квадратов.
- Применение комбинированных методов обработки изображений.

В данной дипломной работе для обработки изображений применяется пиксельный анализ.

Пиксели, образующие изображение, с соответствующими им значениями интенсивностей организованы в массив, размерность которого определяется природой данных. Упорядочение пикселей в этом массиве соответствует пространственной организации сцены. Для многих задач массив пикселей является двумерным.

Представления на пиксельном уровне являются исходными для любых приложений интерпретации изображений с помощью цифровых вычислительных машин. Такое представление изображения является общим для различных задач компьютерной обработки изображений.

# **1 Анализ предметной области и методов распознавания изображений по теме работы**

## **1.1 Актуальность автоматической обработки портретного изображения**

Обработка изображения применяется во множестве сфер человеческой деятельности, отметим некоторые из них.

**Обработка портрета в социальных сетях.** Например, Facebook анализирует фотографии, на которых находится человек или группа лиц и с высокой степенью вероятности опознает человека на них.

Многие ИТ-компании, включая Microsoft, уже несколько лет используют такие технологии, чтобы превращать утомительную работу по каталогизации фотографий в более быструю и результативную [1].

**Обработка портретов в сфере безопасности.** Простой пример использования технологии распознавания лиц для защиты личных данных — разблокировка смартфона «по лицу». Такую технологию можно внедрить и в пропускную систему: человек смотрит в камеру, а она определяет разрешить ему войти или нет.

**Обработка портрета для подсчета количества людей.** Технологию распознавания портретного можно использовать при подсчете количества людей, посещающих какое-либо мероприятие (например, конференцию или концерт). Вместо того чтобы вручную подсчитывать участников, устанавливается камера, которая может захватывать изображения лиц участников и выдавать общее количество посетителей. Это поможет автоматизировать процесс и сэкономить время.

Актуальность темы «Разработка программного продукта для автоматизации обработки портретного изображения» заключается в том, что разработанный программный продукт, позволяет обрабатывать портретное

изображение на основе пиксельного анализа фотоснимков что позволит сократить временные рамки получения результатов, сократить материальные затраты, увеличить скорость выполнения обработки данных, а так же широкие возможности для редактирования данных [12].

Объект исследования – технологии обработки, улучшения фотоснимков и анализ сцен, содержащих детали, подлежащие распознаванию.

Предмет исследования – анализ частей портретного изображения на основе технологий обработки, улучшения и редактирования фотоснимков и анализ сцен, содержащих детали, подлежащие распознаванию.

Целью дипломной работы является выявление проблем анализа портретного изображения и разработка информационной системы для обработки, редактирования и улучшения характеристик изображения.

В рамках проекта решаются следующие задачи:

- анализ предметной области по теме дипломной работы;
- анализ методов распознавания изображений;
- разработка программного продукта;
- апробация программного продукта на предоставленных фотоснимках;

Главная проблема распознавания портретного изображения заключается в том, что имеющиеся аппаратные комплексы не могут с требуемой точностью распознавать лица на изображении, а параметры редактирования шаблонов не используются, либо имеют узкий функционал. Для лучшего понимания проблемы распознавания портретного изображения рассмотрим дерево проблем и дерево целей изображенные на рисунке 2,3.

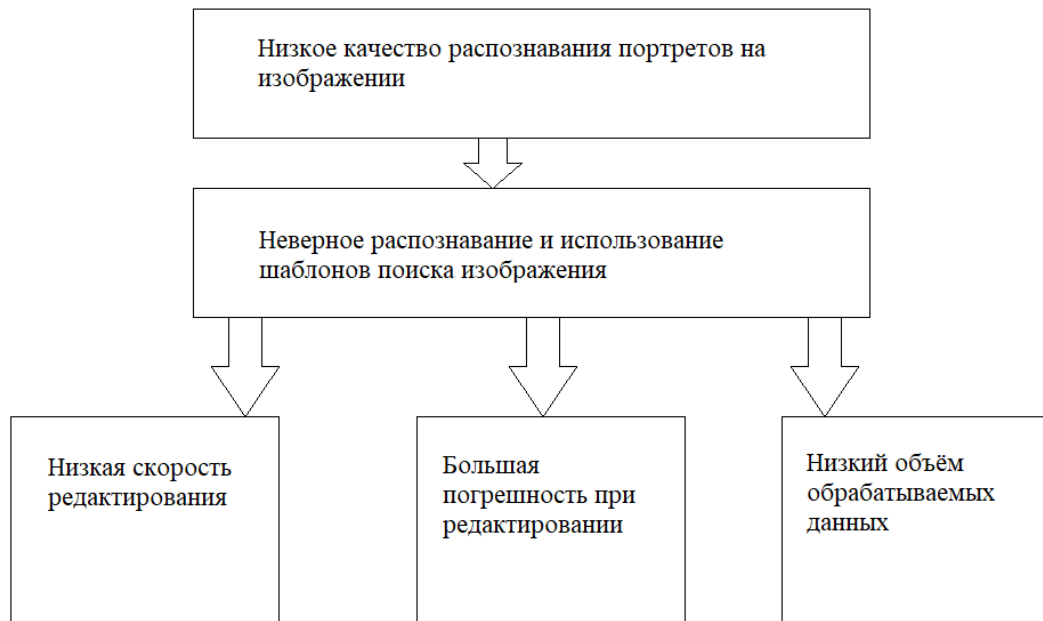


Рисунок 1 – Дерево проблем для распознавания портретного изображения



Рисунок 2 – Дерево целей для улучшения качества распознавания портретного изображения



## **1.2 Обзор существующих программ и приложений для автоматической обработки портретного изображения**

В настоящее время для обработки портретного изображения существует большое количество программных комплексов и приложений, таких как: AdobePhotoshop, Movaviphotoeditor, photomaster, GIMP, paint, ACDSee, Picasa и т.д. Помимо программных комплексов и приложений существуют так же веб сервисы позволяющие изменять изображения без использования специального ПО.

Отметим недостатки, которые присущи данным программным комплексам:

1. Требуют платные лицензии (AdobePhotoshop, Movaviphotoeditor, ACDSee).

2. Имеют ограниченное количество функций во время бесплатного использования. Пробная версия MovaviPhotoEditor не поддерживает сохранение файла и не позволяет сделать скриншот экрана всеми известными способами.

3. Имеют высокие системные требования (AdobePhotoshop, Movaviphotoeditor, ACDSee).

4. Не предназначены для автоматической обработки изображения. Некоторые функции, автоматически реализуемые в проекте дипломной работы, в большинстве программ, таких как: Movaviphotoeditor, photomaster, GIMP, paint, ACDSee, Picasa, могут выполняться только вручную.

5. Требуют специальных навыков для правильного использования. Для успешной работы с такими программами как Photoshop, GIMP, необходимо обучение.

6. Имеют узкий функционал возможностей, который не позволяет использовать все методы обработки изображения (Paint, photomaster, Movaviphotoeditor).

Программный продукт имеет свои преимущества и недостатки, отметим некоторые из них.

Преимущества:

1. Бесплатное ПО
2. Не требует использования сторонних библиотек и сторонних программ для работы с изображением.
3. Для работы программы не требуется большого объема вычислительных мощностей, таких как в photoshop, gimp, ACDSееи т.д.
4. Имеет возможность автоматической обработки изображения.
5. Программа проста в использовании и не требует обучения для работы.
6. Возможность использовать подключаемые модули и функции реализуя возможность дальнейшего дополнения и расширения функционала.

Недостатки:

1. Отсутствует возможность обработки векторного изображения (Photoshop, GIMP).
2. Невозможность запуска приложения на мобильных устройствах и планшетах (Photoshop, GIMP).
3. Невозможность интеграции с облачными сервисами (photoshop, picasa).

### 1.3 Бинаризация изображения

Во многих программах для обработки изображения сначала идёт процесс бинаризации.

Главная цель бинаризации, это радикальное уменьшение количества информации, с которой приходится работать. Просто говоря, удачная бинаризация сильно упрощает последующую работу с изображением. С другой стороны, неудачи в процессе бинаризации могут привести к искажениям, таким, как разрывы в линиях, потеря значащих деталей, нарушение целостности объектов, появление шума и непредсказуемое искажение символов из-за неоднородностей фона.

Процесс бинаризации – это перевод цветного (или в градациях серого) изображения в двухцветное черно-белое. Главным параметром такого преобразования является порог  $t$  – значение, с которым сравнивается яркость каждого пикселя. По результатам сравнения, пикселю присваивается значение 0 или 1. Существуют различные методы бинаризации, которые можно условно разделить на две группы – глобальные и локальные. В первом случае величина порога остается неизменной в течение всего процесса бинаризации. Во втором изображение разбивается на области, в каждой из которых вычисляется локальный порог.

Различные методы бинаризации имеют свои слабые места: так, например, метод Оцу предполагает наличие в изображении двух классов пикселей(текстовые и фоновые) и ищет оптимальный порог, разделяющий эти два класса так, чтобы их внутриклассовая дисперсия была минимальна. Выбор порога в этом методе основан на минимизации внутригрупповой дисперсии двух групп пикселей, разделяемых оператором пороговой бинаризации. При рассмотрении метода будем считать, что гистограмма задана в виде дискретной функции распределения вероятности  $P$ . Значения  $P(0), \dots, P(I)$  представляет собой наблюдаемые вероятности значений яркости  $0, \dots, I$ ;  $P(i) =$

$\frac{|\{(r,c)|\text{Image}(r,c) = i\}|}{|R * C|}$ , где  $R * C$  равно площади изображения. При проектировании алгоритмов автоматического определения порога бинаризации обычно делаются некоторые предположения относительно формы распределения значения яркости пикселей изображения. В методе Оцу распределение значений яркости считается бимодальным [15].

Ключевым слабым местом этого метода является то, что он может приводить к утрате мелких деталей и «слипанию» близлежащих символов, а также появлением ложных объектов в случае неоднородностей фона с низкой контрастностью. Отсюда следует, что каждый метод должен быть применен в своей области.

В обработке изображений часто используется процедура «пороговой бинаризации» - разбиение изображения на две области, одна из которых содержит все пиксели со значением ниже некоторого порога, а другая содержит все пиксели со значением выше этого порога.

Алгоритмы адаптивной бинаризации изображений базируются на использовании локальной или глобальной гистограммы изображения. Подход, основанный на использовании гистограммы яркостей пикселей, является одним из самых известных и, безусловно, наиболее простым [10].

Оптимальная пороговая сегментация основана на приближении гистограммы изображения к некоторой кривой с использованием весовых сумм двух или более вероятностей интенсивности с нормальным распределением. Тогда порог - это набор ближайших уровней яркости, соответствующих минимуму вероятности между максимумами двух или более нормальных распределений.

