

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)

02.03.03 Математическое обеспечение и администрирование информационных систем
(код и наименование направления подготовки, специальности)

Технология программирования
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Моделирование портрета человека с использованием нейронной сети»

Студент

В.А. Талицких
(И.О. Фамилия)

(личная подпись)

Руководитель

к.т.н, доцент, Э.В. Егорова
(ученая степень, звание, И.О. Фамилия)

Тольятти 2020

Аннотация

Темой выпускной квалификационной работы является «Моделирование портрета человека с использованием нейронной сети».

Объектом исследования является процесс моделирование лица человека и его распознавание. В исследовании рассматриваются задачи разработки алгоритма указанного процесса и разработка его программного продукта.

Цель исследования – разработка авторского алгоритма моделирования портрета человека с использованием нейронной сети.

Для достижения цели были поставлены следующие задачи:

- проанализировать существующие алгоритмы распознавания лица;
- выявить недостатки существующих алгоритмов;
- сформировать требования к алгоритму моделирования человека;
- разработать алгоритм моделирования портрета человека с использованием нейронной сети;
- протестировать алгоритм на реальных примерах.

Структура выпускной квалификационной работы представлена введением, тремя разделами, заключением, списком литературы.

В первом разделе проводится обзор различных методов решения данного алгоритма, а также теоретические сведения об нейронных сетях и распознавания лиц.

Во втором разделе описывается задача реализации программного обеспечения по разработанному алгоритму, рассматривая различные технологии моделирования человека.

В третьем разделе производится тестирование разработанного продукта в различных условиях работы, и формируются результаты проделанного тестирования.

Данная бакалаврская работа состоит из пояснительной записки на 53 стр., включая 27 рисунков, списка из 30 источников, как на иностранном языке, так и на русском, включая и электронные ресурсы, а также приложение.

Abstract

The title of the graduation work is "Modeling a portrait of a person using a neural network."

The structure of graduation work contains an introduction, three chapters, conclusion and list of references.

The graduation work consists of an explanatory note on 53 pages, introduction, including 27 figures, 1 table, the list of 30 references including 10 foreign sources and 1 application.

The key issue of the graduation work is modeling a portrait of a person using a neural network.

The object of the research is the process of modeling a person's face and its recognition. The study discusses the tasks of developing an algorithm for this process and developing a software implementation of the developed algorithm in the Python programming language.

The introduction describes the relevance of the study.

The first chapter reviews various methods for solving this algorithm, as well as theoretical information about neural networks and face recognition.

The second chapter describes the task of implementing software according to the developed algorithm, considering various technologies for human modeling, identification and recognition of user faces.

In the third chapter, the developed product is tested in various working conditions, and the results of the testing are formed.

The final part of the graduation work gives detailed information about testing the developed product. The application of the developed algorithm to images of various sizes is described.

Содержание

Введение.....	5
1 Анализ области применения алгоритма моделирование портрета человека с использованием нейронной сети	7
1.1 Описание работы алгоритмов моделирования портрета человека с использованием нейронной сети.....	7
1.2 Формализация требований к создаваемому алгоритму моделирование портрета человека с использованием нейронной сети.....	14
2 Проектирование алгоритма моделирование портрета человека при помощи нейронной сети	18
2.1 Составление структуры разрабатываемого алгоритма моделирование портрета человека с использованием нейронной сети.....	18
2.2 Выбор инструмента разработки программы на основе составленной модели	21
2.3 Реализация программного продукта моделирования портрета человека при помощи нейронной сети.....	34
3 Тестирование разработанного алгоритма моделирование портрета человека с использованием нейронной сети	40
3.1 Тестирование программного продукта на основе разработанного алгоритма моделирование портрета человека с использованием нейронной сети	40
3.2 Проведение сравнительного анализа для определения эффективности работы программного продукта	47
Заключение	52
Список используемых источников.....	54
Приложение А Исходный код разработанного программного обеспечения.....	58

Введение

В современном мире технология распознавания лиц является актуальной, как никогда. Практически везде используется эта технология, например, при помощи идентификации лица в Китае оплачивают покупки, правоохранительные органы используют данную технологию для нахождения преступников, в некоторых магазинах технология распознавания лиц может выявлять тех, кто ранее воровал продукты с прилавков, даже можно выявить заболевания у человека.

Возможность распознавания лица осуществляется при помощи машинного обучения, искусственного интеллекта или нейронной сети.

Актуальность данной работы состоит в разработке алгоритма моделирования портрета человека с использованием нейронной сети.

Новизна бакалаврской работы заключается в новом алгоритме моделирования портрета человека с использованием нейронной сети.

Практическая ценность состоит в разработке алгоритма для моделирования портрета человека с использованием нейронной сети.

Объект исследования – процесс функционирования алгоритмов моделирования портрета человека с использованием нейронной сети.

Предмет исследования – алгоритмы моделирования портрета человека с использованием нейронной сети.

Цель исследования – разработка алгоритма моделирования портрета человека с использованием нейронной сети.

Для достижения цели были поставлены следующие задачи:

- проанализировать существующие алгоритмы моделирования человека;
- выявить недостатки существующих алгоритмов;
- Сформировать требования к алгоритму моделирования человека;
- Разработать алгоритм моделирования портрета человека с использованием нейронной сети;

– протестировать алгоритм на реальных примерах.

Бакалаврская работа состоит из введения, трех глав, заключения, списка использованных источников и приложения.

В первой главе рассматриваются процессы работы алгоритма моделирования портрета человека с использованием нейронной сети, а также способы и сферы применения данного алгоритма. На основании этих данных будет формулироваться задача и требования к ее решению.

Во второй главе будет рассматриваться сама архитектура, которая была выбрана для решения задачи, а также средства реализации и описываются причины, по которым они выбраны.

В третьей главе предоставляются результаты тестирования и работы данного алгоритма на реальных примерах.

В заключении подводятся итоги исследования, создаются окончательные выводы и описываются результаты выполненной работы.

1 Анализ области применения алгоритма моделирование портрета человека с использованием нейронной сети

1.1 Описание работы алгоритмов моделирования портрета человека с использованием нейронной сети

Тема «моделирование портрета человека» или же «распознавание лица» является одной из интересных, важных, и даже ключевых задач в современном мире, как с теоретической, так и с практической точек зрения.

В современном мире в системе распознавания лиц нуждаются практически все сферы организации общественной жизни:

- банки;
- правоохранительные органы;
- предприятия;
- магазины;
- различные государственные структуры и объекты;
- образовательные учреждения.

Распознавание и идентификация людей помогает правоохранительным органам, обеспечивая помощь в нахождении преступников среди населения и помогает при их задержании, отслеживая местоположение потенциальных нарушителей закона.

