

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт Математики, физики и информационных технологий  
(наименование института полностью)

Кафедра «Прикладная математика и информатика»  
(наименование)

01.03.02 Прикладная математика и информатика  
(код и наименование направления подготовки, специальности)

Компьютерные технологии и математическое моделирование  
(направленность (профиль)/специализация)

## ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка алгоритма идентификации людей по биометрическим данным лица»

Студент

Д.В. Ханипов

(Инициалы Фамилия)

*Ханипов*

(личная подпись)

Руководитель

к.т.н., В.С. Климов

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

Е.В. Косс

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

## Аннотация

Тема выпускной квалификационной работы: Разработка алгоритма идентификации людей по биометрическим данным лица.

Идентификации людей по биометрическим данным лица крайне актуальна, так как эта тема крепко связана с безопасностью. Алгоритмы идентификации людей по биометрическим данным лица все больше используются в современной жизни, ярким примером являются разблокировка смартфона по лицу.

Выпускная квалификационная работа посвящена анализу методов использующихся в алгоритмах идентификации людей по биометрическим данным лица, а также разработки алгоритма идентификации людей по биометрическим данным лица.

Цель выпускной квалификационной работы – Исследование методов распознавания и идентификации лиц людей и разработка рабочего алгоритма идентификации людей по биометрическим данным лица.

Объект выпускной квалификационной работы – идентификации людей по биометрическим данным лица.

Предмет выпускной квалификационной работы – алгоритм идентификации людей по биометрическим данным лица, в основе которого машинное обучение.

Задачи работы:

- анализ общего вида задачи,
- исследование методов решения задачи,
- разработка алгоритма идентификации,
- реализация программного обеспечения;

Структура выпускной квалификационной работы состоит из введения, трех глав, заключения, списка литературы.

В первой главе происходит анализ проблемы задачи идентификации людей по биометрическим данным лица.

Во второй главе происходит анализ метод для решения задачи идентификации людей по биометрическим данным лица, и обоснования выбранного подхода дальнейшей реализации.

В третий главе описывается разработка алгоритма идентификации людей по биометрическим данным лица, приводится пример работы разработанного алгоритма.

Пояснительная записка данной выпускной квалификационной работы, состоит из 20 рисунков, 1 таблица, 7 формул и 25 источник используемой литературы. Объем пояснительной записки составляет 41 страницы.

## **Abstract**

The topic of the graduate qualification work: Development of an algorithm for identifying people by facial biometric data.

Identification of people by facial biometrics is extremely relevant, as this topic is strongly related to security. Algorithms to identify people using biometric face data are increasingly used in modern life, a prime example is the unlocking of the smartphone by face.

The aim of the graduate qualification work – research methods of recognition and identification of faces and development of working algorithm for the identification of people by biometric data face.

The object of the graduate qualification work – identification of people by biometric facial data.

The subject of the graduate qualification work is an algorithm for identifying people by biometric facial data, which is based on machine learning.

Objectives of the work:

- analysis of the general view of the problem,
- study of methods for solving the problem,
- development of the identification algorithm,
- software implementation;

The structure of the graduate qualification work consists of an introduction, three chapters, a conclusion, and a list of references.

The first chapter analyzes the problem of identifying people by biometric face data.

The second chapter analyzes the method for solving the problem of identification of people by biometric face data, and the rationale for the chosen approach for the further implementation.

The third chapter describes the development of an algorithm for the identification of people based on facial biometrics, gives an example of the developed algorithm.

Graduation qualification work consists of an explanatory note, including 20 figures, 1 table, 7 formulas and 25 sources of references. The volume of the explanatory note is 41 pages.

## Оглавление

Введение.....	7
Глава 1 Анализ алгоритмов идентификации людей по биометрическим данным лица .....	8
1.1 Общие решение задачи системы распознавания лиц.....	8
1.2 Проблематика идентификации людей по биометрическим данным лица .....	9
1.3 Метод Виолы – Джонса.....	11
1.4 Гистограмма направленных градиентов и метод опорных векторов .....	14
Глава 2 Анализ задачи идентификации людей по биометрическим данным лица .....	19
2.1 Постановка задачи .....	19
2.2 Работа алгоритма Виолы-Джонса .....	19
2.3 Обоснование выбранного метода.....	22
Глава 3 Разработка алгоритма идентификации людей по биометрическим данным лица.....	26
3.1 Среда разработки и язык .....	26
3.2 Локализация признаков.....	26
3.3 Обучение распознавателя.....	30
3.4 Сбор полной программы и тестирование .....	32
Заключение .....	38
Список используемой литературы .....	39

## Введение

В современном мире всё чаще люди задумываются о безопасности своих домов, предприятий, личных вещей и данных на мобильных устройствах. Одним из таких способов повышения безопасности является применение систем автоматической идентификации работающей на основе технологии распознавания лиц.

В связи с распространением устройств видеонаблюдения и также повышения их качества, и возможности с помощью их отслеживать людей системы распознавания лица, которые могут идентифицировать человека, становятся все актуальнее. Вспоминая примеры использования систем распознавания и идентификации лиц, сразу на ум приходят такие примеры, как «Face ID» и «умный домофон». «Face ID» позволяет быстро при помощи своего лица разблокировать смартфон, а также является одним из безопаснейших методов разблокировки. «Умный домофон» позволяет открывать дверь подъезда, используя лицо без необходимости использования физических ключей.

Цель выпускной квалификационной работы – исследование методов распознавания и идентификации лиц людей и разработка рабочего алгоритма идентификации людей по биометрическим данным лица.

Объект выпускной квалификационной работы – идентификации людей по биометрическим данным лица.

Предмет выпускной квалификационной работы – алгоритм идентификации людей по биометрическим данным лица, в основе которого машинное обучение.

Для достижения поставленной цели в работе решаются следующие задачи:

- анализ и исследование методов решения задачи,
- разработка алгоритма идентификации,
- реализация программного обеспечения;

# **Глава 1 Анализ алгоритмов идентификации людей по биометрическим данным лица**

## **1.1 Общие решение задачи системы распознавания лиц**

Для начала стоит ответить на вопрос, что такое система распознавание лиц. Что бы лучше понять что-то, давайте обратимся к понятию facial recognition system (в переводе на русский система распознавания лица).

Система распознавания лица — это технология, способная сопоставить человеческое лицо с цифровым изображением или видеокадром по базе данных лиц. Зная, что система распознавания лица решает задачу идентификации людей по биометрическим данным лица следует, что главная задача идентификации людей по биометрическим данным это сопоставления человеческого лица с цифровым изображением или видеокадром по базе данных лиц [3].

Решение данной задачи стоит разбить на две подзадачи первая определение лица на изображении, вторая само сравнение исходного изображения и изображениями из базы данных.

Само сравнение также имеет два этапа, первый это этап перевода изображение в удобный вид для сравнения в нашем случаи перевод изображения в вектор, далее это будет называться векторизация изображения. Вторым этапом это сравнения полученные векторы.

Таким образом, приходим к общему виду задачи, распознавания людей по биометрическим данным лица этапы решения которую я описал в своей статье для научно-практической конференции «Студенческие дни науки ТГУ».

Задача идентификации людей по лицу решается в три этапа. Первый этап это этап распознавания объектов на изображение в случае задачи идентификации распознавание лица. Вторым этапом является векторизация

изображения, то есть перевод изображения в вектор. Третьим этапом является сравнение полученных векторов [9].

## **1.2 Проблематика идентификации людей по биометрическим данным лица**

Давайте рассмотрим две проблемы, встречающиеся во время идентификации людей по биометрическим данным лица.

Первая это проблема точности идентификации, данная проблема может быть вызвана некачественным изображением или схожестью проверяемых объектов в нашем случае лиц людей. Данная проблема может вызвать такие ошибки в работе программы как не нахождение человека в базе данных известных лиц, или наоборот человека, который не зарегистрирован в базе данных, за кого-то кто зарегистрирован в базе данных. Это может помочь злоумышленникам в обмане системы распознавания лица или просто усложнить жизнь пользователям этой системы. Так же большая точность влечет замедление работы программы, что также может негативно сказаться на пользователях [11].

Вторая проблема быстрота работы идентификации, данная проблема является противоположной проблеме точности, ведь чтобы достичь высокой точности надо затратить больше времени, чем, если удовлетвориться более низкой точностью. В случае же уменьшения точности также будем увеличиваться скорость работы алгоритмы.