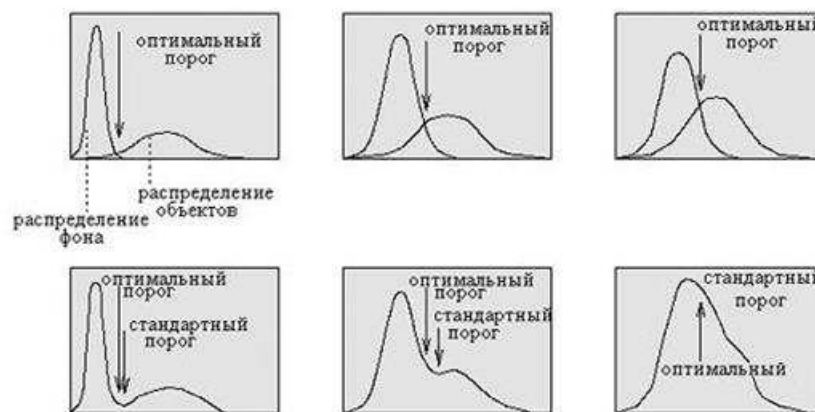


Рисунок 3 – Пороги бинаризации изображения

#### 1.4 Нахождение контуров и операции с ними

Контурный анализ позволяет описывать, хранить, сравнивать и производить поиск объектов, представленных в виде своих внешних очертаний – контуров. В данном дипломном проекте для распознавания и нахождения лиц используется метод контурного анализа.

Предполагается, что контур содержит всю необходимую информацию о форме объекта. Внутренние точки объекта во внимание не принимаются. Это ограничивает область применимости алгоритмов контурного анализа, но рассмотрение только контуров позволяет перейти от двумерного пространства изображения – к пространству контуров, и тем самым снизить вычислительную и алгоритмическую сложность. Контурный анализ позволяет эффективно решать основные проблемы распознавания образов – перенос, поворот и изменение масштаба изображения объекта [13].

Методы контурного анализа инвариантны к этим преобразованиям. Контур – это граница объекта, совокупность точек (пикселей), отделяющих объект от фона. В системах компьютерного зрения используется несколько способов кодирования контура – наиболее известны код Фримена, двумерное

кодирование, полигональное кодирование. Но все эти способы кодирования не используются в контурном анализе. Вместо этого, в контурном анализе контур кодируется последовательностью, состоящей из комплексных чисел. На контуре фиксируется точка, которая называется начальной точкой. Затем, контур обходится (допустим – по часовой стрелке), и каждый вектор смещения записывается комплексным числом  $a+ib$ . Где  $a$  – смещение точки по оси  $X$ , а  $b$  – смещение по оси  $Y$ . Смещение берется относительно предыдущей точки.

В силу физической природы трехмерных объектов, их контуры всегда замкнуты и не могут иметь самопересечения. Это позволяет однозначно определить путь обхода контура (с точностью до направления – по или против часовой стрелки). Последний вектор контура всегда приводит к начальной точке. Каждый вектор контура будем называть элементарным вектором (ЭВ). А саму последовательность комплексных чисел – вектор-контуром. Вектор-контуры будем обозначать большими греческими буквами, а их элементарные вектора – малыми греческими буквами. Таким образом, вектор-контур  $\Gamma$  длины  $k$  можно обозначить как:  $\Gamma = (y_0, y_1, \dots, y_{k-1})$

В контурном анализе используется именно комплексное кодирование потому, что операции над контуром именно как над вектором комплексных чисел обладает замечательными математическими свойствами, по сравнению с другими способами кодирования [4]. Комплексное кодирование близко к двумерному кодированию, где контур определяется как совокупность ЭВ, представленных своими двумерными координатами. Но разница в том, что операция скалярного произведения для векторов и для комплексных чисел – различны. Именно это обстоятельство и дает преимущество методам контурного анализа. Поиск контуров осуществляется с помощью функции `FindContours()`.

Пример кода:

```
import numpy as np
import cv2
im = cv2.imread('test.jpg')
```

```
imgray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
ret, thresh = cv2.threshold(imgray, 127, 255, 0)
im2, contours, hierarchy= cv2.findContours(thresh, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
```

Есть три аргумента в функции `cv2.findContours()`, первая – исходное изображение, второе – режим поиска контуров, третий – метод приближения контуров, который выводит измененное изображение, контуры и иерархию. Каждый отдельный контур представляет собой массив Numpy (x,y) координат граничных точек объекта.

Для рисования контуров используется функция `DrawContours`. Её также можно использовать для рисования любой формы, если у вас есть ее граничные точки. Её первый аргумент – исходное изображение, второй аргумент – контуры, которые должны быть переданы как список. Третий аргумент – индекс контуров, а остальные аргументы – цвет, толщина и т.п.

Чтобы нарисовать все контуры изображения:

```
cv2.drawContours(img, contours, -1,(0,255,0),3).
```

Чтобы нарисовать индивидуальный контур:

```
cv2.drawContours(img, contours, 3,(0,255,0),3).
```

Но чаще всего используется этот метод:

```
cnt = contours[4] cv2.drawContours(img, [cnt], -0,(0,255,0),3).
```

Третий аргумент сохраняет координаты (x,y) границы формы. Если будет пройден `cv2.CHAIN_APPROX_NONE`, все граничные точки будут сохранены.

## **1.5 Метод Виолы-Джонса (каскады Хаара)**

После нахождения контура портретного изображения идёт последующее нахождение составных частей лица, таких как, нос, лоб, глаза, губы и т.д.

Для нахождения составных частей используются шаблоны (они же примитивы Хаара). В дальнейшем работа с такими шаблонами позволит выполнять редактирование изображения, реализуя такие функции как Удаление черных точек на лице, Удаление красных глаз на лице, Изменение форм лица таких как: губы, брови, скулы, лоб, уши, нос, глаза, Изменение цвета кожи.

Метод Виола и Джонса использует технологию скользящего окна. То есть рамка, размером, меньшим, чем исходное изображение, двигается с некоторым шагом по изображению, и с помощью каскада слабых классификаторов определяет, есть ли в рассматриваемом окне лицо. Метод скользящего окна эффективно используется в различных задачах компьютерного зрения и распознавания объектов.

Цифровое изображение представляется в виде двумерной матрицы, где элемент матрицы представляет собой значение пикселя на отрезке [0;255]. Если изображение – черно-белое, то тогда изображение можно описать одной матрицей, если цветное, то изображение описывается 3 двумерными матрицами для каждого из цветов RGB (R – красный, G – зеленый, B – синий). Существуют и другие представления цветного изображения.

Одним из основных принципов метода Виолы-Джонса является представление черно-белого изображения или в оттенках градациях серого в интегральном виде [6]. Интегральное представление позволяет посчитать суммарную яркость произвольного прямоугольника на изображении. Интегральное изображение по размерам совпадает с исходным. Для подсчета используется формула на рисунке 4.

$$I(x,y) = \sum_{i=0, j=0}^{i \leq x, j \leq y} I(i,j)$$

Рисунок 4 - Формула для получения интегрального изображения

Значение в точке X, Y матрицы [9], полученной из исходного изображения [\*Гонсалес P] это сумма всех точек в прямоугольнике (0, 0, X, Y). Тогда интеграл по любому прямоугольнику (ABCD) в изображении представим



как:  $\text{SumOfRect}(ABCD) = \Pi(A) + \Pi(C) - \Pi(B) - \Pi(D)$ . Таким образом, получили быструю и удобную формулу для вычисления сумм яркостей пикселей в прямоугольнике.

Используя интегральное изображение, можно очень легко и быстро вычислять признаки Хаара. В алгоритме Виолы-Джонса используются прямоугольные признаки или примитивы Хаара, изображенные на рисунке 5.

Признак Хаара является набором прямоугольных областей изображения, примыкающих друг к другу и разделенных на две группы. Возможных признаков Хаара огромное множество (разнообразные комбинации областей разной ширины и высоты с разными позициями на изображении).



Рисунок 5 - Пример шаблонов Хаара

Первоначальный набор признаков зависит от реализации конкретной задачи. Чтобы вычислить значение конкретного признака Хаара для какого-либо изображения, надо сложить яркости пикселей изображения в первой и второй группах прямоугольных областей по отдельности, а затем вычесть из первой полученной суммы вторую. Полученная разность и есть значение конкретного признака Хаара для данного изображения.

Если шаблоны соответствуют конкретным областям на изображении, то считается, что на изображении есть человеческое лицо. Для каждого из множества шаблонов считается разность между яркостью белой и черной областей. Это значение сравнивается с эталоном, после этого принимается решение о том, есть ли здесь часть человеческого лица или нет.

Если применить шаблоны ко всему изображению мы не найдем там лиц, т.к. они будут меньше шаблонов. Для того чтобы искать на всем фото лица

разных размеров используется метод скользящего окна. Именно внутри этого окна и высчитываются примитивы. Окно проходит по всему изображению. После каждого прохождения изображения окно уменьшается, чтобы найти лица меньшего масштаба.

Используя данный метод можно легко находить шаблоны на портретном изображении для их дальнейшего редактирования.

Недостатки метода:

- Неустойчивость при смене освещения.
- Неустойчивость при изменении масштаба или повороте изображения.
- Маленькая скорость работы — если нужно обнаружить область  $n \times n$  на изображении  $m \times m$ .

Методы устранения недостатков:

- Освещение нейтрализуется нормировкой или переходом к бинаризации области.
- Изменения масштаба и небольшие повороты нейтрализуются изменением разрешения при корреляции.
- Скорость оптимизируют путём поиска с большим шагом или при маленьком разрешении.

Преимущества метода:

- Устойчивость к смене освещения, даже если это локальная смена освещения, устойчивость к шумам (примитивы представляют собой простейший полосовой фильтр).
- Если примитивы были не очень маленькие, то сильно устойчивее корреляции при изменении масштаба (размер примитивов при этом не будет влиять на точность, если обход с маленьким шагом).
- Возможно обнаружение более одного лица на изображении.

Изначально алгоритм был предложен для распознавания только лиц, но его можно использовать для распознавания других объектов. Одним из вкладов Виолы и Джонса было применение таблицы сумм, которую они назвали

интегральным изображением. Обобщенная схема распознавания в алгоритме Виолы-Джонса показана на рисунке 6.

Обобщенная схема алгоритма выглядит следующим образом: перед началом распознавания алгоритм обучения на основе тестовых изображений обучает базу данных, состоящую из признаков, их паритета и границы. Далее алгоритм распознавания ищет объекты на разных масштабах изображения, используя созданную базу данных.