Благодаря этой технологии проводить денежные операции стало намного проще, так как не нужно запоминать кодовые слова, носить с собой документы, достаточно лишь посмотреть в камеру, система идентифицирует клиента и откроет доступ к его личному кабинету, где он может проводить денежные операции.

Различные предприятия уже сегодня могут отслеживать нахождение своего персонала на рабочем месте, а также фиксируется время прихода и ухода сотрудника, что способствует упразднению рутинных бумажных отчетов от каждого персонала. Подобный сервис тестировала сеть салонов «Эконика», где сервис распознавал лица сотрудников, которые приходили на работу.

Система показала себя превосходно, так как, посмотрев на камеру планшета, система распознавала сотрудника за 2-3 секунды.

В магазинах используется эта система для выявления потенциальных неадекватных людей, которые могут причинить вред товарам, а также анализирует поведения покупателя для повышения лояльности к сети магазинов. Например, в 2017 году продуктовый магазин «Walmart» начал использовать технологию для определения, насколько покупатель доволен обслуживанием и посещением магазина. Если система замечает недовольного клиента, она подает сигнал сотрудникам, которые постараются помочь этому человеку.

Сегодня даже большинство электронных устройств дает возможность получать доступ системе при распознавании лица владельца, таким образом, обеспечивая сохранность данных владельца от посторонних людей.

Конечно, существуют и отрицательные стороны данной темы. На сегодняшний день систему распознавания лиц могут использовать для получения личной выгоды или с целью проникновению в личную жизнь. К сожалению, можно взломать или купить базу клиентов банков, где содержится вся их биометрия, которую можно использовать в противозаконных целях.

Также можно использовать сервисы по нахождению информации о человеке вплоть до его страниц в социальных сетях, номеров телефона и домашнего адреса при помощи одной фотографии.

Хотя на сегодняшний день существует множество сервисов для распознавания лиц, в нашей стране данный вопрос только лишь набирает обороты и является актуальной темой для исследования.

Поставленную задачу «моделирование портрета человека» можно отнести к области нейронных сетей и компьютерного зрения. Эти направления довольно тесно связаны и имеют в основном, общие решения.

На первый взгляд кажется, что в данной теме нет ничего сложного, однако нужно понимать определения таких понятий, как нейронная сеть, машинное обучение, понимать принцип работы алгоритма распознаваний лица.

Нейронная сеть – это последовательность нейронов, которые соединены между собой синапсами. Нейронная сеть построена по принципу организации и функционирования биологических нейронных сетей. Нейронную сеть невозможно запрограммировать, ее, как и человеческий мозг можно обучать. Другими словами, нейронная сеть - это интерпретация мозга человека в машинном коде, где хранится и передается информация. На рисунке 1.1 демонстрируется нейрон в человеческом мозге [8].

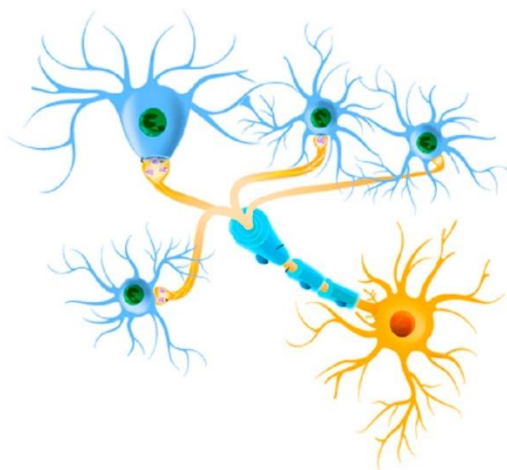


Рисунок 1.1 – Нейрон в человеческом мозгу

Нейронные сети используются для решения сложных задач, которые требуют аналитических вычислений подобных тем, что делает человеческий мозг. На рисунке 1.2 показан пример того, как выглядит нейронная сеть [14].