Как будет рассмотрено далее, разные алгоритмы имеют разную точность и скорость работы. Одна из целей этой работы разработать алгоритм, который будет являться чем-то средним между проанализированными далее алгоритмами.

Прежде чем переходить к методам распознавания объектов разберемся с понятием цифровое изображение. Хотя многим интуитивно понятны определения изображения и цифрового изображения, давайте

Под изображением будем понимать действительнзначную функцию,  $f(y, x)$ , определенную на некотором подмножестве пространства  $R^2$ . Значение функции  $f(y, x)$  называется интенсивностью или яркостью, реже уровнем серого [6], [11], [17].

Изображение  $f(y, x)$  будем называть цифровым изображением, если величины  $y$  и  $x$  принимают конечное число дискретных значений. В силу того, что величины  $y$  и  $x$  принимают конечное число дискретных значений, цифровое изображение можно представить в виде матрицы из  $M$  строк и  $N$  столбцов, где  $(y, x)$  задают координаты, а  $f(y, x)$  – значение элемента матрицы. В дальнейшем для большего удобства будем использовать целочисленные значения  $y = \{0, 1, 2, \dots, M - 1\}$  и  $x = \{0, 1, 2, \dots, N - 1\}$ , принимая за начало координат левый верхний угол изображения, где  $(y, x) = (0, 0)$ , ось  $y$  направлена вниз, а ось  $x$  – вправо. Область действительной координатной плоскости, охватываемая координатами изображения, называется пространственной областью, а  $y$  и  $x$  – пространственными переменными или пространственными координатами [17].

Таким образом, цифровое изображение в виде матрицы можно записать следующим образом:

$$f(y, x) = \begin{pmatrix} f(0,0) & \dots & f(0, M - 1) \\ \vdots & \ddots & \vdots \\ f(N - 1, 0) & \dots & F(N - 1, M - 1) \end{pmatrix} \quad (1)$$

Каждый элемент матрицы (1) называются элементами изображения или пикселями (образовано от английского словосочетания picture element – «элемент изображения») [8].

В процессе получения цифрового изображения помимо определения величин  $M$  и  $N$  (пространственного размера изображения) возникает еще число  $L$  – количество дискретных уровней (градаций) яркости. Из соображений удобства и особенностей хранения цифровых изображений

значение  $L$  обычно выбирают равным целочисленной степени двойки:  $L = 2^k$

Предполагается, что дискретные уровни яркости расположены с постоянным шагом и принимают целые значения в интервале  $[0, L - 1]$ . Тогда для хранения цифрового изображения необходимо следующее количество бит:

$$b = M \cdot N \cdot k, \quad (2)$$

Для вычислений или разработки алгоритмов бывает полезно отобразить шкалу  $L$  значений яркости на интервал  $[0, 1]$ , так что они перестают быть целочисленными. Однако при хранении и визуализации изображений, в большинстве случаев полученные значения масштабируются обратно на интервал целых значений  $[0, L - 1]$ .

### 1.3 Метод Виолы – Джонса

Первым методом рассмотрим метод Виолы – Джонса. В этом методе используют «набор функции, которые напоминают базовые функции Хаара» [10], [15], [25]. В источнике, авторы о своем методе в переводе на русский, заявляют. «Чтобы очень быстро вычислить эти функции во многих масштабах, мы вводим интегральное представление изображения для изображений (интегральное изображение очень похоже на таблицу суммированной площади, используемую в компьютерной графике для отображения текстур). Интегральное изображение может быть вычислено по изображению с использованием нескольких операций на пиксель. После вычисления любая из этих функций, подобных Хаар, может быть вычислена в любом масштабе или местоположении в постоянном времени» [10], [15], [25].

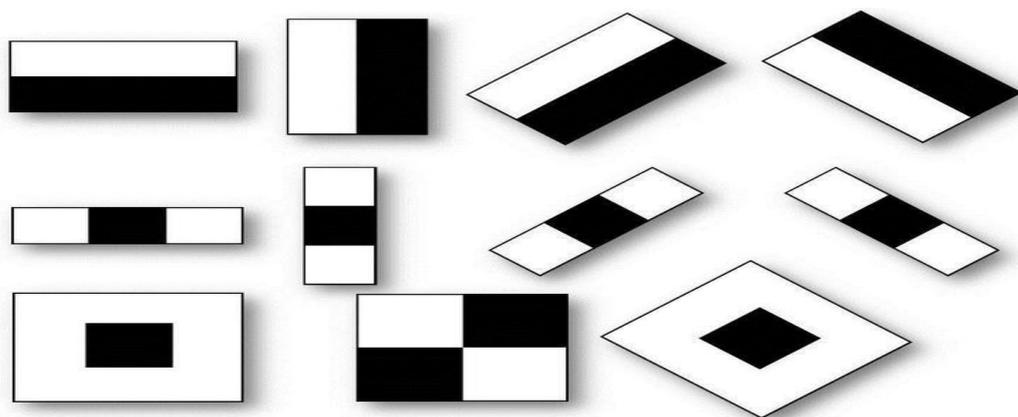


Рисунок 1 – Примитивы Хаара

На рисунке 1 показаны примитивы Хаара, на основе которых и построен метод. Рассмотрим анализ этого детектора Хаара по сравнению с подходом, при котором вычисляется пирамида изображений. Авторы метода провели сравнения и в переводе на русский заявляют.

«Как и большинство систем обнаружения объектов, наш детектор сканирует входные данные во многих масштабах; начиная с базового масштаба, в котором объекты обнаруживаются размером 24x24 пикселя, изображение сканируется в 11 масштабах, каждый из которых в 1,25 раза больше предыдущего. То обычный подход заключается в вычислении пирамиды из 11 изображений, каждое из которых в 1,25 раза меньше предыдущего изображения. Затем по каждому из этих изображений сканируется детектор фиксированного масштаба. Вычисление пирамиды, хотя и простое, требует значительного времени. На обычном оборудовании, чрезвычайно сложно вычислить пирамиду со скоростью 15 кадров в секунду» [15], [25].

«Учитывая вычислительную эффективность этих функций, процесс распознавания лиц может быть завершен для всего изображения в любом масштабе со скоростью 15 кадров в секунду, что меньше времени, чем требуется для оценки только 11-уровневой пирамиды изображений. Любая процедура, требующая пирамиды такого типа, обязательно будет выполняться медленнее, чем наш детектор» [15], [25].

Признак для данной области (отклик) на представленное свойство можно вычислить, используя выражение:

$$R = \frac{S_{\bar{b}}}{N_{\bar{b}}} - \frac{S_{\bar{ч}}}{N_{\bar{ч}}}, \quad (3)$$

где  $N$  – количество пикселей заключенных вокруг области объекта;

$S$  – сумма яркостей области объекта.

Но данная формула используется, когда показывают объекты с пересекающимися областями, то есть находящиеся на одной линии, если рассматривать случай, когда объект наезжает на другой объект, то можно сказать что области данных объектов не пересекаются и формула изменяется:

$$R = \frac{S_{\bar{b}}}{N_{\bar{b}}} - \frac{S_{\bar{ч}} - S_{\bar{ч} \cap \bar{b}}}{N_{\bar{ч}} - N_{\bar{ч} \cap \bar{b}}}, \quad (4)$$

где  $N$  – количество пикселей заключенных вокруг области объекта;

$S$  – сумма яркостей области объекта.

Находиться свойство, имеющие признак постоянности при использовании любых линейных преобразований яркости для обработки, при условии не изменения объектом класса, но при допуске к изменению в яркостном распределении изображения [10], [25].

Метод Виолы-Джонса основан на примитивах Хаара, единственное, что меняется способ нахождения признака разграничения областей объектов, в основе метода лежит формула:

$$R = S_{\bar{б}} - N_{\bar{ч}}, \quad (5)$$

где  $N$  – суммарное значение яркостей точек закрытых темной частью;

$S$  – суммарное значение яркостей точек закрытых светлой частью.

Так что видно, что благодаря использованию примитивов Хаара идет

поточечное разделение изображения по осям. То есть объекты соотносятся друг к другу по яркости соседних точек между объектами.