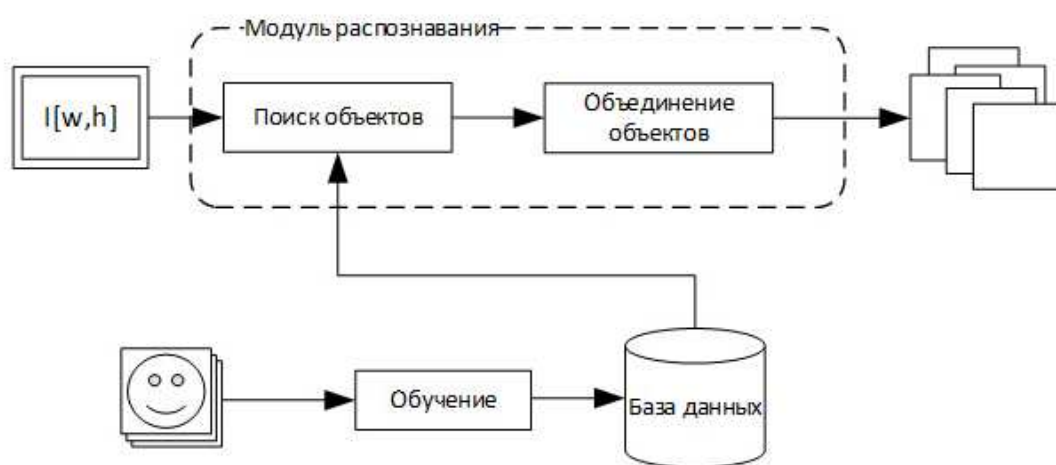


Рисунок 6 - Схема распознавания в алгоритме Виолы-Джонса

Алгоритм Виолы-Джонса на выходе дает всё множество найденных необъединенных объектов на разных масштабах. Следующая задача – принять решение о том, какие из найденных объектов действительно присутствуют в кадре, а какие – дубли.

Если признаки на большом изображении рассчитать заранее и при сдвиге окна поиска брать уже посчитанные и актуальные для него — поиск будет значительно быстрее корреляции (нужно сравнить меньшее количество элементов).

## 1.6 Метод свертки изображения

Свёртка (англ. convolution) — это операция, показывающая «схожесть» одной функции с отражённой и сдвинутой копией другой. В данном дипломном проекте свёртка используется для операций фильтрации, таких как: изменение тона лица, удаления эффекта красных глаз на изображении, удаление шумов на лице (чёрных точек, родинок, прыщей, ссадин и т.д).

В случае работы с портретным изображением — свёртка представляет собой операцию по вычислению нового значения заданного пикселя. При этой операции также учитывается значение окружающих его соседних пикселей [5].

Главным элементом свёртки является ядро свёртки — это матрица произвольного размера и отношения сторон. Чаще всего используется квадратная матрица с размерами 3x3 и 5x5.

У ядра свёртки есть важный параметр — структурирующий элемент. Это элемент матрицы (чаще всего — центр), который прикладывается к заданному пикселю изображения.

При вычислении нового значения выбранного пикселя изображения, структурирующий элемент свёртки прикладывается своим центром к этому пикселю. Соседние пиксели так же накрываются структурирующим элементом.

Далее, вычисляется сумма произведений значений пикселей изображения на значения, накрывшего данный пиксель центрального элемента. Полученная сумма и является новым значением выбранного пикселя. Таким образом, если применить свёртку к каждому пикселю изображения, то получится некий эффект, зависящий от выбранного структурирующего элемента свертки.

Для более полного понимания работы алгоритма рассмотрим пример: пусть у нас есть клеточное поле, которое соответствует пикселям исходного изображения:

[47]	[48]	[49]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
[47]	[50]	[42]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
[47]	[48]	[42]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]

Рисунок 7 - Клеточное поле пикселей исходного изображения

При этом создадим так же поле, являющее собой ядро свёртки:

[0]	[1]	[0]
[0]	[0]	[0]
[0]	[0]	[0]

Рисунок 8 - Ядро свёртки

Рассмотрим результат операции наложения ядра свёртки на пиксель со значением 50:

$$\text{результат} = 47*0 + 48*1 + 49*0 + 47*0 + 50*0 + 42*0 + 47*0 + 48*0 + 42*0 = 48$$

[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
[ ]	[48]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]

Рисунок 9 - Результат операции

## 1.7 Оператор Собеля

Оператор Собеля — это дискретный дифференциальный оператор, вычисляющий приближение градиента яркости изображения. Оператор вычисляет градиент яркости изображения в каждой точке. Так находится направление наибольшего увеличения яркости и величина её изменения в этом направлении. Результат показывает, насколько «резко» или «плавно» меняется яркость изображения в каждой точке, а значит, вероятность нахождения точки на грани, а также ориентацию границы.

Наиболее общим способом поиска разрывов является обработка изображения с помощью скользящей маски, называемой также фильтром, ядром, окном или шаблоном, которая представляет собой некую квадратную матрицу, соответствующую указанной группе пикселей исходного изображения. Элементы матрицы принято называть коэффициентами. Оперирование такой матрицей в каких-либо локальных преобразованиях называется фильтрацией или пространственной фильтрацией.

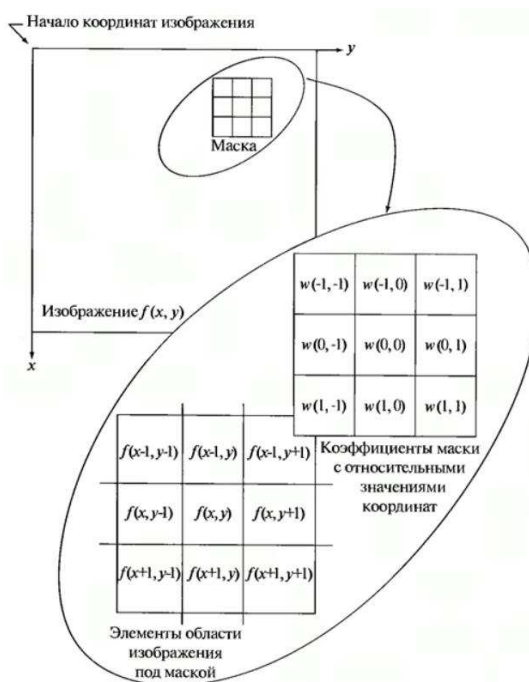


Рисунок 10 – Схема пространственной фильтрации

Результатом работы оператора Собеля в точке области постоянной яркости будет нулевой вектор, а в точке, лежащей на границе областей различной яркости — вектор, пересекающий границу в направлении увеличения яркости.

Оператор Собеля состоит из двух отдельных операций:

- Сглаживание треугольным фильтром в перпендикулярном к производной направлении:  $h(-1) = 1; h(0) = 2; h(1) = 1$ .
- Нахождение простого центрального изменения в направлении производной:  $h'(-1) = 1; h'(0) = 0; h'(1) = -1$ .

Наиболее часто оператор Собеля применяется в алгоритмах выделения границ. Оператор Собеля основан на свёртке изображения небольшими целочисленными фильтрами в вертикальном и горизонтальном направлениях, поэтому его относительно легко вычислять. Оператор использует ядра  $3 \times 3$ , с которыми свёртывают исходное изображение для вычисления приближенных значений производных по горизонтали и по вертикали.

$z_1$	$z_2$	$z_3$
$z_4$	$z_5$	$z_6$
$z_7$	$z_8$	$z_9$

Рисунок 11 – Окрестность  $3 \times 3$  изображения

Программная реализация оператора Собеля может эффективно использовать SIMD (single instruction, multiple data) - расширения системы команд современных процессоров (т. н. векторизация кода), при этом выигрыш в скорости вычисления оператора может составлять до пяти раз по сравнению с высокоуровневой реализацией.

## 2 Разработка программного продукта

### 2.1 Обзор программного продукта

Программный продукт реализован на языке программирования Python.

Python— высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.

Python поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное.

Основные архитектурные черты: динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных. Код в Python организовывается в функции и классы, которые могут объединяться в модули (они в свою очередь могут быть объединены в пакеты) [17].

Эталонной реализацией Python является интерпретатор CPython, поддерживающий большинство активно используемых платформ. Он распространяется под свободной лицензией PythonSoftwareFoundationLicense, позволяющей использовать его без ограничений в любых приложениях, включая проприетарные. Есть реализации интерпретаторов для JVM, MSIL (с возможностью компиляции), LLVM и другие. Проект PyPy предлагает реализацию Python с использованием JIT-компиляции, которая значительно увеличивает скорость выполнения Python-программ.

Python — активно развивающийся язык программирования, новые версии (с добавлением/изменением языковых свойств) выходят примерно раз в два с половиной года. Вследствие этого и некоторых других причин на Python



отсутствуют стандарт ANSI, ISO или другие официальные стандарты, их роль выполняет CPython.

Python поддерживает динамическую типизацию, то есть тип переменной определяется только во время исполнения. Поэтому вместо «присваивания значения переменной» лучше говорить о «связывании значения с некоторым именем». В Python имеются встроенные типы: булевый, строка, Unicode-строка, целое число произвольной точности, число с плавающей запятой, комплексное число и некоторые другие. Из коллекций в Python встроены: список, кортеж, словарь, множество и другие. Все значения являются объектами, в том числе функции, методы, модули, классы.

При разработке программного продукта использовалась также библиотека OpenCV.

OpenCV (OpenSourceComputerVision) — библиотека компьютерного зрения с открытым исходным кодом) — библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом.

Основные модули библиотеки:

**Sxcore** — ядро, содержит базовые структуры данных и алгоритмы:

- базовые операции над многомерными числовыми массивами;
- матричная алгебра, математические функции, генераторы случайных чисел;
- запись/восстановление структур данных в/из XML;
- базовые функции 2D графики.

**CV** — модуль обработки изображений и компьютерного зрения:

- базовые операции над изображениями (фильтрация, геометрические преобразования, преобразование цветовых пространств и т. д.);
- анализ изображений (выбор отличительных признаков, морфология, поиск контуров, гистограммы);
- анализ движения, слежение за объектами;
- обнаружение объектов, в частности лиц;

— калибровка камер, элементы восстановления пространственной структуры.

**Highgui** — модуль для ввода/вывода изображений и видео, создания пользовательского интерфейса:

— захват видео с камер и из видео файлов, чтение/запись статических изображений;

— функции для организации простого UI (все демо приложения используют HighGUI).

## 2.2 Поиск портрета на изображении

Для нахождения портрета на изображении используется алгоритм бинаризации, нахождения контуров, а так же каскады Харра. В методе бинаризации изображение разбивается на части и в каждой такой части каждый пиксель сравнивается с соседними 8 пикселями. Если значение центрального пикселя больше соседнего, ему присваивается значение 0, в противном случае 1. И так для каждого пикселя у нас получается некоторое число.

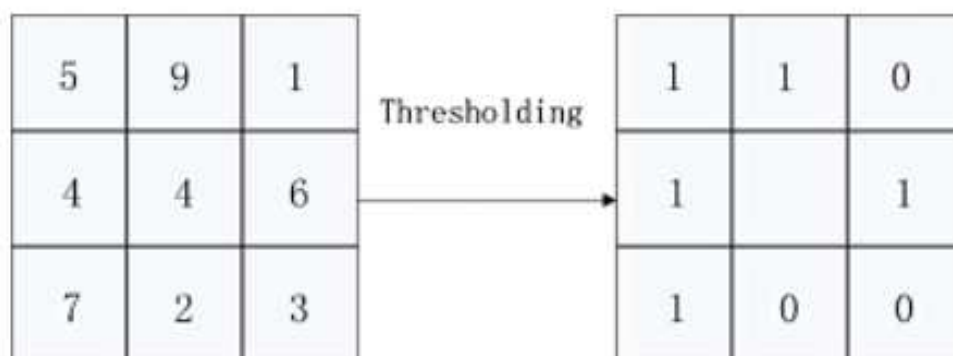


Рисунок 12 – Схема работы бинаризации для нахождения портрета на изображении

Далее на основе этих чисел для всех частей, на которые мы разбивали фотографию, считается гистограмма. Все гистограммы со всех частей объединяются в один вектор характеризующий изображение в целом.