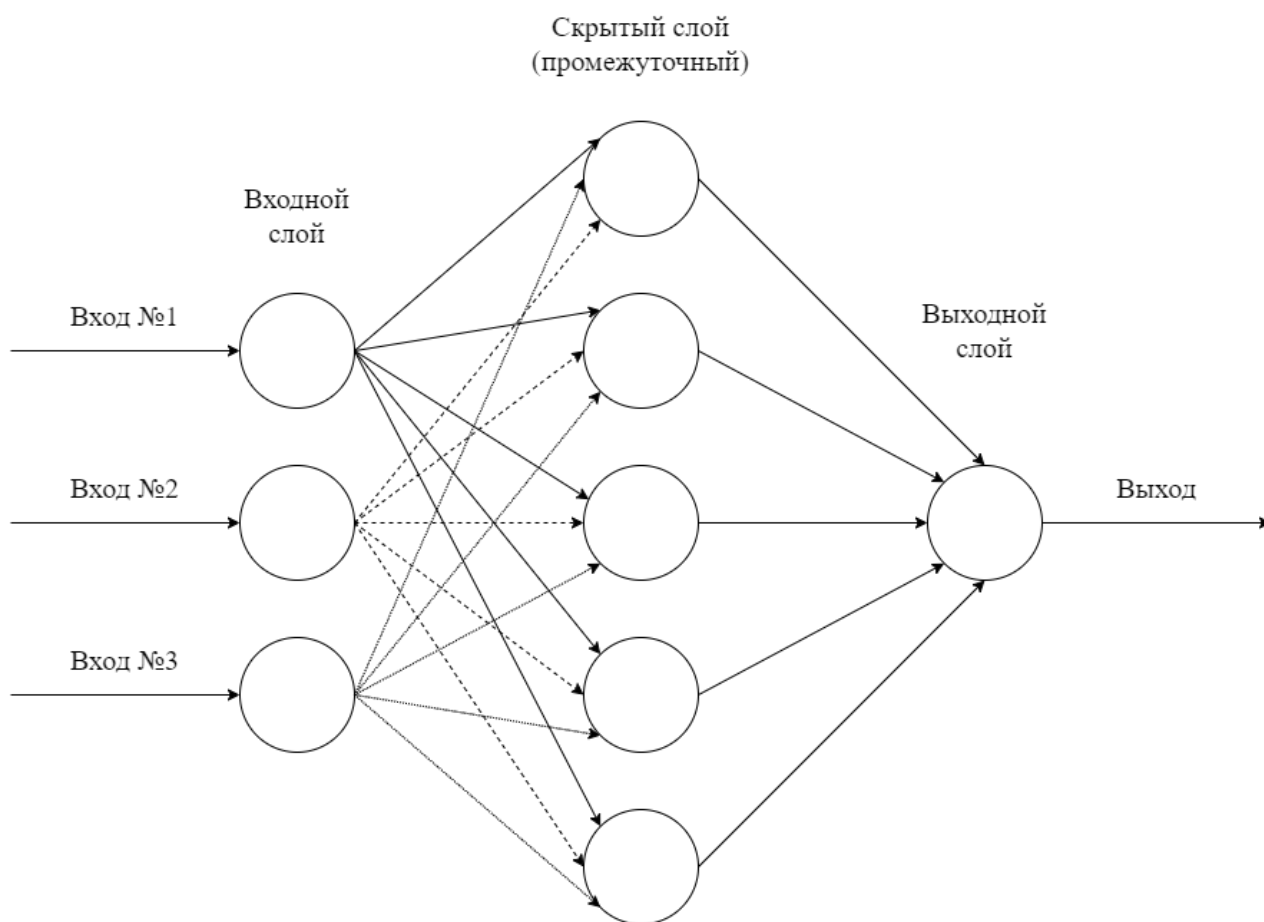


Рисунок 1.2 – Демонстрация работы нейронной сети

Если сравнить структуру нейрона человеческого мозга с рисунка 1.1 и структуру искусственной нейронной сети, показанной на рисунке 1.2, то можно увидеть сходство между ними [5].

Одними из существующих применений нейронных сетей являются:

- классификация распределяет данные по параметрам, например, кто подходит по возрасту для участия в конкурсе, а кто нет. У кого хорошая кредитная история и ему можно выдать кредит, а кто не подходит по этому параметру;
- предсказание, которое предугадывает следующий шаг. Например, какие акции упадут в цене, а какие наоборот возрастут;
- распознавание – является самым распространяемым применением нейронной сети. Используется как в распознавании объектов на фотографиях, так и в распознавании лиц.

Машинное обучение – подраздел искусственного интеллекта, который изучает методы построения алгоритмов и систем, результаты которых накапливаются по мере приобретения опыта. Более подробно приведено в работах [24] – [27].

Машинное обучение делится на два типа [3]:

- обучение по прецедентам или же индуктивное обучение, которое основывается на выявлении общих закономерностях эмпирических данных.

- дедуктивное обучение, которое основывается на данных предоставляемыми экспертами в виде базы данных.

Дедуктивное обучение относят к области экспертных систем, в связи с этим в области машинного обучения принято считать обучение по прецедентам синонимом машинного обучения.

Термин машинного обучения появился в результате отделения науки о искусственном интеллекте и на методы обучения сетей и виды архитектур сетей, а также машинное обучение впитало в себя математическую статистику, еще методы оптимизации и классические математические дисциплины. В свою очередь, машинное обучение имеет и свою собственную специфику, которая связана с проблемой вычислительной эффективностью [2].

Машинное обучение делится на следующие типы поставленных задач:

- обучение с учителем – представлен в виде «задача, ответ»;

- обучение без учителя – система ищет самостоятельно сходства в объектах, а после образует кластеры из полученных данных;

- частичное обучение – как и в обучении с учителем представляется в виде «задача, ответ», однако сам ответ известен лишь для части из полученных данных;

- трансдуктивное обучение – где даны данные о предыдущих выборках и нужно на их основе сделать предсказания относительно других входных данных;

- обучение с подкреплением – данные представляются в виде таких пар, как «ситуация, выбранное решение», а ответами являются значения функционала качества;
- динамическое обучение – входные данные могут идти потоком, а после, по каждому прецеденту принимать незамедлительные решения.
- обучение может быть как с учителем, так и без него.
- активное обучение – отличается от остальных тем, что следующий претендент может быть выбран самостоятельно, а после уже быть известным;
- метаобучение – прецеденты уже были когда-то раньше решенными задачами в ходе обучения.

Любой из представленных выше способов может стать основой для будущего машинного обучения, а также классификации алгоритмов обучения нейросети [3].

В наше время машинное обучение уже имеет крупный недостаток, а проблема заключается в том, что оно не универсально и под каждую отдельно поставленную задачу нужно обучать нейронную сеть заново, с нуля [25].

Первым шагом системе распознавания лиц необходимо найти лицо на входном изображении и выделить область с лицом. Для этого программное обеспечение использует разнообразные алгоритмы: например, определение схожести фотографий по пропорциям лица и цвета кожи, после выделяет контуры на изображении, после выделяет симметрии при помощи при помощи нейронной сети.

Наиболее эффективным на данное время считается метод Виолы-Джонса [17], который может использоваться в режиме реального времени [18]. При помощи данного метода система распознает лица даже при повороте на 30 градусов. На рисунке 1.3 демонстрируется схема распознавания лица с помощью данного алгоритма.

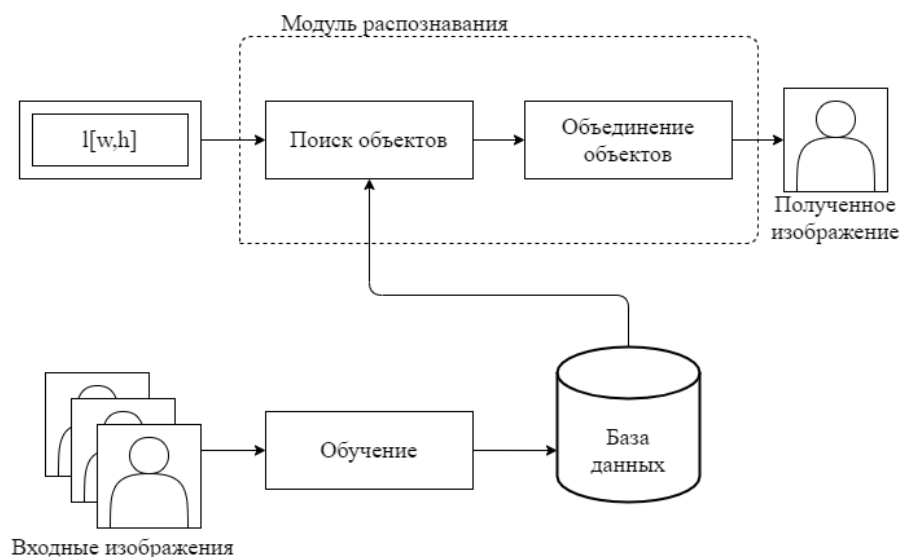


Рисунок 1.3 – Схема распознавания лица методом Виолы-Джонса

Этот метод основывается на признаках Хаара, которые представляют собой набор прямоугольных масок разной формы черно-белого цвета, в свою очередь маски накладываются на различные части изображения [17]. После, алгоритм производит сложение яркостей всех пикселей входного изображения, и рассчитывает разность полученных значений. Следующим шагом, система сравнивает полученные результаты и, как, только определив лицо на фотографии, продолжает его отслеживать для выбора оптимального ракурса и качества входного изображения. Для этого используются алгоритмы предсказания вектора движения или корреляционные алгоритмы.

После нахождения удачного ракурса, удовлетворяющие условию, система начинает работу по распознаванию лица, а также к его сравнению с имеющейся заранее подготовленной базой.