#### **1.4 Гистограмма направленных градиентов и метод опорных векторов**

«Гистограмма направленных градиентов — набор особых точек, которые используются в задаче обработке изображений и иных задачах компьютерного зрения, с целью распознавания и локализации объектов» [4]. Дополняю понятие, проведенное выше надо сказать что, гистограмма направленных градиентов является переведенным понятием с английского Histogram of Oriented Gradients (HOG), [16]. Поэтому, в моменты, когда в этой выпускной квалификационной упоминается аббревиатура HOG, будет иметься в виду гистограмма направленных градиентов. «Гистограмма направленных градиентов основана на подсчете количества направлений градиента в определенных областях изображения» [19].

Так же стоит описать характерные особенности метода, сделаем это с помощью цитаты из оригинального источника. «Метод, реализуемый HOG схож с другими, но характерной особенностью является то, что при нахождении гистограммы значения равномерно вычисляется на плотной сетке 12 распределенных ячеек и использует нормализацию перекрывающегося локального контраста для увеличения точности» [16].

Авторы метода в своей работе описали алгоритм нахождения пешеходов на статичных изображениях [16], но к нынешнему времени алгоритм был расширен и стал также удобен в распознавание людей и различных других объектов в видеопотоке.

В своей работе авторы предположили, что внешний вид и форма объекта на определенном участке изображения можно описать с использованием градиентов интенсивности (направлением интенсивности контуров). Для получения данных дескрипторов необходимо разделить

изображение на маленькие связанные друг с другом ячейки и рассчитать HOG в каждой из них. Скомбинировав получившиеся диаграммы, получается дескриптор (шаблон), который в дальнейшем можно использовать для локализации. Для получения наибольшей точности было принято решение использовать не одно изображение, а несколько изображений с общим объектом на каждом из них (тот, который необходимо локализовать), такое объединение принято называть блоком. Также для увеличения точности локальные гистограммы нормализуют по контрасту, что позволяет получить большую инвариантность к освещению и другим шумам [16].

Дескриптор HOG обладает рядом преимуществ, по сравнению с другими дескрипторами. Так как HOG работает локально (для каждой ячейки), то метод не зависит от геометрических и фотометрических преобразований объекта, и позволяет обнаружить его в любом случае (в любом месте), однако метод чувствителен к изменению ориентации объекта. Во многих детекторах объектов первым шагом является нормализация по цвету, гамма-коррекция и иные маски, однако для дескриптора HOG в этом нет необходимости, так как последующая нормализация в ячейках приводит к тому же результату. «В общем виде реализация детектора HOG состоит из следующих этапов: вычисление градиента, группировка направлений, блоки дескрипторов, нормализация блоков, SVM-классификатор», [18].

На первом шаге формирования детектора HOG в каждой ячейке рассчитывается значение градиентов. Наиболее распространённым методом расчета градиента является использование одномерной дифференцирующей маски в различных положениях. «Для работы данного метода требуется произвести цветовую или яркостную фильтрацию с использованием следующих ядер:  $-1, 0, 1$  и  $-1, 0, 1^T$ »

На втором шаге вычисляются локальные гистограммы. Для каналов гистограммы направлений на основании каждого пикселя в ячейке проводится взвешенное голосование, основанное на значении градиентов. «Ячейки могут быть как прямоугольной, так и круглой формы, каналы

гистограммы распределяют равномерно либо от 0 до 180, либо от 0 до 360 градусов, в зависимости от того, какой градиент вычисляется» [4].

Третьим шагом является формирование блоков дескрипторов. Для учета изменений яркости и контрастности градиенты необходимо нормировать, а именно объединить в более крупные связанные блоки. В общем виде, «дескриптор HOG – вектор, состоящий из нормированных гистограмм ячеек» [4]. Как правило, каждая ячейка входит более чем в один конечный дескриптор. В практике применяются два основных типа блока – R-HOG (прямоугольные) и C-HOG (круглые), однако из них чаще всего применяются именно квадратные, в нашем случае давайте будет рассмотрены блоки R-HOG, приведу определение цитатой из источника далее. «Блоки R-HOG – квадратные сетки, которые можно охарактеризовать тремя ключевыми параметрами: количеством ячеек, количеством пикселей на ячейку, количеством каналов на 14 гистограмму». В оригинальном источнике авторы метода отметили оптимальными блоки размером 16 на 16, ячейки 8 на 8, при 9 каналах на гистограмму [4], [14], [18].

На четвертом шаге производится нормализация блоков. Авторы в своей работе исследовали четыре следующих способа нормализации:

- L2-норма;
- L2-hys: L2-норма ограничивается сверху (значения  $v > 0,2$ , приравниваются 0,2) и перенормируются;
- L1-норма;
- корень из L1-нормы.

Опытным путем авторы было установлено, что «L1-норма дает менее надежный результат, чем остальные три, в то время как L2-норма, L2-hys и корень из L1-нормы приводят приблизительно к одинаковым хорошим результатам» [4].

Конечным шагом в локализации объектов с использованием HOG является классификация полученных дескрипторов при помощи алгоритмов

машинного обучения. Наиболее распространённым методом для данной задачи в связке с HOG является метод опорных векторов [4], [18].

«Метод опорных векторов — набор алгоритмов обучения с учителем, использующихся для решения задач классификации и регрессионного анализа» [2]. Дополняю понятие, проведенное выше надо сказать что, метод опорных векторов является переведенным понятием с английского *support vector machine (SVM)*. Поэтому, в моменты, когда в этой выпускной квалификационной упоминается аббревиатура SVM, будет, имеется в виду метод опорных векторов.

Принцип работы метода удобно рассмотреть на примере. Пусть даны точки на плоскости, которые разделены на два класса А и Б.

Между классами проводится разделяющая прямая. Новые точки классифицируются на основе их положения относительно разделяющей прямой. Это продемонстрировано на рисунке 2, в виде графика.

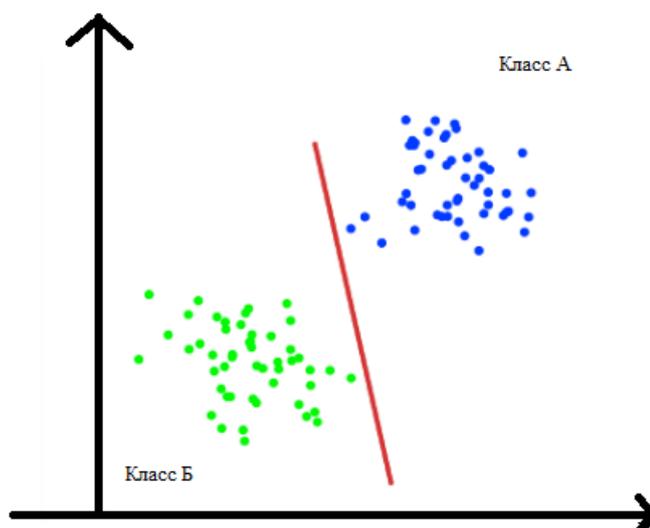


Рисунок 2 – SVM для двумерной задачи

Более сложные примеры, чем и является классификация HOG дескрипторов, имеют размерность большую, чем два, поэтому простая прямая уже не будет разделять множества на классы. Поэтому для многомерных задач рассматривается разделение гиперплоскостями. «Гиперплоскость – пространство, размерность которого на единицу меньше

исходного пространства, в котором оно находится» [4]. Из определения легко понять, что для трехмерной задачи гиперплоскостью будет обычная двумерная плоскость. Для разделения классов можно провести несколько различных гиперплоскостей, удовлетворяющих условию, однако, с точки зрения задачи классификации, лучше выбрать прямую находящуюся на максимальным расстоянием до каждого класса. На рисунке 3 рассмотрим пример для выбора разделяющей прямой.

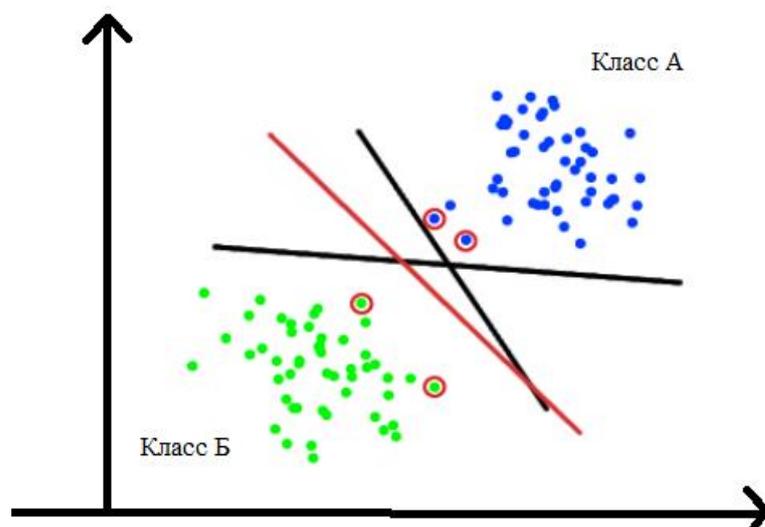


Рисунок 3 – Поиск опорных векторов

Обведенные вектора на рисунке 3 находятся ближе всех к разделяющей гиперплоскости и называются опорными векторами, на основе этого и было дано название методу. Опорные вектора на рисунке 3 обведены.