После того как изображение подверглось бинаризации, запускается алгоритм каскадов Хаара для поиска шаблонов фрагментов лиц на изображении. Если шаблоны соответствуют конкретным областям на изображении, то считается, что на изображении есть человеческое лицо. Для каждого из множества шаблонов считается разность между яркостью белой и черной областей. Это значение сравнивается с эталоном, после этого принимается решение о том, есть ли здесь часть человеческого лица или нет.

Для того чтобы найти на фото лицо используется метод скользящего окна. Именно внутри этого окна и высчитываются примитивы. Окно проходит по всему изображению. После каждого прохождения изображения окно уменьшается, чтобы найти лица меньшего масштаба.

В конечном итоге для редактирования изображения выбирается окно, в котором шаблоны максимально приближены к эталонным.

Пример работающей функции:

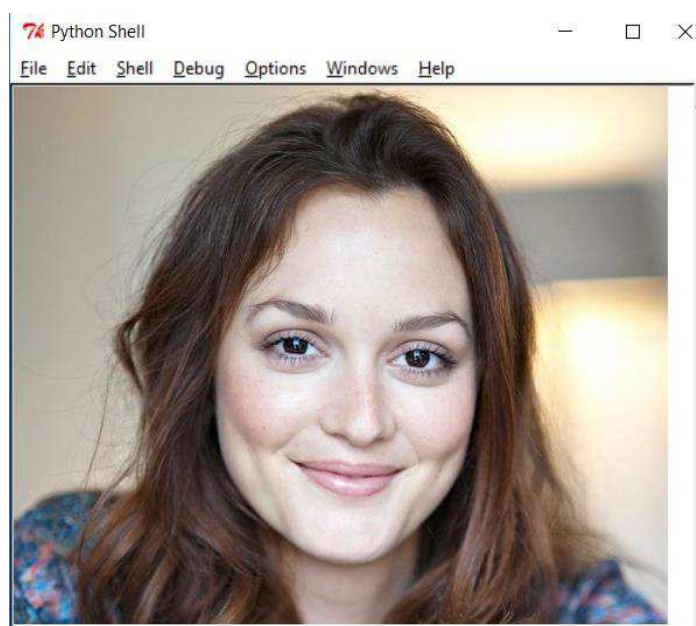


Рисунок 13 - Начальное изображение

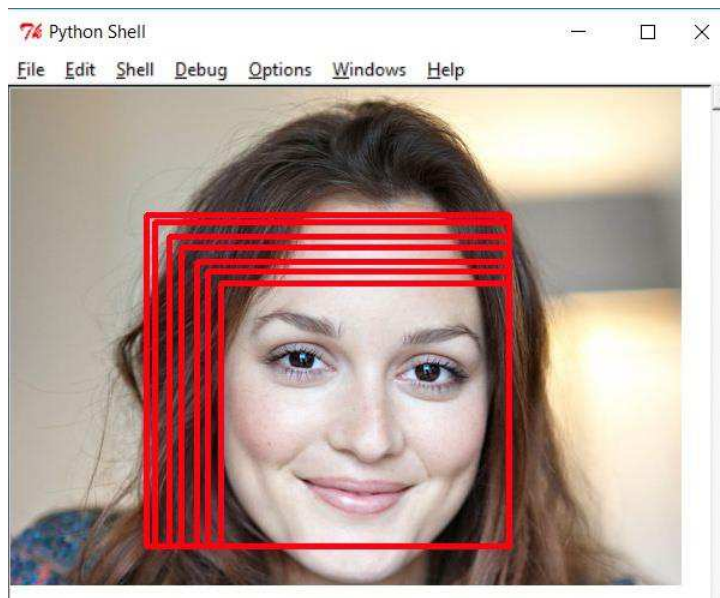


Рисунок 14 - Портрет, найденный на изображении

## 2.3 Удаление эффекта «красных глаз» на изображении

После того как на изображении было найдено лицо, для нахождения области «красных глаз» на изображении используется алгоритм каскадов Харра. Шаблон с областью содержащей «красные глаза» подвергается последующему редактированию, в основе которого используется метод свёртки.

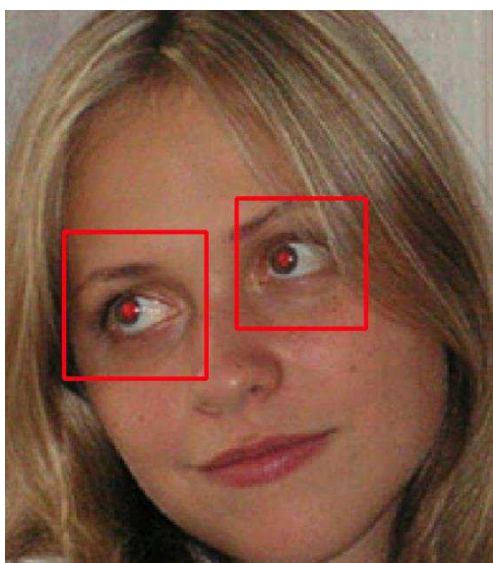


Рисунок 15 - Начальное изображение с найденными шаблонами каскадов Хаара

Далее нужно найти ту часть зрачка, на которой глаз имеет красный цвет. Цветное изображение разделяется на три канала, для этого существует специальная команда для разбивания цвета изображения на каналы - split. После этого создаётся маска, элементы которой равны 1 для каждого пикселя, где красный канал выше порога (150) и где красный канал больше, чем сумма зеленого и синего каналов. Это поможет найти области содержащие красный цвет на изображении для последующего редактирования шаблона.



Рисунок 16 – Полученное, в результате промежуточной обработки изображение

Как только нужная область изображения найдена, она подвергается обработке при помощи метода свёртки.

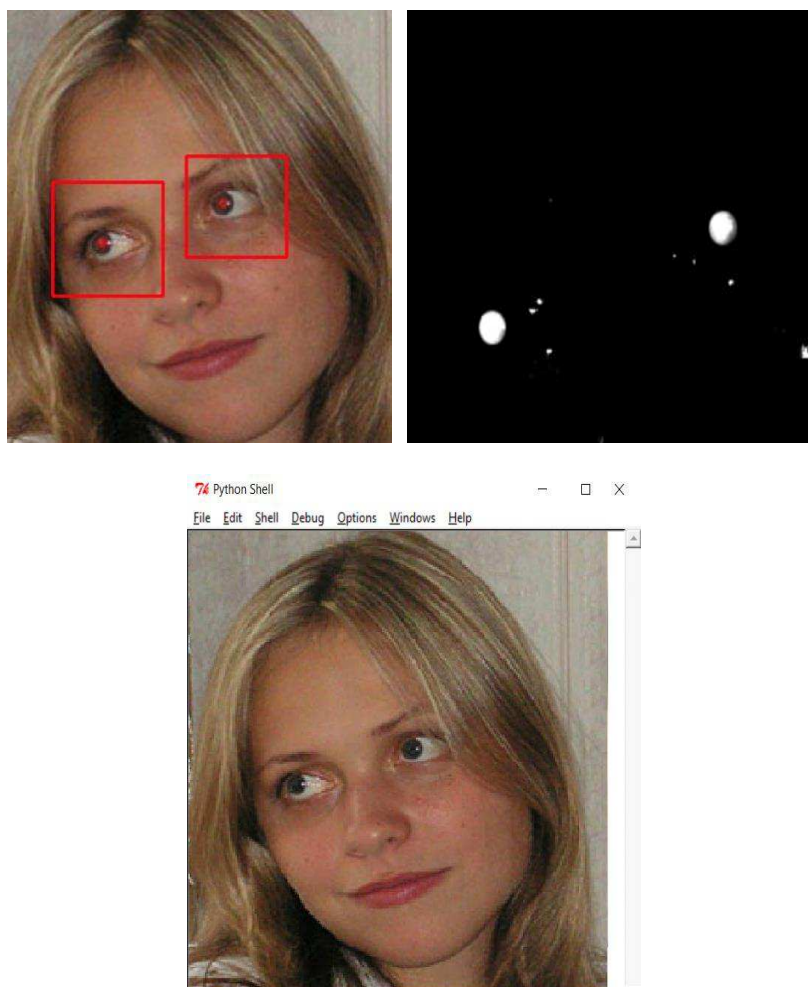


Рисунок 17 - Полученное в результате обработки изображение

При вычислении нового значения выбранного пикселя изображения, структурирующий элемент свёртки прикладывается своим центром к этому пикселю. Соседние пиксели так же накрываются структурирующим элементом.

Далее, вычисляется сумма произведений значений пикселей изображения на значения, накрывшего данный пиксель центрального элемента. Полученная сумма и является новым значением выбранного пикселя. Таким образом, если применить свёртку к каждому пикселю изображения, то получится эффект устранения пикселей красного цвета на другой (например чёрный).

#### **2.4 Удаление морщин на изображении**

При найденном на портретном изображении лице, для поиска морщин используется процесс бинаризации, далее используется алгоритм нахождения контуров, а затем алгоритм поиска Собеля [2]. Шаблон с областями, которые содержат морщины, подвергается последующему редактированию, в основе которого используется метод свёртки.

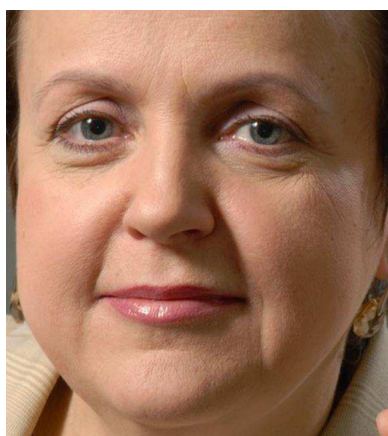


Рисунок 18 - Начальное изображение

Оператор Собеля основан на свёртке изображения небольшими сепарабельными целочисленными фильтрами в вертикальном и горизонтальном направлениях, поэтому его относительно легко вычислять.

Оператор вычисляет градиент яркости изображения в каждой точке. Так находится направление наибольшего увеличения яркости и величина её изменения в этом направлении. Результат показывает, насколько «резко» или «плавно» меняется яркость изображения в каждой точке, а значит, вероятность нахождения точки на грани. Резкие изменения яркости изображения позволяют находить области содержащие морщины.