Как правило, программой выделяется приблизительно 100 характерных лицевых точек, которые у каждого человека являются уникальными. Самыми важными измерениями для программ распознавания лиц являются расстояние между глазами, ширина ноздрей, длина носа, высота и форма скул, ширина подбородка, высота лба и так далее [13].

Теперь следует перейти к рассмотрению всех требований к будущему программному продукту.

1.2 Формализация требований к создаваемому алгоритму моделирование портрета человека с использованием нейронной сети

Благодаря современным технологиями, распознавание лица работает как с полученной фотографии с камеры или заранее полученного изображения, так и моделированием 3D изображения, полученного с помощью специальных датчиков или при помощи нейронной сети [9].

При использовании двумерных изображений можно успешно распознать лицо можно только при съемке в анфас и при хорошем освещении, что в свою очередь подходит для охранных систем в различных госучреждениях [10].

В общественных местах рекомендуется распознавание лица при помощи трехмерной модели, это происходит благодаря нескольким синхронизированным камерам, которые фотографируют с различных ракурсов. В результате полученных снимков формируется трехмерная модель объекта, с которой и работает система, после чего формируется модель человека по уникальным точкам. Следующим шагом, система сравнивает полученные данные с данными имеющимся в базе данных, и, при совпадении параметров, человек идентифицируется [10].

На рисунке 1.4 показан один и вариантов распознавания лица человека по точкам.

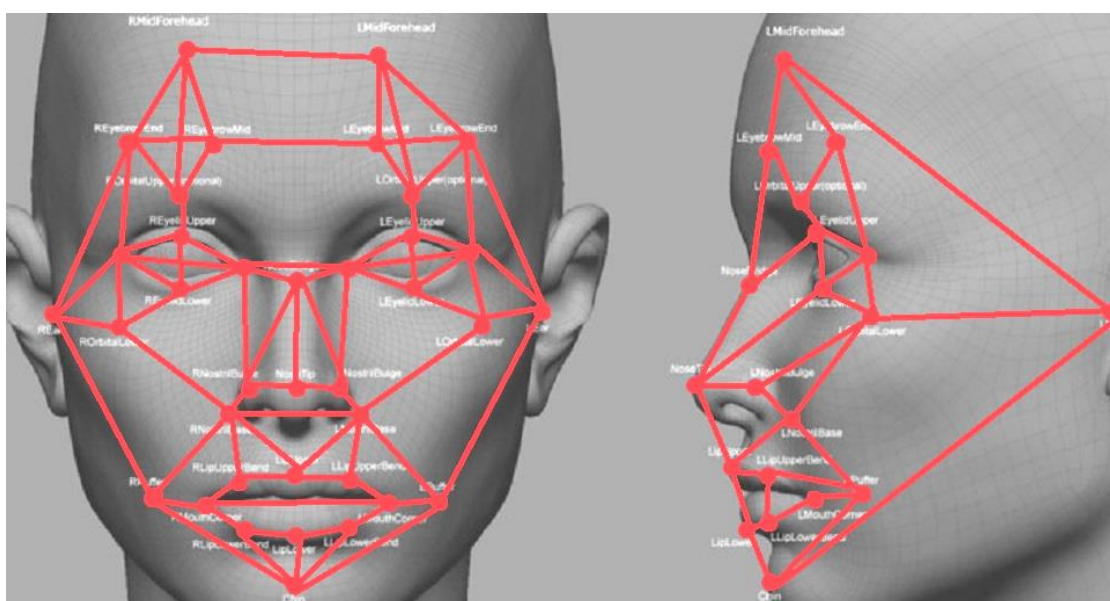


Рисунок 1.4 – Пример распознавания лица человека по точкам

Одной из популярных нейросетей по распознаванию лиц является технология FaceNet [13], которое преобразовывает изображение лица в компактное евклидово пространство, где, чем больше похожи лица, тем они ближе к истине.

FaceNet использует функцию потерь, которая называется TripletLoss. В этой функции минимизируется дистанция между якорем и изображениями, содержащими похожую внешность и максимизирует дистанцию между разными изображениями [13].

На формуле 1 демонстрируется формула расчета функции TripletLoss.

$$Loss = \sum_{i=1}^N \left[\left\| f_i^a - f_i^p \right\|_2^2 - \left\| f_i^a - f_i^n \right\|_2^2 + \alpha \right]_+ \quad (1)$$

где $f(a)$ – энкодинг якоря, $f(p)$ – энкодинг схожих лиц, $f(n)$ – энкодинг разных лиц, не похожих друг на друга, α – константа позволяющая быть уверенным, что нейронная сеть не будет проводить оптимизацию напрямую, т.е. $f(a) - f(p) = f(a) - f(n) = 0$.

FaceNet еще также называют сиамской сетью.

Сиамская сеть – тип архитектуры нейронной сети, позволяющий обучиться дифференцированию входных данных. Другими словами, сиамская сеть помогает понять, какие фотографии похожи, а какие не похожи.

На рисунке 1.5 демонстрируется работа сиамской сети.

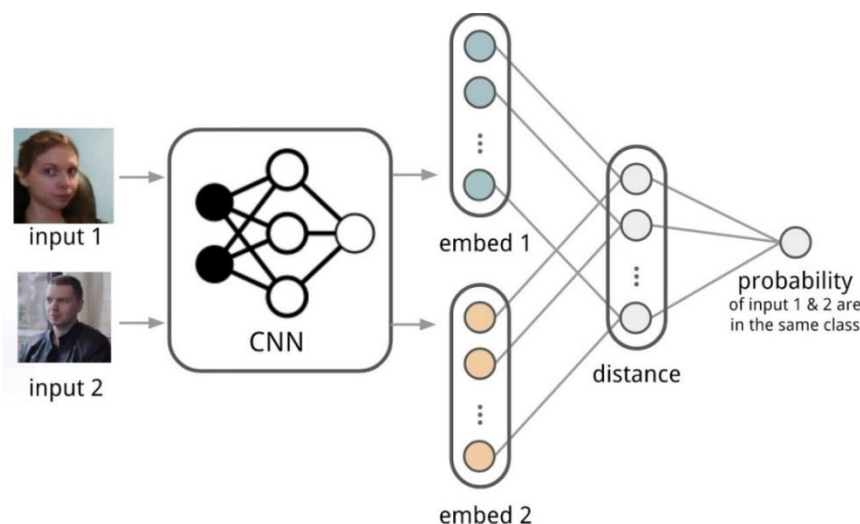


Рисунок 1.5 – Работа сиамской сети

Как видно по рисунку 1.5, сиамские сети состоят из двух идентичных друг другу нейронных сетей, которые содержат в себе одинаковые точные веса. Каждая сеть принимает одно из двух входных изображений. После, выходы последних слоев каждой сети отправляются в функцию, где и определяется, содержат ли эти изображения одинаковые идентификаторы или нет.