Делая выводы по главе получаем, что для решения задачи системы распознавания лиц нужно последовательно решить три этапа. Для того чтобы это сделать можно использовать множество с методов.

## **Глава 2 Анализ задачи идентификации людей по биометрическим данным лица**

### **2.1 Постановка задачи**

В данной выпускной квалификационной работе будет решаться задача идентификации людей по биометрическим данным лица. Для решения задачи необходимо идентифицировать человека по биометрическим признакам лица на входном изображении. Чтобы идентифицировать человека нужно, чтобы имелась заранее подготовленная база данных из изображений с людьми, которых нужно идентифицировать на входном изображении. На выходе должно выводиться обработанное входное изображение, на котором выделено лицо человека и имеется подпись с фамилией этого человека, или если на входном изображении не имеется человек из базы данных, то должна выводиться надпись, предупреждающая о том, что данного человека нет в базе данных.

Нашу задачу идентификации людей по биометрическим данным лица можно свести к классификации. Нужно будет на изображении найти лицо, а также его биометрические признаки в виде положения глаз. Этот биометрический признак позволяет более точно идентифицировать человека.

### **2.2 Работа алгоритма Виолы-Джонса**

В первую очередь нужно локализовать расположение лица на изображении. Как ранее было рассмотрено для этого существует несколько способов, в этой выпускной квалификационной работе будет использован метод Виолы-Джонса, использующий каскады Хаара. В сравнение с другими методами, метод Виолы-Джонса достаточно точен и быстр для решения поставленной задачи данной выпускной квалификационной работы.

Для более точной локализации лица на изображение, входное изображение нужно преобразовать в серый цвет. На рисунке 4 представлено преобразованное в серый цвет входное изображение.



Рисунок 4 – Преобразованное входное изображение

Далее с помощью каскадов Хаара произведем локализацию лица на изображение. Это необходимо для более точного нахождения такого биометрического признака как расположение глаз, а также для более точной идентификации лица. На рисунке 5 изображено локализованное лицо из преобразованного входного изображения.



Рисунок 5 – Локализованное лицо

Как видно на рисунке 5 лицо было локализовано достаточно точно. Что поможет в локализации положения глаз.

Следующим шагом будет локализация положения глаз. Положение глаз является одним из биометрических признаков лица человека, и именно поэтому в данной выпускной квалификационной работе положение глаз будет использоваться в качестве биометрического признака лица человека, для лучшей идентификации человека [1], [20]. На рисунке 6 представлено изображение, на котором уже выделены локализованные лицо и глаза.



Рисунок 6 – Локализация глаз и лица

Следующим шагом следует тренировка распознавателя по собранным признакам для самой идентификации человека. Распознаватель обучается на предоставленной ему базе данных людей состоящих из изображений с людьми, которых и нужно идентифицировать. С помощью обученный ранее распознаватель при использовании входного изображения происходит распознавание этого изображение и сравнение с изображениями, которыми обучили распознаватель ранее, и при достаточном совпадении признаков на изображении выводиться id, которое в нашем случае является фамилией,

человека. В случае если совпадения не произошло выводится надпись обозначающая, что данного человека нет в базе данных.



Рисунок 7 – Оpoznанное изображение

На рисунке 7 представлен пример того, если входное изображение распознается распознавателем. Распознаватель идентифицировал человека на изображении и поэтому над выделенным лицом имеется надпись с фамилией человека на изображении. Пример же того если человека нет в базе данных, будет рассмотрен в третий главе во время тестирования.

### 2.3 Обоснование выбранного метода

Общий вид задачи можно разделить на три этапа. Первый этап это этап распознавания объектов на изображении в случае задачи идентификации распознавание лица. Вторым этапом является векторизация изображения, то есть перевод изображения в вектор. Третьим этапом является сравнение полученных векторов.

Если со сравнением векторов все понятно, самый известный и всеми используется формула расчета Евклидова расстояния, то с первыми двумя этапами сложнее [6], [7].

$$d_2(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2}, \quad (6)$$

где  $p$  и  $q$  – это сравниваемые вектора.

Для первого этапа сравним HOG с другими методами локализации, такими как вейвлеты Хаара (использующиеся в методе Виолы-Джонса) и контексты формы представлен в оригинальной работе [14], где авторы проводили тестирование своего алгоритма на двух наборах данных с изображениями людей, а именно MIT и INRIA [21].

Таблица 1 – Сравнительный анализ методов локализации изображений

Дескриптор	Набор данных	Доля пропущенных изображений	Доля ошибок первого рода
HOG	MIT	$\approx 0$	0,0001
HOG	INRIA	0.1	0,0001
Вейвлеты Хаара	MIT	0.01	0,0001
Вейвлеты Хаара	INRIA	0.3	0,0001
PCA-SIFT, контексты формы	MIT	0.1	0,0001
PCA-SIFT, контексты формы	INRIA	0.5	0,00001

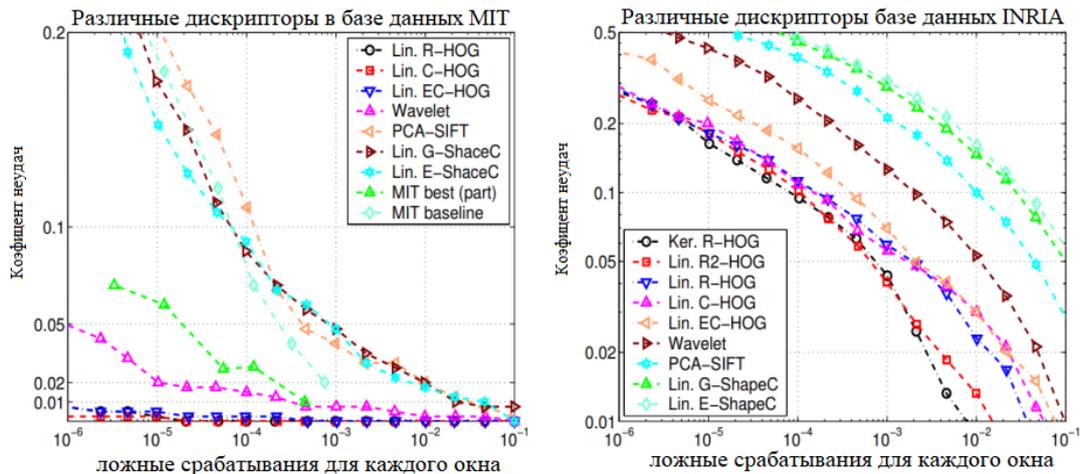


Рисунок 8 – График сравнительного анализа

Как видно из таблицы 1 и рисунка 8, вейвлеты Хаара показали сравнительно отличные результаты, на основе базы данных людей, поэтому метод Виолы-Джонса, основанный на вейвлетах Хаара, является оптимальным для данной задачи.

Не малым преимуществом дескрипторов Виолы-Джонса является известность, доступность и наличие инструментов для реализации, которые упрощают разработку и написание кода, например, реализация дескрипторов Виолы-Джонса с помощью библиотеке OpenCV [15].

Для решения задачи классификации сегодня наиболее актуальны методы, основанные на машинном обучении, так как они показывают наилучшие результаты. Именно поэтому в этой работе оно и будет использовано. Выделяют следующие преимущества метода Виолы-Джонса, на основе которых было принято решение, использовать именно данную архитектуру для задачи классификации лица:

- возможность реализации параллельных вычислений (на отдельных слоях), а, следовательно, возможность реализации обучения на графических процессорах,
- достаточная точность работы метода,

- хорошая устойчивость к деформациям и сдвигам распознаваемого изображения;

Описанные преимущества Виолы-Джонса делают данный метод одним из часто используемых и практичных для задачи распознавания и классификации изображений, поэтому был сделан выбор в пользу такого подхода [2].