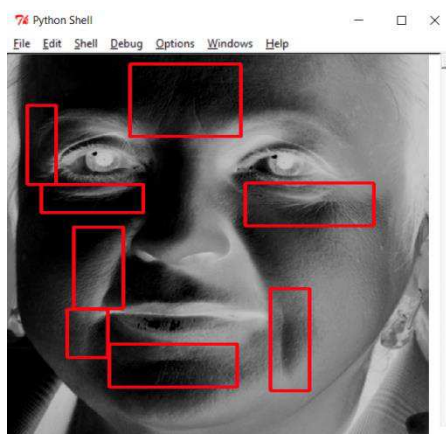


Рисунок 19 - Полученное в результате промежуточной обработки изображение

При вычислении нового значения выбранного пикселя изображения, структурирующий элемент свёртки прикладывается своим центром к этому пикселю. Соседние пиксели так же накрываются структурирующим элементом.

Далее, вычисляется сумма произведений значений пикселей изображения на значения, накрывшего данный пиксель центрального элемента. Полученная сумма и является новым значением выбранного пикселя. Таким образом, если применить свёртку к каждому пикселю изображения, то получится эффект устранения областей содержащих резкую смену яркости изображения. Изменяя такие области на изображении, происходит процесс



устранения морщин, либо заметное улучшение областей содержащих морщины.

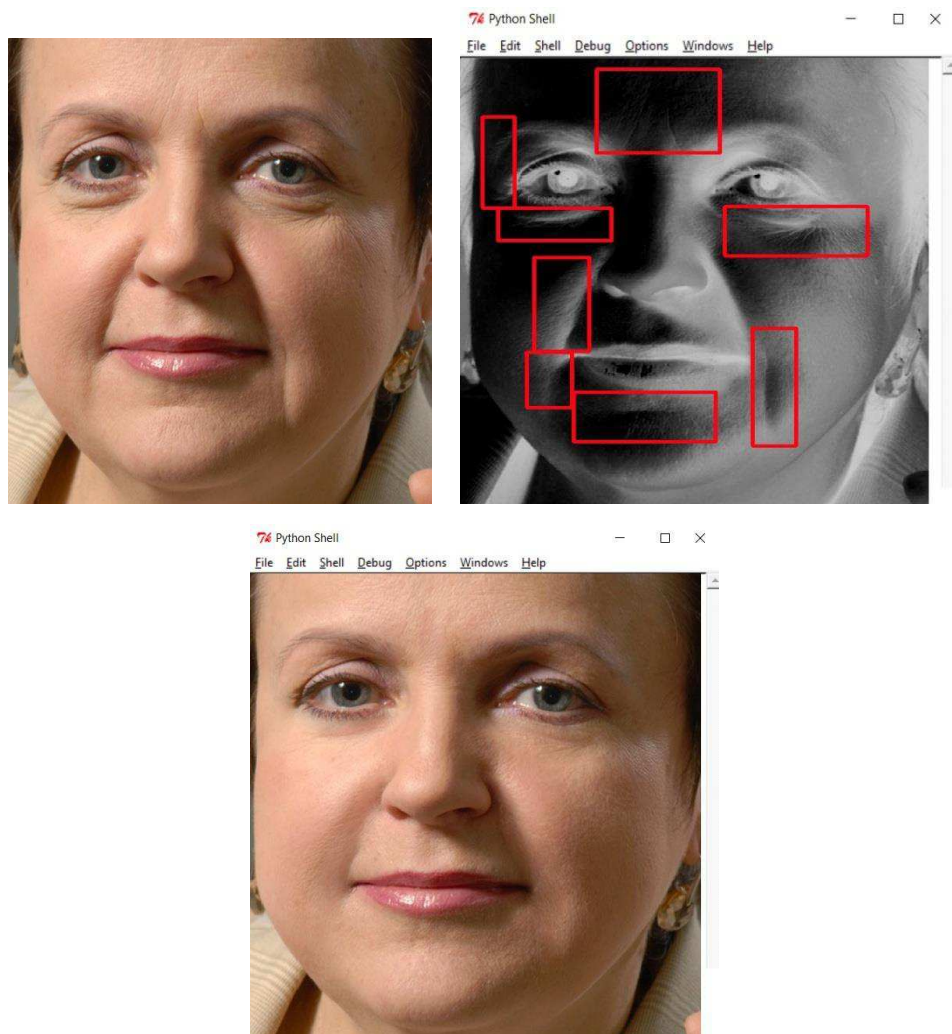


Рисунок 20 - Полученное в результате обработки изображение

## 2.5 Удаление родинок на лице

Если на изображении найдено лицо, для последующего удаления родинок используется процесс бинаризации. После бинаризации находятся области содержащие черные точки при помощи алгоритма Собеля. Шаблон с областями, которые содержат родинки, подвергается последующему редактированию, в основе которого используется метод свёртки.



Рисунок 21 - Начальное изображение

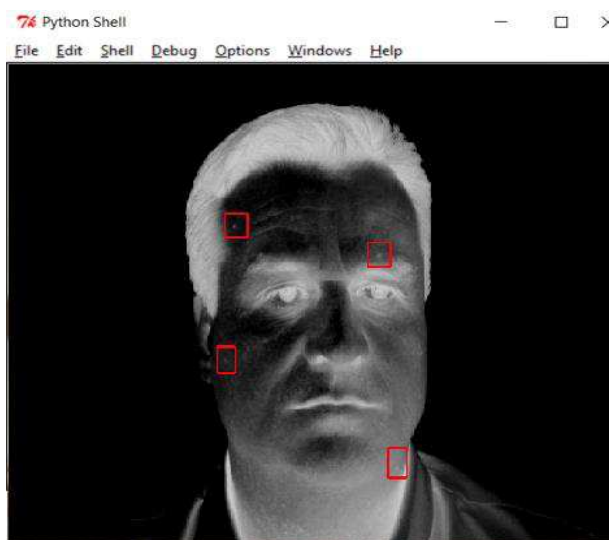


Рисунок 22 - Полученное в результате промежуточной обработки изображение

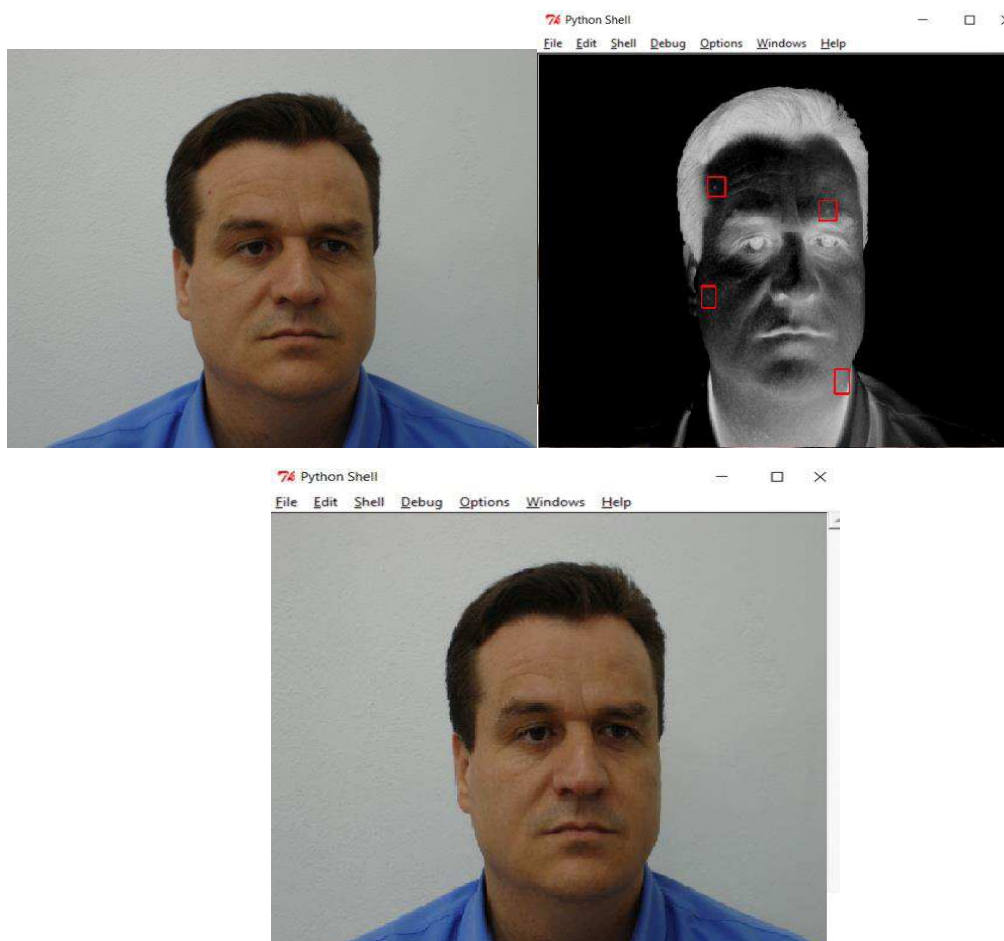


Рисунок 23 - Полученное в результате обработки изображение

## 2.6 Изменение цвета кожи на изображении

В изображении, в котором найдено лицо, для изменения цвета кожи используется процесс пороговой бинаризации. С помощью шаблонов каскадов Хаара находится область, содержащая глаза на портрете. Эта область не подлежит бинаризации и последующему редактированию. Оставшаяся область на изображении редактируется с помощью изменения уровня баланса цвета на изображении.



Рисунок 24 – Начальное изображение



Рисунок 25 - Полученное в результате промежуточной обработки изображение

Область, окрашенная в белый цвет, изменяет значения пикселей в направлении от 0 до 255, где 255 белый цвет, а 0 чёрный. Так как каждый пиксель на изображении уже содержит свой световой канал, изменение значений цветового потока элементов красного, зеленого и синего приводит к изменению общего тона изображения без резких скачков. Таким образом, получается равномерное изменение цветового баланса кожи.