Рассмотрев и проанализировав существующие методы распознавания лица, такие как FaceNet [13], OpenFace [16] был составлен список требований к разрабатываемому алгоритму моделирования портрета человека с использованием нейронной сети.

Алгоритм моделирования портрета человека с использованием нейронной сети должен обладать следующими характеристиками:

- использовать как можно меньше ресурсов процессора, видеоадаптера и оперативной памяти, это связано с тем, что необходимый программный продукт должен иметь возможность работать исправно на компьютерах с различной вычислительной мощностью, тем самым будет возможность пользоваться данным продуктом широкому спектру организаций даже со слабыми устройствами;
- уметь распознавать лицо с различных ракурсов и при плохом освещении. Это связано с тем, что программа должно уметь работать при разных условиях, как и днем, так и ночью [10];
- идентифицировать лица, затрачивая как можно меньше времени, чтобы избежать каких-либо задержек при идентификации человека;
- быть кроссплатформенной, чтобы была возможность использовать программы на различных операционных системах.

Полученный список требований был скомпонован в следующую модель информационной системы.

Система должна:

- импортировать фотографию из файловой системы ОС или же из вебкамеры;

- анализировать полученное изображение;
- применять алгоритм к импортированному изображению;
- распознавать лицо на изображении.

Рассмотрев основные требования к будущему разрабатываемого алгоритма, следует перейти на стадию проектирования.

Выводы по первому разделу.

Были проанализированы существующие алгоритмы распознавания лица человека, а также различные варианты работы алгоритма. Для каждого алгоритма был рассмотрен принцип работы.

Проведённый анализ существующих алгоритмов показал, что данная тема является актуальной и востребованной на современном рынке, но, к сожалению, далеко не все страны активно используют идентификацию человека по его лицу.

Исходя из полученных данных, был сформулирован список нужных требований к разрабатываемому алгоритму для распознавания лица человека.

Во втором разделе будет рассмотрен подробно процесс разработки данного алгоритма.

2 Проектирование алгоритма моделирование портрета человека при помощи нейронной сети

2.1 Составление структуры разрабатываемого алгоритма моделирование портрета человека с использованием нейронной сети

Основной задачей алгоритма является реализация распознавания лица с применением нейронных сетей и машинного обучения, где в ходе обучения программный код с каждым разом начинает распознавать и идентифицировать человека с наименьшими затратами по времени. Рассмотрим процедуру распознавания лица и все взаимодействия, связанные с данным процессом [9].

На стадии планирования структуры проекта было принято решение добавить возможность работы с базой фотографий пользователей, выдавая им идентификационный номер, после система делает несколько снимков человека с разных ракурсов и сохраняет в отдельную папку с базой лиц, после система проводит распознавание личности и указывает имя пользователя.

Основной задачей алгоритма моделирования портрета человека с помощью нейронной сети является применение нейронной сети, рассмотрим, в связи с этим, процедуру моделирования лица. Блок схема представлена на рисунке 2.1.

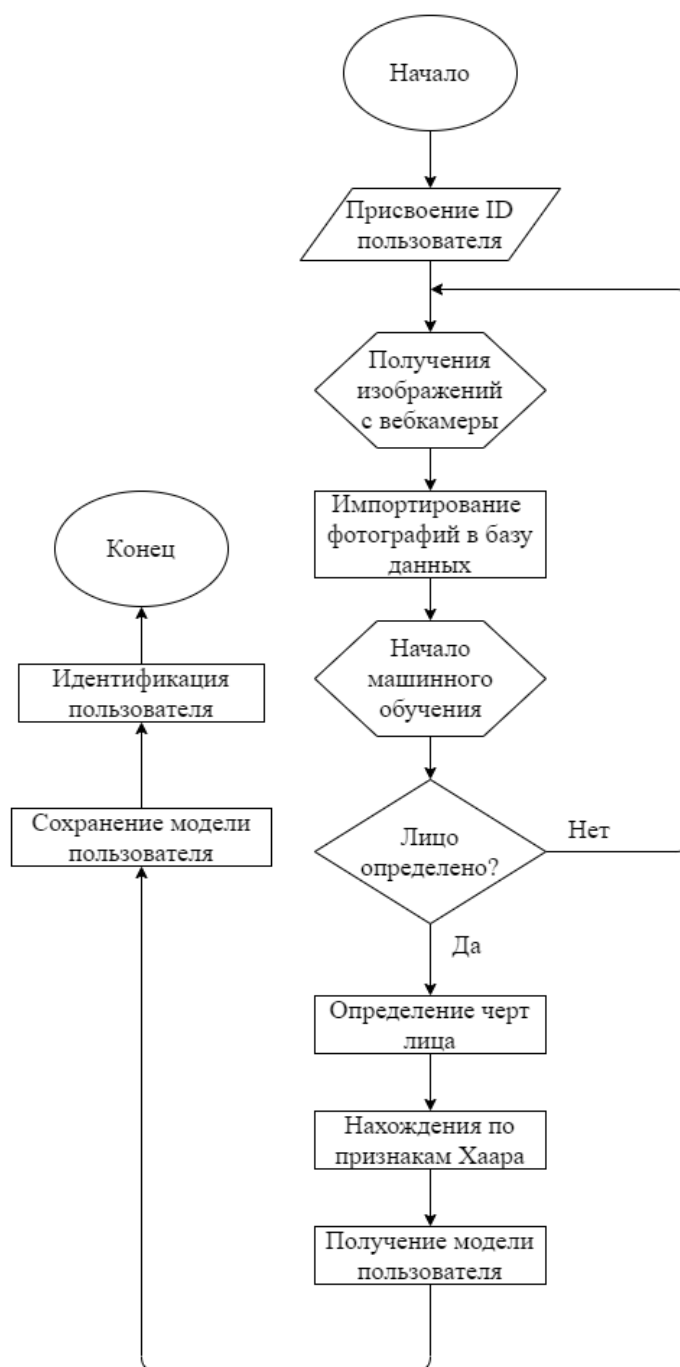


Рисунок 2.1 – Блок схема процедуры моделирования лица

Следующим этапом была поставлена задача разработать логическую модель будущего программного продукта.

На рисунке 2.2 представлена логическая модель разрабатываемой программы.