В этой главе определились, что в данной выпускной квалификационной работе, в качестве разработанного алгоритма идентификации лиц по биометрическим данным лица будет выступать заранее построенная и обученный алгоритм Виолы-Джонса.

## **Глава 3 Разработка алгоритма идентификации людей по биометрическим данным лица**

### **3.1 Среда разработки и язык**

В качестве высокого языка программирования для написания программного кода в данной выпускной квалификационной работе будет использован язык Python. Он является крайне удобным для решения задачи данной выпускной квалификационной работы, так как имеет множество специализированных библиотек облегчающие реализацию алгоритмов необходимых для решения задач классификации, и в частности для решения задачи идентификации людей по биометрическим данным лица в данной работе [13].

В качестве среды разработки будет использован сервис Google Colaboratoty или сокращенно Google Colab. Данная среда позволяет любому писать и выполнять произвольный код на языке Python через браузер и особенно хорошо подходит для машинного обучения, анализа данных и обучения. Google Colab — это размещенная на хостинге служба ноутбуков, которая не требует настройки для использования, но при этом предоставляет бесплатный доступ к вычислительным ресурсам, включая графические процессоры. Именно доступ к графическому процессору является крайне важным аспектом в данной работе, по который и был сделан выбор в пользу среды разработки Google Colab.

### **3.2 Локализация признаков**

Для написания алгоритма локализации будет использована библиотека OpenCV реализованный в Python. В данной библиотеке уже реализован каскад Хаара, поэтому написание работающего алгоритма не составит большого труда.

Первым делом подключаем все библиотеке нужные для реализации программы, как сказано ранее подключаем библиотеку OpenCV для Python под названием cv2. Далее на очереди библиотеки для работы с файлами и Google диском в среде Google Colab. Эти библиотеки нужны так как база данных, состоящая из изображений людей, и само входное изображения которое предназначено для идентификации на нём человека будет находиться на Google диске. Ниже будет приведен фрагмент листинга кода предназначенного для этого на рисунке 9:

```
import cv2
from google.colab.patches import cv2_imshow
from google.colab import files
from google.colab import drive
import dlib
drive.mount('/content/drive')
```

Рисунок 9 – Фрагмент кода

На рисунке 9 в приведенном коде и происходит подключение библиотек и Google диска [13].

Далее подключаем заранее взятые из библиотеки cv2 файлы с реализацией каскада Хаары. Эти файлы также располагаем на Google диски и спокойно подключаем. Листинг фрагмента кода с подключением приведён ниже на рисунке 10:

```
cascPath = "/content/gdrive/MyDrive/res/haarcascade_frontalface_default.xml"
eyePath = "/content/gdrive/MyDrive/res/haarcascade_eye.xml"
faceCascade = cv2.CascadeClassifier(cascPath)
eyeCascade = cv2.CascadeClassifier(eyePath)
```

Рисунок 10 – Фрагмент кода

На приведенном фрагменте были созданы переменные с путем к файлам с каскадом Хаара для распознания лиц и глаз. А так же объявляем классификаторы каскада Хаара с помощью этих файлов.

Далее загрузим входное изображение, и следом преобразуем его в изображение в оттенках серого [5].

Для этого будет применен просто линейное преобразование по формуле:

$$Y_{\text{gray}} = 0.2126R + 0.7152G + 0.0722B \quad (7)$$

где R, G, B – это цвета, соответственно красный, зелёный, голубой.

Эта формула уже реализована в библиотеке OpenCV, поэтому воспользуемся ей.

Еще одним важным моментом является функции чтения изображения, для того чтобы была возможность использовать изображение в программе. Листинг функции приведен на рисунке 11:

```
def read_img(path):  
    img = cv2.imread(path)  
    width = 604  
    (h,w)=img.shape[:2]  
    ratio = width / float(w)  
    height = int(h * ratio)  
    return cv2.resize(img,(width,height))
```

Рисунок 11 – Фрагмент кода

Как видно из функции на рисунке 11, для работоспособности функции нужно в функции передать всего лишь путь до нужного изображения в нашем случае это будет путь до нужного файла находящимся на Google диске.

Далее проведём листинг фрагмента кода, с реализацией объявления функции чтения, и сохранения файла в переменную, а также преобразование файла в оттенки серого на рисунке 12:

```
frame = read_img(video_capture)
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
```

Рисунок 12 – Фрагмент кода

В данном фрагменте происходит чтение изображения, с помощью рассмотренной выше функции, а также преобразование изображение в оттенки серого с помощью встроенной функции в библиотеке по формуле рассмотренной выше [12].

На рисунке 13 приведён фрагмент кода с использованием функции из `faceCascade`. `faceCascade` – это переменная, которая содержит предварительно обученный алгоритм каскада Хаара для поиска лица:

```
faces = faceCascade.detectMultiScale(
    gray,
    scaleFactor=1.16 ,
    minNeighbors=5,
    minSize=(100, 100),
    flags=cv2.CASCADE_SCALE_IMAGE
)
for (x, y, w, h) in faces:
    cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 3)
    roi_gray = gray[y:y+h, x:x+w]
    roi_color = frame[y:y+h, x:x+w]
    cv2.putText(frame, 'Face', (x, y), font, 2, (255,0,0),5)
```

Рисунок 13 – Фрагмент кода

На рисунке 13 в данном фрагменте объявляется функция из `faceCascade`, которая распознает объекты разного размера в нашем случае это лицо на изображении. Для работы функции используем преобразованное, в серые оттенки по формуле, входное изображение, также выставляем различные параметры подобранные экспериментально. Это функция и является функцией для локализации лица на изображении. Также в этом фрагменте представлен код для выделения прямоугольником распознанного лица [23].

На рисунке 14 приведён фрагмент кода с использованием функции из `eyeCascade`. `eyeCascade` – это переменная, которая содержит предварительно обученный алгоритм каскада Хаара для поиска лица:

```
eyes = eyeCascade.detectMultiScale(roi_gray, minSize=(25, 5))
for (ex,ey,ew,eh) in eyes:
    cv2.rectangle(roi_color, (ex,ey), (ex+ew,ey+eh), (0,255,0), 2)
    cv2.putText(frame, 'Eye', (x + ex,y + ey), 1, 1, (0, 255, 0), 1)
```

Рисунок 14 – Фрагмент кода

На рисунке 14 в данном фрагменте объявляется функция из `eyeCascade`, которая распознает объекты разного размера в нашем случае это лицо на изображении. Для работы функции используем преобразованное, в серые оттенки по формуле, входное изображение, также выставляем различные параметры подобранные экспериментально. Это функция и является функцией для локализации глаз на изображении. Также в этом фрагменте представлен код для выделения прямоугольником положения глаз.

### 3.3 Обучение распознавателя

Следующим этапом работы является обучения распознавателя. Обучения модели распознавателя будет происходить на основе изображений из базы данных. На рисунке 15 представлена общая схема обучения распознавателя.

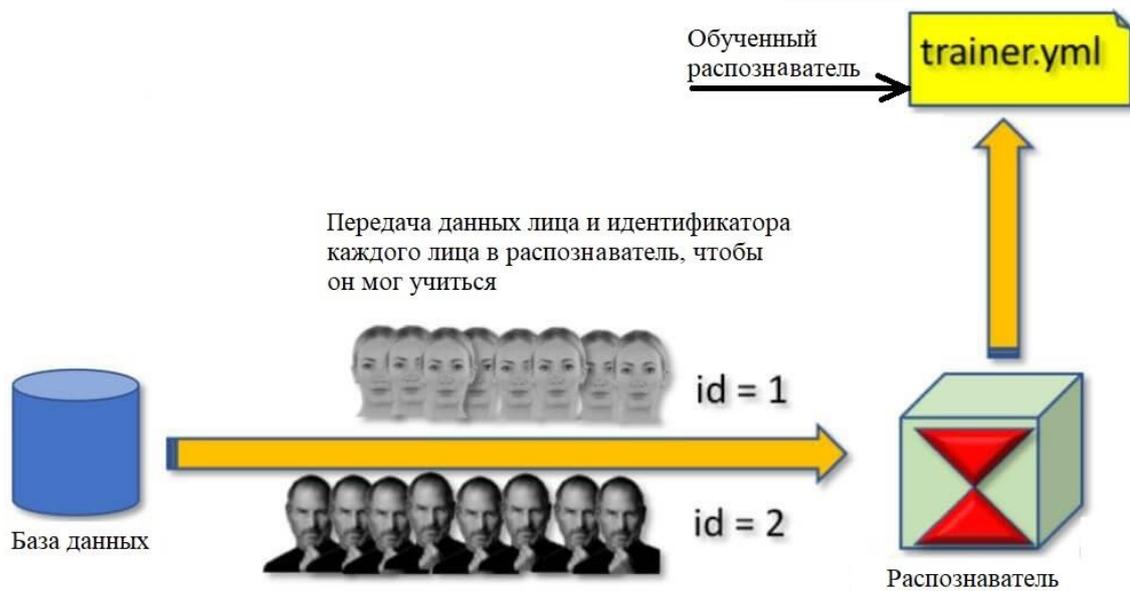


Рисунок 15 – Тренировка распознавателя

Из рисунка 15 видно, что для обучения нужно просто распознавать лица из базы данных и присваивать им некоторый идентификатор, в нашем случае индикатором будут выступать фамилии.