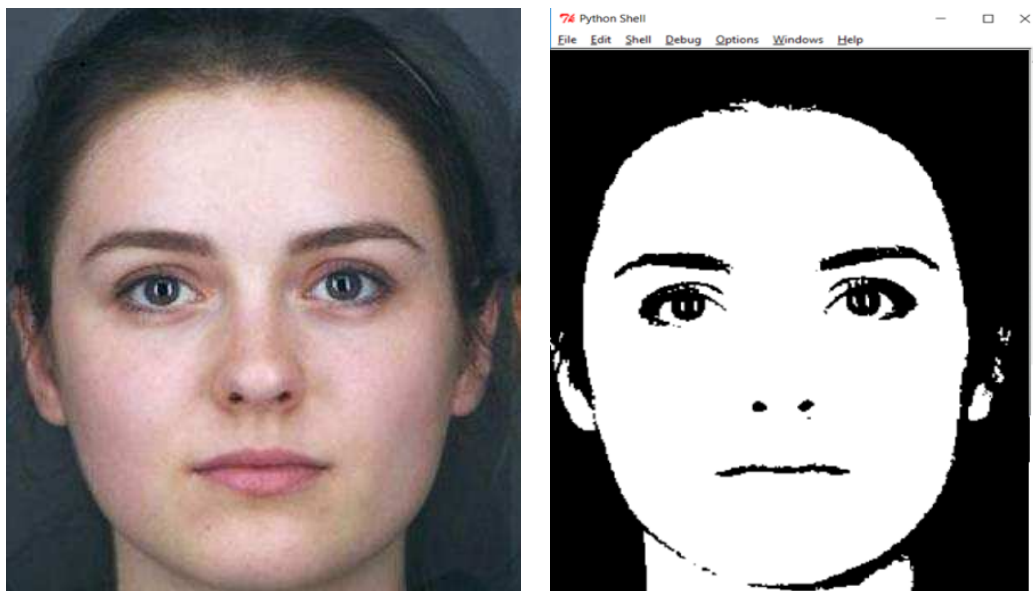


Рисунок 26 - Полученное в результате обработки изображение

### 3 Апробация программного продукта

При открытии программы появляется окно рабочей области. Для работы необходимо загрузить изображение. Для этого необходимо нажать на кнопку «добавить изображение», в появившемся диалоговом окне выбрать файл с изображением, нажать кнопку «открыть». Программа позволяет загружать изображения определенного формата, а именно jpg,jpeg,jpe,png.

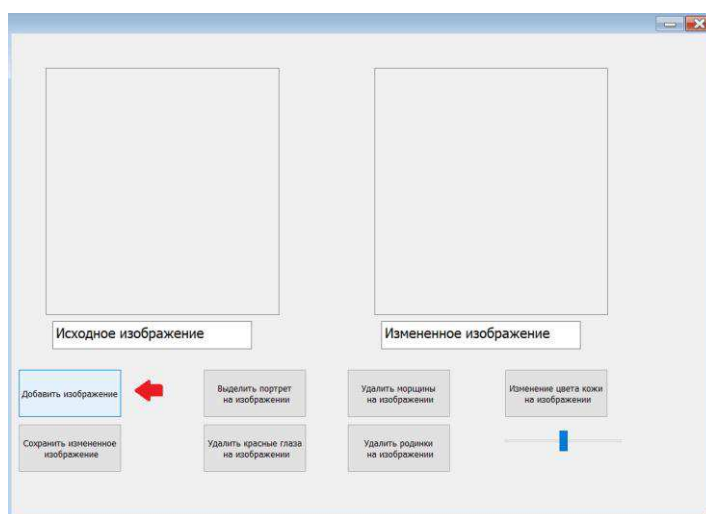


Рисунок 27 – Окно рабочей области программы

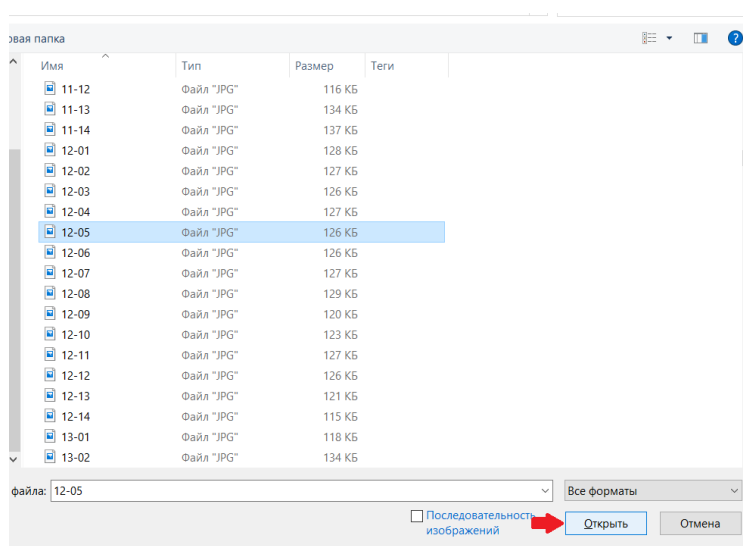


Рисунок 28 – Окно выбора файла изображения

После загрузки изображение появится в рабочей области программы. Для того чтобы посмотреть правильность работы алгоритмов каскада Хаара можно выделить портрет на изображении, нажав на кнопку «выделить портрет на изображении». Результат работы алгоритма можно будет увидеть в окне «измененное изображение».

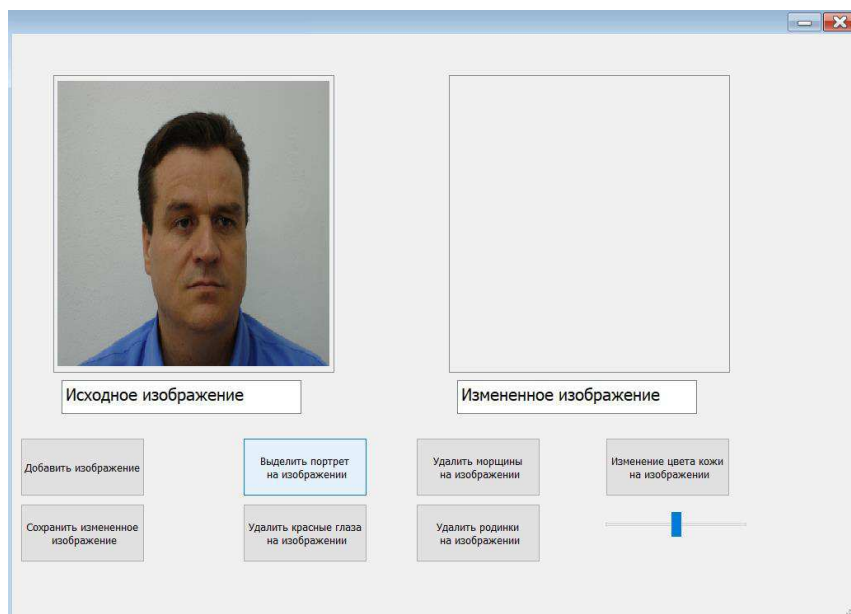


Рисунок 29 – Изображение в рабочей области программы

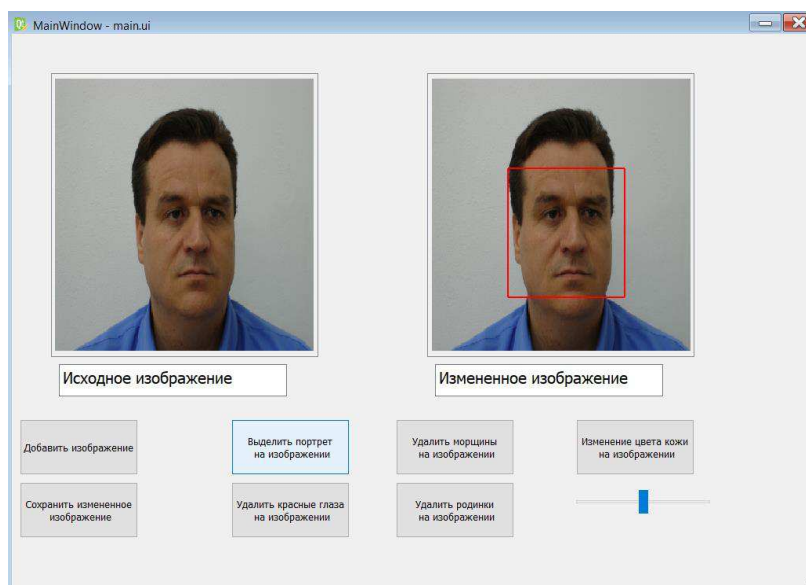


Рисунок 30 – Выделение портрета на изображении

Аналогичным образом, с помощью нажатия кнопок выполняются функции «удалить эффект красных глаз на изображении», «удалить морщины на изображении», «удалить родинки на изображении», ниже представлены примеры каждой из них:

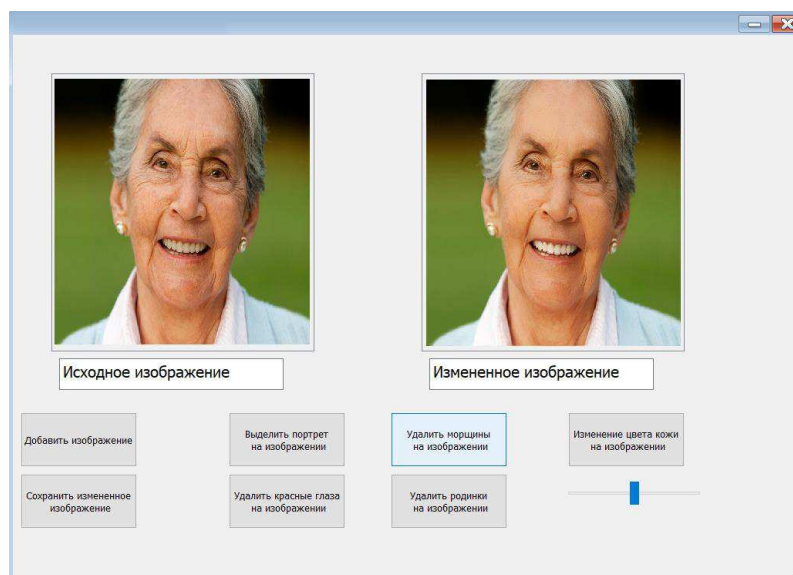


Рисунок 31 – Удаление морщин на изображении

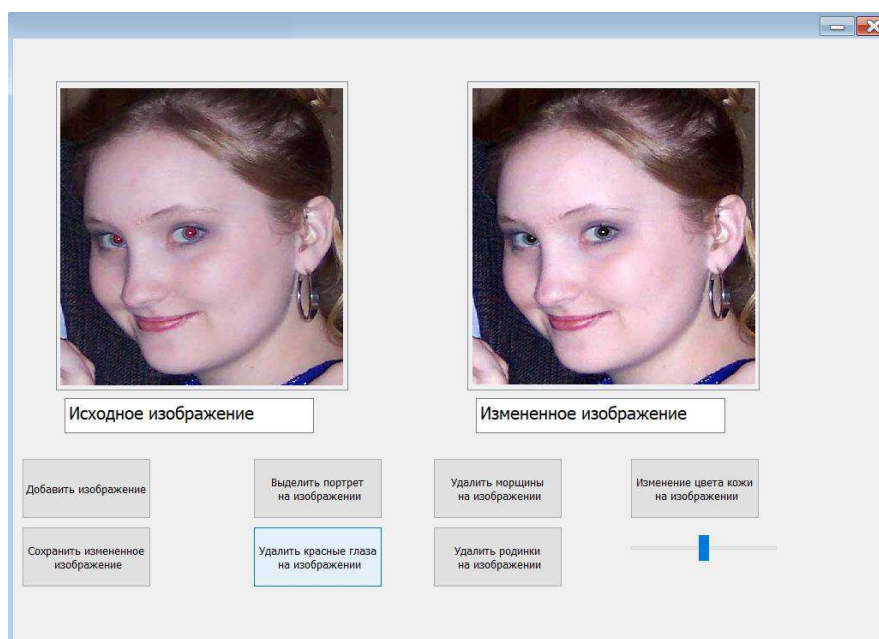


Рисунок 32 – Удаление эффекта «красных глаз» на изображении



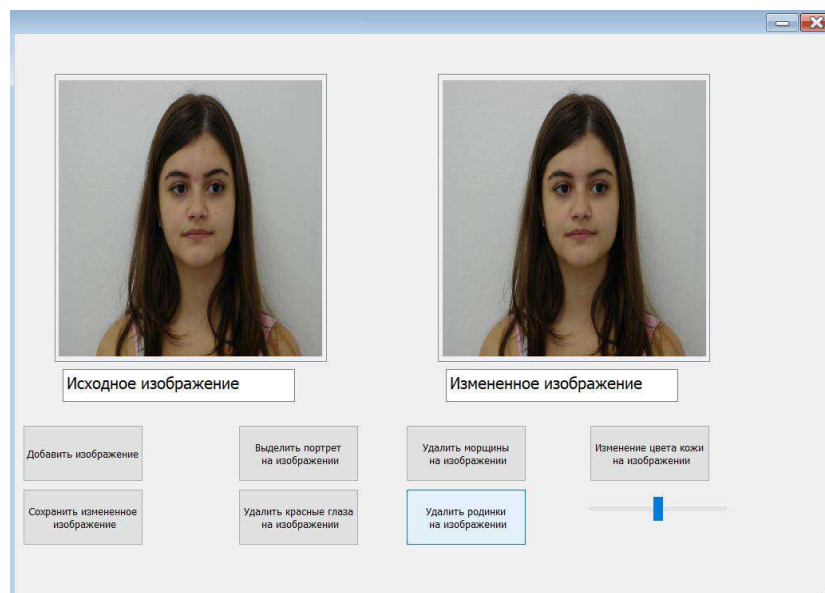


Рисунок 33 – Удаление родинок на изображении

Для использования функции «изменение цвета кожи на изображении» необходимо передвигать ползунок влево или вправо для придания кожи более светлого и тёмного оттенка соответственно. После сдвига ползунка регулировки необходимо нажать кнопку «изменение цвета кожи на изображении».

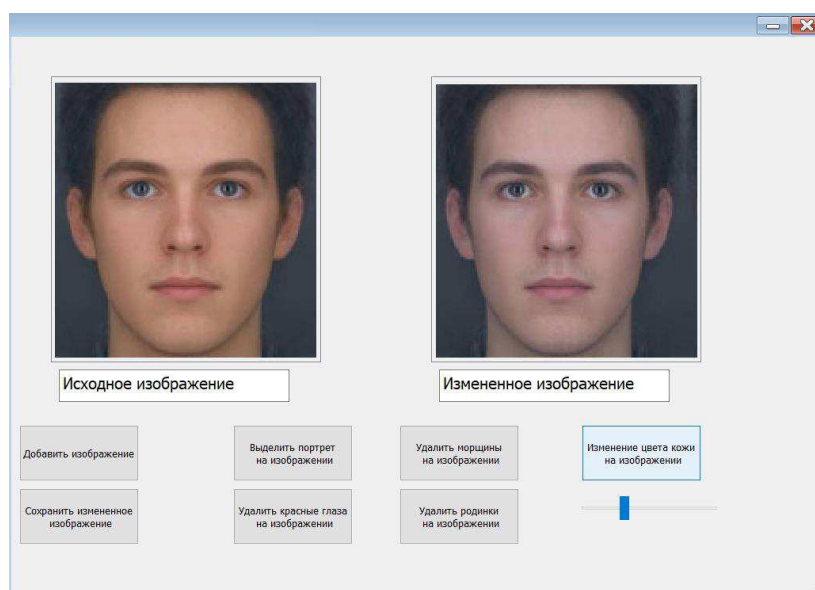


Рисунок 34 – Изменение цвета кожи (светлый оттенок)

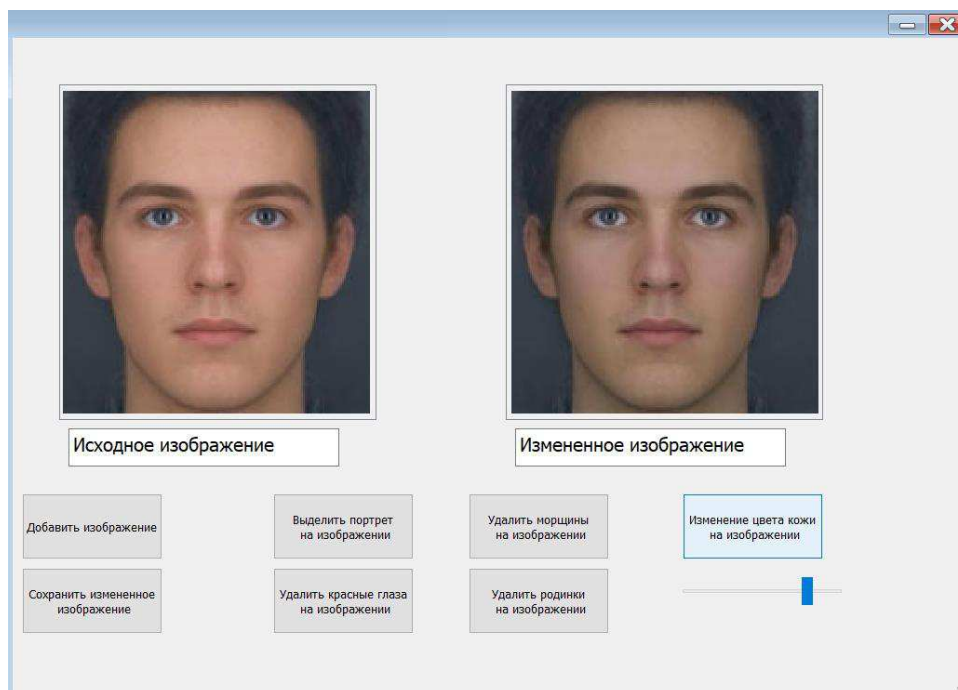


Рисунок 35 – Изменение цвета кожи (темный оттенок)

Для сохранения изображения необходимо нажать на кнопку «сохранить изображение» в появившемся окне указать директорию и имя файла, после этого выбрать кнопку сохранить

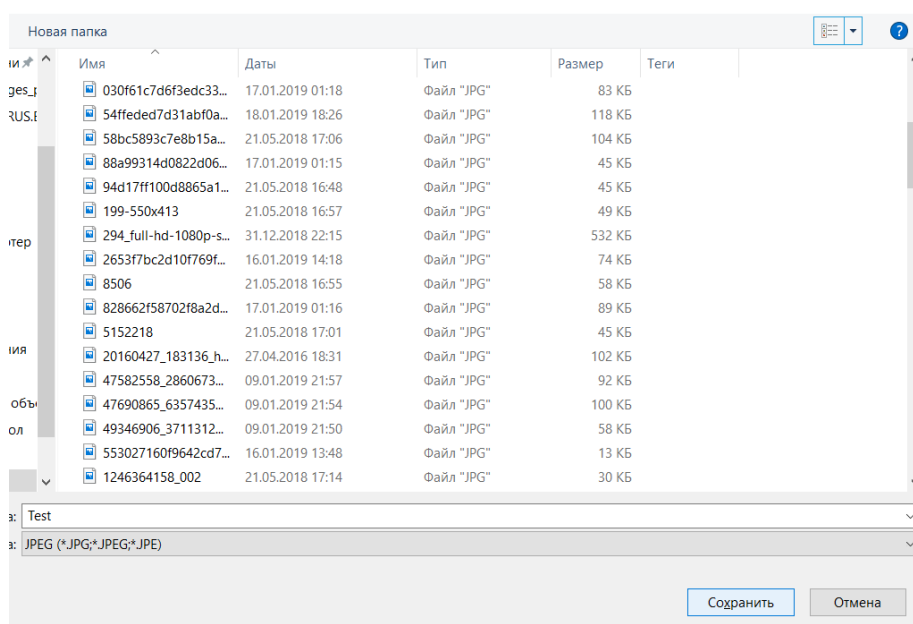


Рисунок 36 – Сохранение файла

## ЗАКЛЮЧЕНИЕ

В первой главе рассмотрена предметная область, выявлены проблемы и актуальность по выбранной теме научного исследования.

Актуальность темы «Разработка программного продукта для автоматизации обработки портретного изображения» была подтверждена, так как разработанный программный продукт, позволяет обрабатывать портретное изображение на основе пиксельного анализа фотоснимков что позволит сократить временные рамки получения результатов, сократить материальные затраты, увеличить скорость выполнения обработки данных, а так же широкие возможности для редактирования данных.

Главная проблема обработки портретного изображения заключается в том, что в настоящее время не существует универсального способа решения задач для достаточно широкого диапазона характеристик. Поэтому исследователям приходится находить новые модели или исследовать эффективность применения уже существующих.

Так же в первой главе произведен обзор существующих программных комплексов и приложения для автоматической обработки портретного изображения, были рассмотрены методы распознавания изображений по теме дипломной работы.

Во второй главе описаны методы обработки портретного изображения, сравнительный анализ полученных результатов с контрольными, представлен программный продукт, разработанный в ходе выполнения дипломной работы.

В третьей главе рассмотрена апробация программного продукта.

Итогом дипломной работы стал готовый к использованию продукт для обработки портретного изображения.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Брэд Смит Технологии распознавания лиц: необходимость в государственном регулировании и корпоративной ответственности [Электронный ресурс]. – Режим доступа: <https://news.microsoft.com/ru-ru/facial-recognition-technology/> (дата обращения: 07.03.2019).
2. Спицын В.Г. Анализ методов выделения краев на цифровых изображениях, Доклады ТУСУРа, № 2 (22), часть 2, декабрь 2010 [Электронный ресурс]. – Режим доступа: <http://masters.donntu.org/2013/fknt/dotsenko/library/art2.htm> (дата обращения: 07.03.2019).
3. Ватаманюк И.В. Применение методов оценивания размытости цифровых изображений в задаче аудиовизуального мониторинга. [Электронный ресурс]. – Режим доступа: <https://cyberleninka.ru/article/v/primeneniemetodovotsenivaniyarazmytosti-tsifrovyyh-izobrazheniy-v-zadache-audiovizualnogo-monitoringa> (дата обращения: 23.04.2019).
4. Гагарина Л.Г., Морозова Н.В., «Современные информационные технологии реализации алгоритмов визуализации цветных изображений» [Электронный ресурс]. – Режим доступа: <https://www.dissercat.com/content/razrabotka-programmno-apparatnykh-sredstv-dlya-obrabotki-informatsionnykh-signalov-na-osnove/read> (дата обращения: 23.04.2019).
5. Гонсалес Р., Вудс Р., Цифровая обработка изображений [Электронный ресурс]. – Режим доступа: <https://www.twirpx.com/file/489868/> (дата обращения: 23.04.2019).
6. Дмитриев Е.А., Танаев И.В., Швейкин В.В. Метод Виолы-Джонса [Электронный ресурс]. – Режим доступа: <https://sibac.info/studconf/science/vi/60676> (дата обращения: 07.03.2019).