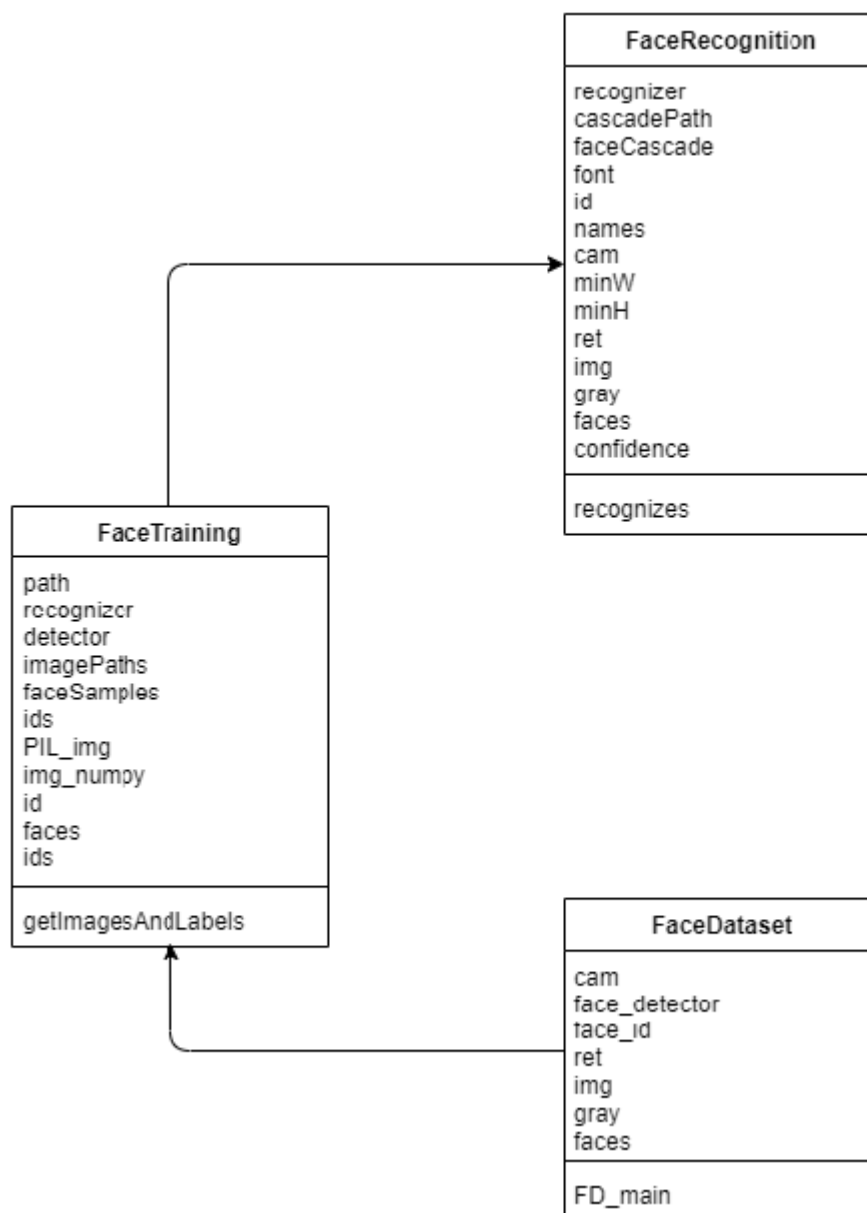


Рисунок 2.2 – Логическая модель данных разрабатываемого программного продукта

На данном этапе взаимодействия с внешними системами был создан состав переменных, которые были использованы для создания программного продукта в решении поставленной задачи распознавания лица.

Первым этапом идет добавление лица человека в базу данных, нейронная сеть на этом этапе моделирует полученный портрет.

Вторым этапом идет обучение работы нейронной сети с данным пользователем.

На заключительном этапе получается на выходе работа вебкамеры, которая персонализирует заранее заданных пользователей и идентифицирует их. В результате написания данного пункта была рассмотрена логическая модель программного продукта.

2.2 Выбор инструмента разработки программы на основе составленной модели

Управление программным обеспечением осуществляется со стороны внешней системы визуализации и в качестве типа приложения выбрана разработка консольного приложения.

Основными причинами выбора Python, как язык программирования для реализации дипломной работы являются некоторые причины.

Высокая скорость разработки. Если сравнивать Python с компилируемыми или строго типизированными языками, такими как C, C++ и Java, то данный язык программирования в несколько раз повышает производительность труда разработчика, а объем разрабатываемого программного кода составляет треть эквивалентного программного кода на вышеупомянутых языках. Данный факт приводит к меньшему затрату времени на отладку и меньший объем затрат труда на сопровождение [1].

Отсутствие компиляции под каждую операционную систему. Программы, которые написаны на языке Python [23] запускаются сразу же на любой операционной системе и на любой платформе, пропуская длительные этапы компиляции и связывания, необходимые в таких языках программирования, как C++, Java C#, что еще больше увеличивает скорость разработки программного обеспечения.

Переносимость ПО. Наибольшая часть разработанных программ выполняется без изменений на всех основных современных платформах. Перенос программного кода из одной операционной системы в другую, как правило, заключается в копировании исходных файлов с одного носителя на другой.

Библиотеки. Репозиторий пакетов Python содержит огромное число собранных и переносимых функциональных возможностей, которые представлены в виде различных библиотек [26]. Стандартная библиотека содержит в себе огромный функционал, который востребован в прикладных программах, начиная как с простого математического подсчёта, так заканчивая сложными задачами, например, моделирование портрета человека при помощи нейронной сети. Допускается расширение функционала как своими собственными библиотеками и алгоритмами, так и библиотеками, созданными разработчиками по всему миру. Из числа сторонних разработок можно назвать инструменты компьютерного зрения, инструменты написания собственных нейронных сетей и многое другое.

Примером тому служит библиотека OpenCV, которая позиционирует себя, как свободный и мощный инструмент для взаимодействия с изображениями [15].

Внедрение компонентов. Механизмы интеграции способствуют работоспособности части кода с другими частями приложения благодаря механизмам интеграции.

Интеграция предусматривает использование языка для расширения возможностей функционала разрабатываемого программного обеспечения. Примером интеграций может служить взаимодействие программного кода на языке Python с компонентами на языке Java, а также вызов функций из библиотек, созданных на языках C и C++ [26]. Также есть возможность обращаться к программам, написанным на языках C и C++, а также существует взаимодействие такими платформами, как COM и .NET.

К сожалению, данный язык программирования имеет и ряд минусов, если сравнивать его с другими средствами разработки.

К минусам данного языка можно отнести:

- скорость работы;
- потребление памяти.

Скорость работы. Данный язык программирования является не самым быстрым среди его аналогов. Скорость выполнения программ может быть ниже относительно других инструментов разработки.

Потребление памяти. В связи с тем, что в Python отсутствуют строго-типизированные данные, то потребление памяти иногда оказывается не минимальным [23].

Но для реализации данного программного продукта был выбран именно данный язык программирования и это связано с тем, что для реализации алгоритма моделирования портрета человека лучше всего использовать библиотеки созданные для Python. Вдобавок, положительных аспектов при выборе языка программирования оказалось намного больше, чем отрицательных.