На рисунке 16 будет приведен листинг, предназначенный для обучения распознавателя:

```

#путь до базы данных изображений
path = 'dataset'
recognizer = cv2.face.LBPHFaceRecognizer_create()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml");
# функция чтобы получить изображения и их данные
def getImagesAndLabels(path):
    imagePath = [os.path.join(path,f) for f in os.listdir(path)]
    faceSamples=[]
    ids = []
    for imagePath in imagePath:
        PIL_img = Image.open(imagePath).convert('L')
        img_numpy = np.array(PIL_img,'uint8')
        id = int(os.path.splitext(imagePath)[-1].split(".")[1])
        faces = detector.detectMultiScale(img_numpy)
        for (x,y,w,h) in faces:
            faceSamples.append(img_numpy[y:y+h,x:x+w])
            ids.append(id)
        eyes = eyeCascade.detectMultiScale(roi_gray, minSize=(25, 5))
        for (ex,ey,ew,eh) in eyes:
            faceSamples.append(img_numpy[y:y+h,x:x+w])
            ids.append(id)
    return faceSamples,ids
faces,ids = getImagesAndLabels(path)
recognizer.train(faces, np.array(ids))
# Сохранения обученной модели в trainer/trainer.yml
recognizer.write('trainer/trainer.yml')

```

Рисунок 16 – Фрагмент кода

На рисунке 16 в приведенном выше листинге, мы можем наблюдать функцию, предназначенную для обучения распознавателя. В ней и происходит обучение модели распознавателя с сохранением одного в отдельный файл [22], [24].

### 3.4 Сбор полной программы и тестирование

Теперь у нас все готово для написания полной программы, которая предназначена решать задачу идентификации людей по биометрическому признаку лица.

Чтобы было более наглядно не будем приводить весь листинг программы, а только рассмотрен блок – схему на рисунке 17, по которой и происходила сборка программы. Данная блок – схема может быть не до конца корректна, так как является некой общей схемой, но в этом нет ничего плохого, так как все этапы написания программы будут расписаны далее.

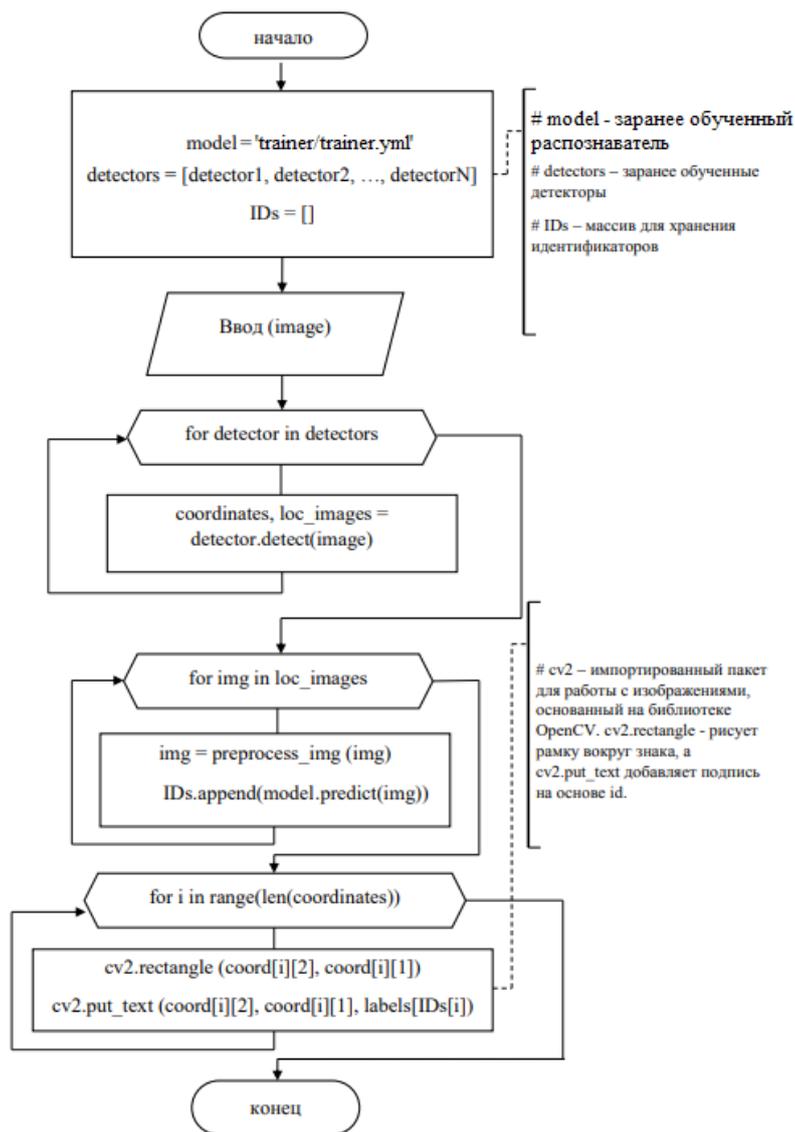


Рисунок 17 – Блок – схема

На рисунке 17 показана блок – схема, на котором первым пунктом подключаем обученную модель распознавателя. А так же заранее обученный детектор на основе каскада Хаара. Далее загружаем и читаем изображение с

помощью ранее рассмотренной функции. Вводим массив в `id` и называем элементы массива в соответствие полученных `id` в обученной модели распознавателя на базе данных.

Далее производим распознавание входного изображения с помощью ранее рассмотренных функции локализации лица и глаз. После получения данных распознавания, сравниваем данные обученного распознавателя и при достаточном совпадении выводим на экран изображение с обведённым лицом и надписью, которая является `id` совпадающего человека, и также является его фамилией.

В том же случае если совпадения нет, то выводим на экран изображение с обведённым лицом и надписью, которая означает то, что человека нет в базе данных и соответственно в обученной модели распознавателя.

Далее давайте рассмотрим схему с примерами работы программы на рисунках 18, 19 и 20. Рисунки 18 и 19 являются примером работы программы, в котором входное изображение имеет искомого человека в базе данных. Рисунок 20 наоборот является примером работы программы, в котором входное изображение не имеет искомого человека в базе