7. Ефремов А. Секреты RAW Профессиональная обработка [Электронный ресурс]. – Режим доступа: <https://avidreaders.ru/book/sekrety-raw-professionalnaya-obrabotka.html> (дата обращения: 23.04.2019).
8. Малашин Р. О. Методы структурного анализа изображений трехмерных сцен: [Электронный ресурс]. – Режим доступа: <http://tekhnosfera.com/metody-strukturnogo-analiza-izobrazheniy-trehmernyh-stsen> (дата обращения: 23.04.2019).
9. Местецкий Л.М. Математические методы распознавания образов [Электронный ресурс]. – Режим доступа: <http://www.ccas.ru/frc/papers/mestetskii04course.pdf> (дата обращения: 23.04.2019).
10. Метод адаптивной бинаризации растрового изображения Вдовин В.А., Муравьев А.В., Певзнер А.А. [Электронный ресурс]. – Режим доступа: <https://docplayer.ru/47498084-Metod-adaptivnoy-binarizacii-rastrovogo-izobrazheniya-65-a-method-of-adaptive-binarization-of-the-raster-image.html> (дата обращения: 23.04.2019).
11. Петров М.Н., Молочков В.П. Компьютерная графика [Электронный ресурс]. – Режим доступа: [https://www.studmed.ru/petrov-m-molochkov-v-kompyuternaya-grafika\\_6e6077bf22e.html](https://www.studmed.ru/petrov-m-molochkov-v-kompyuternaya-grafika_6e6077bf22e.html) (дата обращения: 23.04.2019).
12. Потапов А. А., Пахомов А. А., Никитин С. А., Гуляев Ю. В., Новейшие методы обработки изображений [Электронный ресурс]. – Режим доступа: [https://www.studmed.ru/potapov-aa-gulyaev-yuv-nikitov-sa-pahomov-aa-german-va-noveyshie-metody-obrabotki-izobrazheniy-chast-2\\_3fccc1ea030.html](https://www.studmed.ru/potapov-aa-gulyaev-yuv-nikitov-sa-pahomov-aa-german-va-noveyshie-metody-obrabotki-izobrazheniy-chast-2_3fccc1ea030.html) (дата обращения: 23.04.2019).
13. Прэтт У. Цифровая обработка изображений [Электронный ресурс]. – Режим доступа: [http://sernam.ru/book\\_prett1.php](http://sernam.ru/book_prett1.php) (дата обращения: 23.04.2019).

14. Рудаков А.В., Федорова Г.Н. Технология разработки программных продуктов [Электронный ресурс]. – Режим доступа: [http://itam.irk.ru/biblioavia/catalog/1505188188\\_351.pdf](http://itam.irk.ru/biblioavia/catalog/1505188188_351.pdf) (дата обращения: 23.04.2019).
15. Харинов М.В. Альтернатива иерархическому методу Оцу для цветного изображения [Электронный ресурс]. – Режим доступа: <https://cyberleninka.ru/article/n/alternativa-ierarhicheskomu-metodu-otsu-dlya-tsvetovogo-izobrazheniya> (дата обращения: 23.04.2019).
16. Харинов М. В. Улучшение качества приближения на основе иерархической сегментации [Электронный ресурс]. – Режим доступа: <https://cyberleninka.ru/article/n/uluchshenie-kachestva-priblizheniya-izobrazheniya-na-osnove-ierarhicheskoj-segmentatsii> (дата обращения: 23.04.2019).
17. Яковлев С. Модули и пакеты в Python. Глубокое погружение [Электронный ресурс]. – Режим доступа: <https://devpractice.ru/python-modules-and-packages/> (дата обращения: 23.04.2019).

## ПРИЛОЖЕНИЕ А

Таблица А.1 – Описание диаграммы классов для программного продукта для автоматической обработки изображения

Метод	Параметры	Описание
<i>LoadImage</i> ( <i>filename</i> , <i>imagecolor</i> )	<b>Входные:</b> <i>filename</i> – строка содержащая название файла изображения; <i>imagecolor</i> – переменная, задающая цветовой формат получаемого изображения. <b>Выходные:</b> <i>image</i> – двумерный массив пикселей, содержащий в себе цветное значение каждого пикселя.	Метод используется для получения изображения для дальнейшей обработки. Позволяет получать изображение в нескольких цветовых форматах в зависимости от заданных параметров.
<i>CreateImage</i> ( <i>size</i> , <i>depth</i> , <i>channel</i> )	<b>Входные:</b> <i>size</i> – размер области, <i>depth</i> – битовая глубина элементов изображения, <i>channels</i> – количество каналов на пиксель. <b>Выходные:</b> <i>field</i> – двумерный массив пикселей, содержащий в себе цветное значение каждого пикселя.	Метод используется для создания пустой области в которое в дальнейшем будет загружено изображение. Применяется в случаях, когда необходимо обработать часть изображения или создать пустую область заданных размеров.
<i>ImageCopy</i> ( <i>image</i> )	<b>Входные:</b> <i>image</i> – массив пикселей изображения. <b>Выходные:</b> <i>copyimage</i> – массив пикселей изображения.	Метод используется для копирования изображения с сохранением цветных значений каждого пикселя.
<i>CircuitImage</i> ( <i>image</i> , <i>first_contour</i> , <i>header_size</i> )	<b>Входные:</b> <i>image</i> – массив пикселей, содержащий исходное изображение <i>first_contour</i> – указатель, на первый элемент последовательности, <i>header_size</i> – размер заголовка элемента последовательности	Метод используется для нахождения контуров на изображении.

Метод	Параметры	Описание
	<p><b>Выходные:</b> <i>Circuitarea</i> – область содержащая обработанное изображение</p>	
<p><i>HaarCascade</i> (<i>cascade_default</i>, <i>image</i>)</p>	<p><b>Входные:</b> <i>cascade_default</i> – шаблонные элементы областей каскадов, <i>image</i> – массив пикселей, содержащий исходное изображение. <b>Выходные:</b> <i>Cascadearea</i> – область содержащая обработанное изображение</p>	<p>Метод используется для нахождения областей каскадов Хаара на изображении.</p>
<p><i>BinaryImage</i> (<i>image</i>,<i>treshold</i>)</p>	<p><b>Входные:</b> <i>image</i> – массив пикселей, содержащий исходное изображение, <i>treshold</i> – пороговое значение. <b>Выходные:</b> <i>Binaryarea</i> – массив пикселей содержащий обработанное изображение.</p>	<p>Метод реализует обработку изображения методом бинаризации. Пороговое значение сравнивается со значением каждого пикселя для присвоения элементу нового значения цвета.</p>
<p><i>SobelFindImage</i> (<i>image</i>,<i>xord</i>,<i>yord</i>,<i>scale</i>)</p>	<p><b>Входные:</b> <i>image</i> – массив пикселей, содержащий исходное изображение, <i>xord</i> – порядок производной x, <i>yord</i> – порядок производной y, <i>scale</i> – масштабный коэффициент . <b>Выходные:</b> <i>Sobelarea</i> – массив пикселей содержащий обработанное изображение.</p>	<p>Метод реализует обработку изображения методом собеля. Масштабный коэффициент применяется для вычисленных производных значений.</p>
<p><i>PortraitFind</i> (<i>image</i>, <i>haarCascadearea</i>, <i>circuitarea</i>)</p>	<p><b>Входные:</b> <i>image</i> – массив пикселей, содержащий исходное изображение, <i>haarCascadearea</i> – массив пикселей, содержащий область найденных каскадов Хаара, <i>circuitarea</i> – массив</p>	<p>Метод, реализующий поиск областей, содержащих портрет на изображении.</p>



Продолжение приложения

Метод	Параметры	Описание
	<p>пикселей, содержащий область найденных контуров.</p> <p><b>Выходные:</b> <i>Portraitarea</i> – массив пикселей содержащий обработанное изображение.</p>	
<p><i>wrinFind (image, haarCascadearea)</i></p>	<p><b>Входные:</b> <i>image</i> – массив пикселей, содержащий исходное изображение, <i>haarCascadearea</i> – массив пикселей, содержащий область найденных каскадов Хаара.</p> <p><b>Выходные:</b> <i>newimage</i> – массив пикселей, содержащий обработанное изображение.</p>	<p>Метод, реализующий поиск и удаление морщин на портретном изображении. Возвращает обработанное изображение на выходе.</p>
<p><i>birthFind (image, haarCascadearea)</i></p>	<p><b>Входные:</b> <i>image</i> – массив пикселей, содержащий исходное изображение, <i>haarCascadearea</i> – массив пикселей, содержащий область найденных каскадов Хаара.</p> <p><b>Выходные:</b> <i>newimage</i> – массив пикселей, содержащий обработанное изображение.</p>	<p>Метод, реализующий поиск и удаление родинок на портретном изображении. Возвращает обработанное изображение на выходе.</p>
<p><i>skinFind (image, haarCascadearea, shadeobj)</i></p>	<p><b>Входные:</b> <i>image</i> – массив пикселей, содержащий исходное изображение, <i>haarCascadearea</i> – массив пикселей, содержащий область найденных каскадов Хаара, <i>shadeobj</i> – переменная определяющая оттенок изменения тона кожи.</p> <p><b>Выходные:</b> <i>newimage</i> – массив пикселей, содержащий обработанное изображение.</p>	<p>Метод, реализующий изменение оттенка кожи на портретном изображении. Возвращает обработанное изображение на выходе.</p>

Продолжение приложения

Метод	Параметры	Описание
<i>convolutionImage</i> ( <i>image, area</i> )	<p><b>Входные:</b> <i>image</i> – массив пикселей, содержащий исходное изображение, <i>area</i> – массив пикселей, содержащий область для изменения.</p> <p><b>Выходные:</b> <i>newimage</i> – массив пикселей, содержащий обработанное изображение.</p>	Метод, реализующий свертку изображения в указанной области.
<i>ImageShow</i> ( <i>winname, image</i> )	<p><b>Входные:</b> <i>image</i> – массив пикселей, содержащий исходное изображение, <i>winname</i> – номер окна для вывода изображения.</p> <p><b>Выходные:</b> отсутствуют</p>	Метод реализующий вывод изображения на экран.

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий  
Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

 О. В. Непомнящий

подпись      инициалы, фамилия  
«28» 06 2019 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.01 Информатика и вычислительная техника

Разработка программного продукта для автоматической обработки  
портретного изображения.

Руководитель


 20.06.2019  
подпись, дата

доцент, канд. техн. наук  
должность, учёная степень

П. П. Дьячук

инициалы, фамилия


Выпускник

 20.06.2019  
подпись, дата

А. А. Ключвин

инициалы, фамилия

Нормоконтролер

 27.06.19  
подпись, дата

доцент, канд. техн. наук  
должность, учёная степень

В. И. Иванов

инициалы, фамилия

Красноярск 2019