На сегодняшний день существует множество различных библиотек, которые работают с изображениями и с машинным обучением. Этими библиотеками являются [20]:

- OpenCV;
- Dlib;
- NumPy;
- Matplotlib;
- Mahotas;
- SciPy.

Каждую библиотеку стоит рассматривать отдельно, так как они играют не маловажную роль в реализации программного кода.

Библиотека OpenCV содержит набор модулей, где каждый связан с областями компьютерного зрения [28]. Данная библиотека включает в себя следующий набор функций:

- работа с двумерными и трехмерными изображениями;
- система распознавания лиц человека;
- распознавание жестов рук;

- распознавание движений и отслеживание движений;
- идентификация объектов.

Чтобы была поддержка некоторых из вышеперечисленных областей, в OpenCV включается статическая библиотека машинного обучения. В этой библиотеке содержатся [7]:

- бустинг;
- деревья повышения градиента;
- глубокие нейронные сети.

Бустинг – метаалгоритм машинного обучения, где применяется уменьшение смещения и дисперсии в обучении с учителем. Также оно определяется как семейство алгоритмов машинного обучения, которое преобразует более слабые обучающие алгоритмы к более сильным. Большинство алгоритмов бустинга состоит из итеративного обучения слабых классификаторов с целью сборки их в сильный классификатор. Когда они добавляются, им обычно приписываются некоторым образом веса, которые, обычно, связаны с точностью обучения. После того, как слабый классификатор добавлен, веса пересчитываются. Входные данные с неверной классификацией получают больший вес, а правильно классифицированные экземпляры теряют вес. В итоге, последующее слабое обучение фокусируется больше на примерах, где предыдущие слабые обучения дали ошибочную классификацию [25].

Деревья повышения градиента. Смысл данного метода заключается в том, что этот метод машинного обучения разработан для задач регрессии и классификации, где создается модель прогнозирования в виде некоего множества моделей слабого прогнозирования, например, дерева решений.

Глубокие нейронные сети (Deep neural network) [3]. Deep neural network представляет некую искусственную нейронную сеть с несколькими слоями между входным и выходным слоями. Данная сеть находит правильную математическую манипуляцию, где превращает входные данные в выходные, будь то линейное отношение или нелинейное. В свою очередь, она движется по

слоям, рассчитывая вероятность каждого выхода. В пример можно привести обученную глубокую нейронную сеть, которая создана для распознавания породы кошек, пройдет по заранее заданным фотографиям, DNN определяет какая порода у кошки и показывает точность определения. Каждая математическая манипуляция считается слоем, а сложные DNN имеют множество слоев. Глубокие нейронные сети моделируют сложные нелинейные отношения. Архитектуры DNN генерируют различные композиционные модели, где объект выражается в виде многоуровневой композиции примитивов, после дополнительных уровней позволяют составлять элементы из более низких уровней, моделируя сложные данные с меньшим количеством единиц. Данная нейронная сеть представляет собой сеть с прямой связью, где данные передаются от входного уровня к выходному уровню без обратной связи. Первым шагом DNN создает карту виртуальных нейронов и генерирует случайные числовые значения или как их еще можно назвать, «веса» соединениям между ними. Веса и входные данные умножаются и возвращают выходной сигнал от 0 до 1. Если сеть не точно распознала конкретный шаблон, алгоритм будет корректировать весовые коэффициенты. В итоге, алгоритм делает определенные параметры более влиятельными, пока не будут определены правильные математические манипуляции для полной обработки всех полученных данных. На рисунке 2.3 демонстрируется принцип распознавания человека при помощи OpenCV [28].

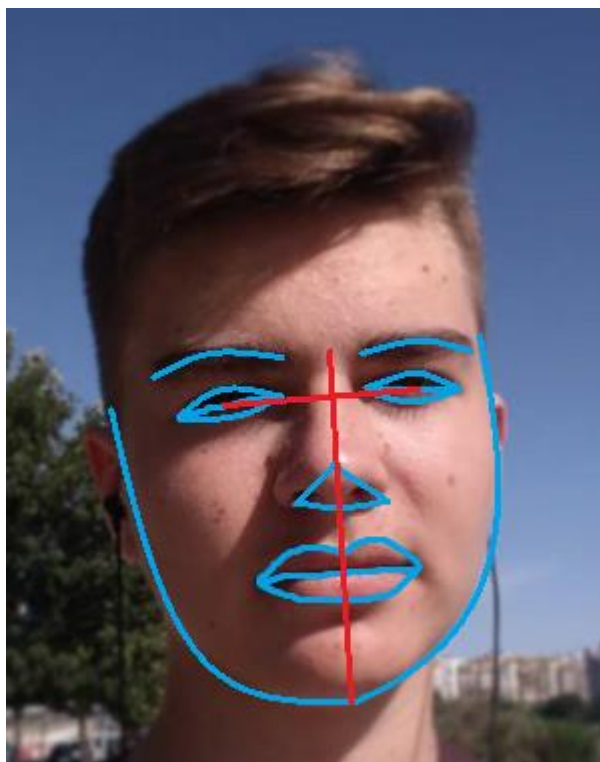


Рисунок 2.3 – Моделирование портрета человека при помощи библиотеки OpenCV

Библиотека `dlib` включает в себя множество полезных функций, а также она является на полной OpenSource лицензии, делая ее свободной для использования [12]. Преимуществом библиотеки `dlib` также является то, что она:

- имеет несколько реализаций SVM (support vector machine);
- имеет несколько реализаций нейронной сети;
- подбирает функции матричной геометрии;
- подбирает алгоритмы градиентного спуска;
- SURF|HOG.

Если обратить внимание на последний пункт SURF|HOG, то он используется как каскад для поиска лиц, вдобавок позволяет искать особые точки на лицах. Недостатком его проявляется в том, что он немного хуже определяет лицо, если оно находится не фронтально по отношению к камере,

но если лицо фронтально, то его точность определения намного выше технологии Haar [19].

Поиск особых точек – очень интересная и важная технология. Существует множество проприетарных решений, а также множество открытых, но, к сожалению, плохих, поэтому, dlib решение – одно из первых открытых и хороших.

NumPy — модуль для python с открытым исходным кодом, представляющий собой общие математические и числовые операции, имеющие вид пре-скомпелированных и быстрых функций, которые объединяются в высокоуровневые пакеты. Они же в свою очередь обеспечивают тот функционал, который приравнивается к функционалу Matlab. NumPy включает в себя большой набор математических функций [29]. В эти функции входят такие инструменты, как:

- анализ данных;
- машинное обучение;
- обработка векторов и матриц.

Многие библиотеки используют NumPy как один из основных элементов своей инфраструктуры.