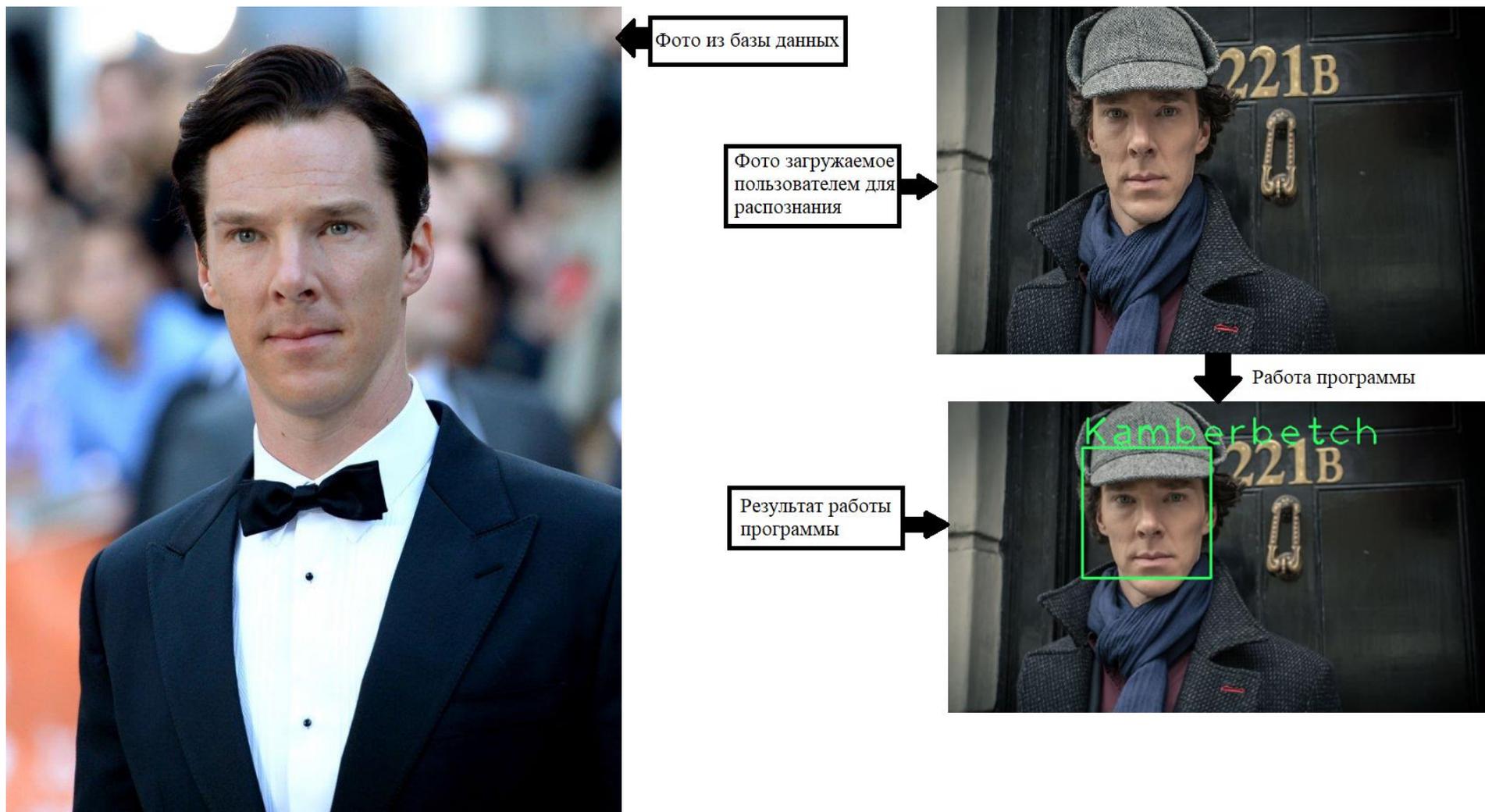


Рисунок 18 – Пример работы программы

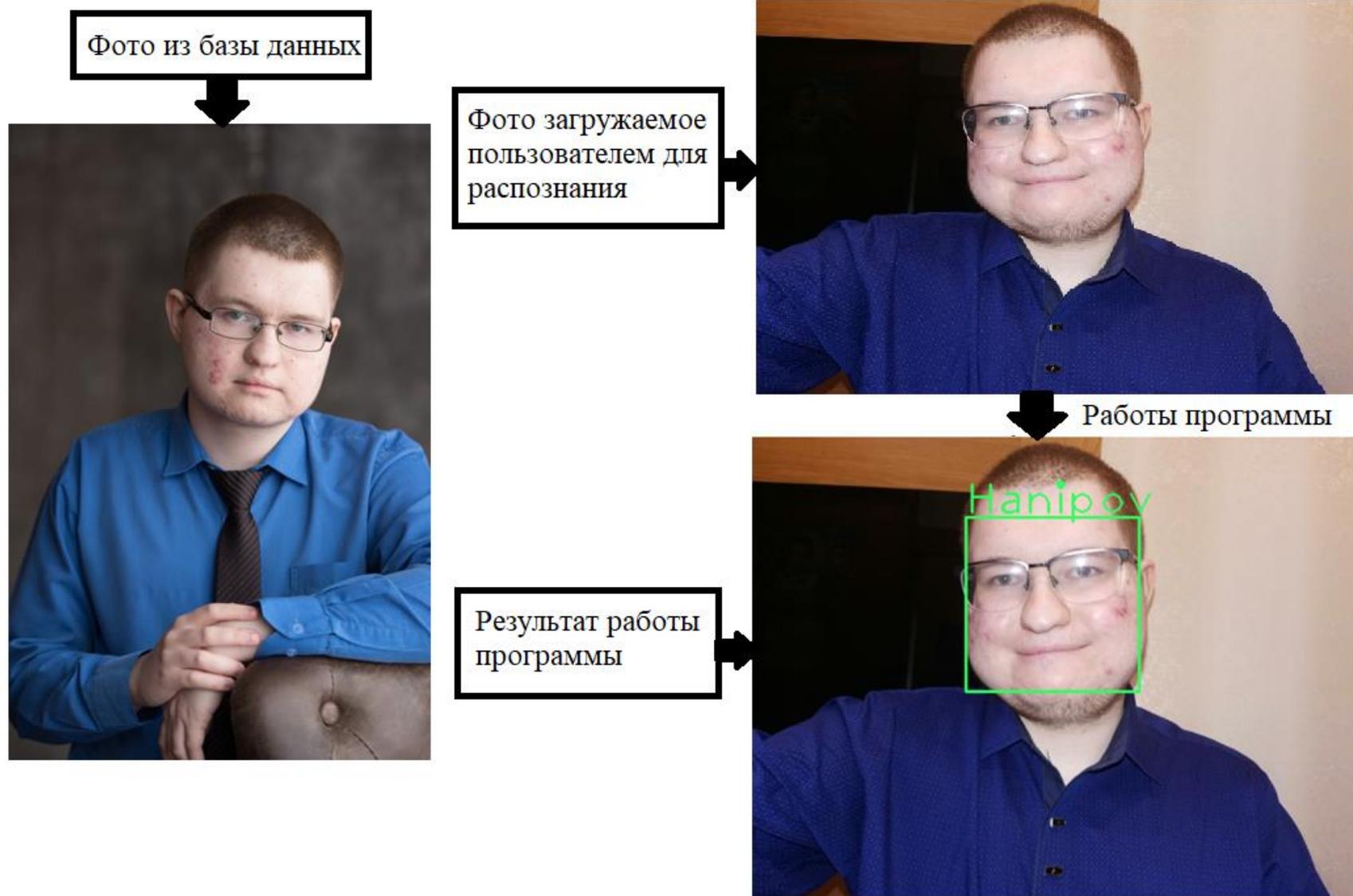


Рисунок 19 – Пример работы программы

На рисунках 18 и 19 представлены работа программы, если искомый человек имеется в базе. В этих примеров, на изображение есть человек из базы данных, поэтому программа вывела идентификатор, который является названием изображения и фамилией человека

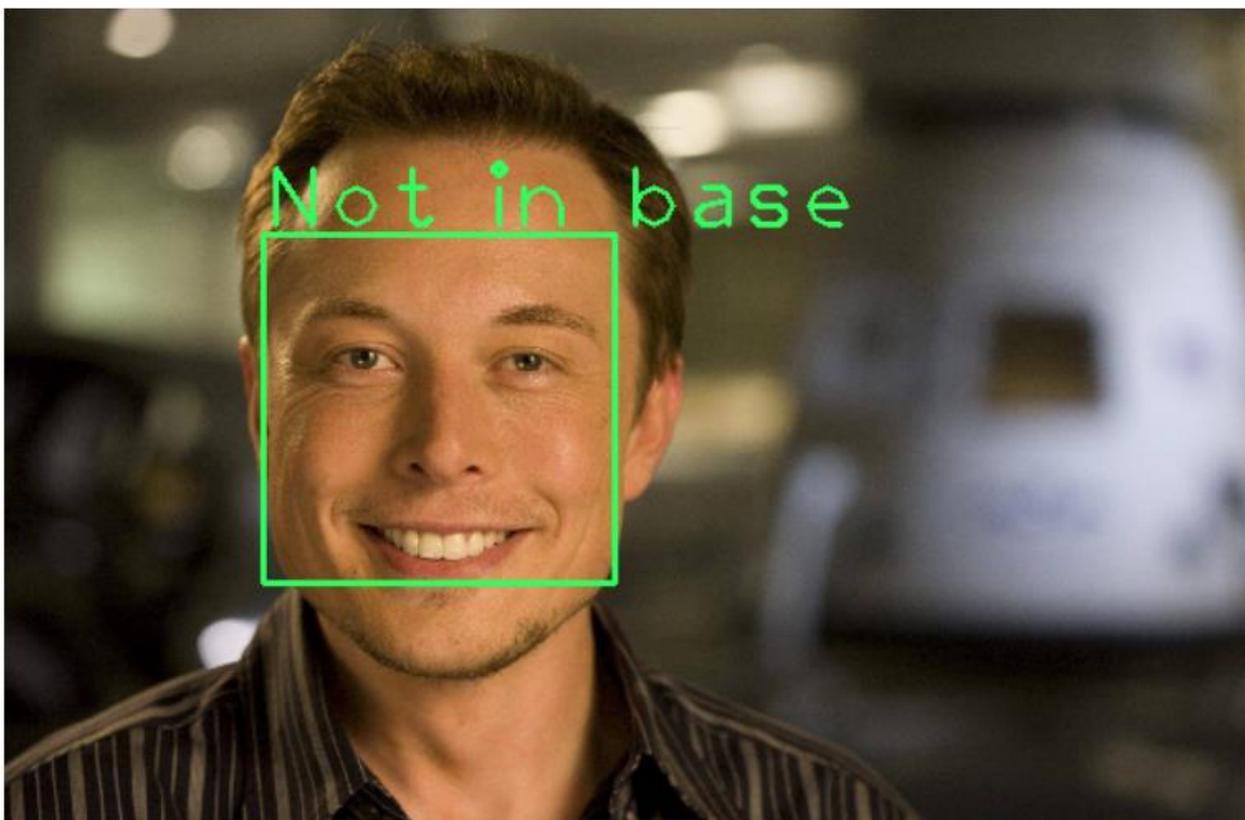


Рисунок 20 – Пример работы программы

На рисунке 20, представлен пример работы программы, если искомого человека на загруженном пользователем изображении нет в базе данных. Так как искомого человека нет в базе данных, программа вывела надпись «Not in base», что означает, что данного человека нет в базе данных.

В этой главе мной был описан ход разработке и методы, использованные в разработке алгоритма и программы для его работы пользователем. Также проведено тестирование с приведением работы всей программы.

## Заключение

В данной работе была рассмотрена история возникновения и развития технологий распознавания лиц, разобрались с общим видом задачи и какие бывают пробелы в идентификации людей по биометрическим данным лиц. Поставлена задача на разработку алгоритма идентификации людей по биометрическим данным лиц, а именно постройка и обучение с использованием метода Виолы-Джонса и каскад Хаара.

В ходе выполнения работы были рассмотрены и проанализированы методы компьютерного зрения, такие как HOG+SVM, CNN и Виолы-Джонса.

Обучена и протестирован распознаватель для идентификации людей по биометрическим данным лиц.

Для создания реализации программы была выбрана среда Google Colab, позволяющая создавать и запускать приложения на языке Python. Разработанная программа было протестировано на корректность работы.

В первой главе мы поняли, что для решения задачи системы распознавания лиц нужно последовательно решить три этапа. Одним, из которых и является разработка алгоритма идентификации лиц по биометрическим данным лица.

Во второй главе определились, что в данной выпускной квалификационной работе, в качестве разработанного алгоритма идентификации лиц по биометрическим данным лица будет выступать заранее построенная и обученный распознаватель, который использует метод Виолы – Джонса и каскады Хаара как основу.

В третий главе мной был описан ход разработке и методы, использованные в разработке алгоритма и программы для его работы пользователем. Также проведено тестирование с приведением работы всей программы.

В рамках этой выпускной квалификационной работы были выполнены все поставленные цели и задачи.

## Список используемой литературы

1. Катасёв А.С., Катасёва Д.В., Кирпичников А.П. Нейросетевая Биометрическая Система Распознавания Изображений Человеческого Лица // Вестник Технологического Университета. - . - 18. - 19. - С. 135-138.
2. Бредихин А.И. Алгоритмы обучения сверточных нейронных сетей / А.И. Бредихин // Вестник Югорского государственного университета. 2019. № 1 (52). С. 41-54.
3. Вакуленко С.А., Жихарева А.А. Практический курс по нейронным сетям – СПб: Университет ИТМО, 2018. – 71 с
4. Википедия [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/Гистограмма\\_направленных\\_градиентов](https://ru.wikipedia.org/wiki/Гистограмма_направленных_градиентов) – Гистограмма направленных градиентов – (дата обращения 10.03.2022).
5. Визильтер Ю.В. Обработка и анализ изображений в задачах машинного зрения / Ю.В. Визильтер. –М.: Физматкнига, – 2010. – 689с.
6. Галимянов Ф.А. Сравнительный анализ алгоритмов реализации метода обратного распространения ошибки для обучения нейронных сетей / Ф.А. Галимянов // Научно-технический вестник Поволжья. 2020. № 2. С. 69-71.
7. Гашников, М.В. Методы компьютерной обработки изображений. 2–е изд. / М.Б. Гашников и др. / под ред. Соифера В.А. — М. : ФИЗМАТЛИТ, 2003. — 784 с
8. Гонсалес Р. Цифровая обработка изображений в среде MATLAB / Р. Гонсалес, Р. Вудс, С. Эддинс. –М.: Техносфера, 2006. - 616 с.
9. Гузий Е.А. Сверточная нейронная сеть для разработки системы распознавания и классификации изображений / Е.А. Гузий, В.В. Федоренко // Молодежный научно-технический вестник. 2017. № 7. С. 51.
10. Ле Мань Ха Свёрточная нейронная сеть для решения задачи классификации / Ле Мань Ха // Труды МФТИ. 2016. №3 (31). С. 56-62.
11. Паклин Н.Б., Орешков В.И. Бизнес-аналитика: от данных к

знаниям: учебное пособие. – 2-е изд., испр. – СПб.: Питер, 2013. – 704 с.: ил.

12. Старовойтов В.В. Цифровые изображения: от получения до обработки/ В.В. Старовойтов, Ю.И. Голуб – Минск: ОИПИ НАН Беларуси, 2014 – 202 с. - ISBN 978-985-6744-80-1.

13. Федоров Д. Ю. Основы программирования на примере языка Python: учеб.пособие / Д. Ю. Федоров. – СПб., 2016. –176 с.

14. Фу К. Структурные методы в распознавании образов: Пер. с англ. – М.: Мир, 1977. – 320 с.

15. Paul V., Michael J. J. Robust Real-Time Face Detection[текст] / V. Paul, J. J. Michael // Kluwer Academic Publishers. 2004. – 18 p.

16. Bradsky, G. Learning OpenCV: Computer Vision with OpenCV Library / G. Bradsky, A. Kaehler. – O'Reilly Media, 2008. – 555 p

17. Histograms of Oriented Gradients for Human Detection Navneet Dalal and Bill Triggs INRIA Rhone-Alps,

18. Katasev A.S. The methods and instruments of Data Mining in tasks of technological equipment faults and damages recognition / 7th International Conference on Pattern Recognition and Image Analysis: New Information Technologies (PRIA-7-2004). – St. Petersburg, 2004. – pp. 720-723.

19. Lu, H. Calculate Deep Convolution Neural Network on Cell Unit/ Haofang Lu, Ying Zhou, Zi-Ke Zhang – Springer Singapore, 2017 – 526 p

20. Pablo N. Benchmarking Haar and Histograms of Oriented Gradients Features Applied to Vehicle Detection[текст] / N. Pablo // Universite Pierre et Marie Curie-Paris. 2007. – 6 p.

21. Rowley, H.A. Neural network-based face detection / H.A. Rowley, S. Baluja, T. Kanade // IEEE Trans. Pattern Anal. Mach. Intell. — 1998. — Vol. 20. — № 1 – 156 p.

22. Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In Advances in neural information processing systems, pages 2377–2385, 2015.

23. Szeliski, R. Computer Vision: Algorithms and Applications / R. Szeliski. – Springer, 2011. - 812 p

24. TensorFlow [Электронный ресурс]. - URL: [https://www.tensorflow.org/hub/tutorials/tf2\\_image\\_retraining](https://www.tensorflow.org/hub/tutorials/tf2_image_retraining) (Дата обращения: 15.04.2022).

25. Viola P., Jones M. Rapid Object Detection using a Boosted Cascade of Simple Features[текст] / P. Viola, M. Jones // Accepted Conference on Computer Vision and Pattern Recognition. – 2001. – 9 p.