Matplotlib – библиотека на языке программирования Python для визуализации данных двумерной и трехмерной графики. Библиотека Matplotlib это легко конфигурируемый пакет, который с такими библиотеками, как NumPy, SciPy, а также IPython предоставляет различные математические возможности, которые есть в среде MATLAB. Данная библиотека позволяет визуализировать данные в различные форматы [30].

Форматы, которые поддерживает библиотека Matplotlib входят:

- JPEG;
- PDF;
- PNG;
- Postscript;

- RGBA;
- SVG;
- SVGZ;
- EPS;
- EMF;
- TIFF.

Mahotas – библиотека, разработанная для python, которая служит для работы компьютерного зрения и обработки приложений и представляет библиотеку, работающую с двумерными и трехмерными изображениями. Наиболее расширенная обработка входных изображений происходит благодаря извлечению информации из изображений [30]. Mahotas обладает большими возможностями для компьютерного зрения, которые позволяют выполнять такие процессы, как морфологическая обработка, свертка. Mahotas - это одна из быстрых библиотек с минимальным объемом кода и зависимостями [20].

SciPy – один из важнейших научных модулей, как и NumPy для Python. В большей степени он отлично подходит для решения задач по обработке и прочей работы с входящими изображениями [21]. Например, в данном модуле присутствует возможность работать с n-мерными массивами. В этот модуль включаются функционал для работы с линейной и нелинейной фильтрации, бинарной морфологии, интерполяции сплайнами и изменениями объектов.

Основными возможностями библиотеки являются:

- поиск минимумов и максимумов заданных функций;
- вычисление интегралов функций;
- обработка сигналов;
- обработка изображений;
- работа с генетическими алгоритмами;
- решение дифференциальных уравнений.

В свою очередь для реализации нейронных сетей существует множество различных библиотек, включающих в себя:

- Theano;
- TensorFlow;
- Keras.

Theano – это библиотека численных вычислений, которые выражены NumPy-подобным синтаксисом и компилируемых для эффективных параллельных вычислений как на CPU, так и на GPU. Библиотека Theano представляет некий препроцессор для системы вычислений с многомерными массивами данных, содержащий в себе математические пакеты Mathematica и MATLAB [29].

TensorFlow – это программная библиотека с открытым исходным кодом для машинного обучения, разработанную для решения задач построения и тренировки нейронной сети с целью автоматического нахождения и классификации образов [24]. Входные данные выражаются в виде потоков данных через граф состояний.

Keras – открытая библиотека нейронной сети. Данная библиотека является надстройкой таких библиотек, как TensorFlow и Theano. Основная цель данной библиотеки является оперативной работой с сетями глубокого обучения [24]. В ней содержатся реализации широко применяемых строительных блоков нейронных сетей, в которые включаются:

- слои;
- целевые и передаточные функции;
- оптимизаторы;
- различные инструменты для упрощения работы с изображениями и текстом.

После того, анализа различных доступных библиотек для работы с двумерными и трехмерными изображениями, а также распознавания лиц, следует выбрать тип разрабатываемого алгоритма.

Немаловажную роль в разработке программного обеспечения изучение сверточной нейронной сети.

Свёрточная нейронная сеть – некая архитектура искусственных нейронных сетей, которую в 1988 году предложил Ян Лекун и ее основная цель это эффективное распознавание образов, куда входит состав технологий машинного обучения. В этой нейронной сети используются некоторые особенности зрительной, где были открыты простые клетки, реагирующие на прямые линии под разными углами, и сложные клетки [4]. Реакция сложных клеток связана с активацией определённого набора простых клеток.

Для обучения структуры сети используются стандартные методы, которые чаще всего используют метод обратного распространения ошибки. Функция активации нейронов использует любые заранее выбранные исследования.

Архитектура сети получила свое название из-за наличия в ней операции свёртки. Ее суть заключается в том, что каждый фрагмент изображения умножается на матрицу свёртки поэлементно, а после результат суммируется, записывая в аналогичную позицию выходного изображения.

На сегодняшний день существует четыре основных слоя нейронной сети:

- сверточный слой;
- слой подвыборки (пулинга);
- слой активации;
- полносвязный слой.

Сверточный слой включает для каждого канала свой фильтр. В свою очередь, ядро свёртки обрабатывает предыдущий слой по фрагментам. Весовые коэффициенты ядра свёртки изначально неизвестны и устанавливаются в ходе самого обучения.

Сверточный слой имеет некую особенность, в которую включается небольшое количество параметров, устанавливаемое непосредственно при обучении нейронной сети. В качестве демонстрации можно взять пример, где исходное изображение размеров в 100×100 пикселей по трём каналам, то есть имеет 30000 входных нейронов.

Сам же сверточный слой использует фильтры с ядром 3 на 3 пикселя с 6 каналами на выходе. В процессе обучения определяется 9 весов ядра. Если произвести расчеты, то получается, что $9 \times 3 \times 6 = 162$. То есть, данному слою требуется найти только 162 параметра, что по сравнению с параметрами полносвязной нейронной сети имеет меньшее количество искомым параметров. На рисунке 2.4 демонстрируется сверточный слой, который преобразовывается по нескольким входным каналам.

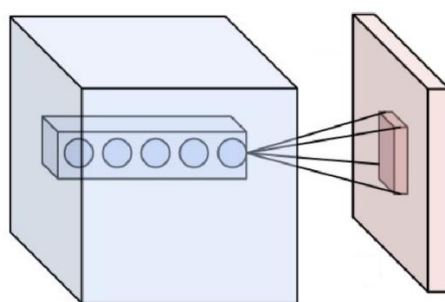


Рисунок 2.4 – Пример работы сверточного слоя

Слой подвыборки или же слой пулинга – это нелинейное уплотнение карты признаков, где группа пикселей с каждым ходом уплотняется вплоть до одного пикселя. Это происходит благодаря нелинейному преобразованию. Преобразования задачи с помощью данного слоя используют непересекающиеся друг с другом прямоугольники, которые в свою очередь шаг за шагом сжимаются в один пиксель, где происходит выборка нужного пикселя, удовлетворяющего максимальному значению среди прочих пикселей в прямоугольниках. Операция подвыборки позволяет в разы уменьшить пространственный объем изображения.

Интерпретация данного слоя выглядит следующим образом, если на предыдущей операции уже были выявлены некоторые признаки, то в последующей обработке подробное изображение уже не нужно и происходит уплотнение до менее подробного изображения.

Помимо слоя подвыборки, вместе с функцией максимума часто используются прочие функции, например, функция среднего значения или L2-

нормирования. На практике было доказано преимущество именно данного слоя подвыборки с функцией максимума, который был включен в различные типовые системы. На рисунке 2.5 показан пример слоя пулинга с функцией максимума на примере фильтра 2 на 2 с шагом 2.

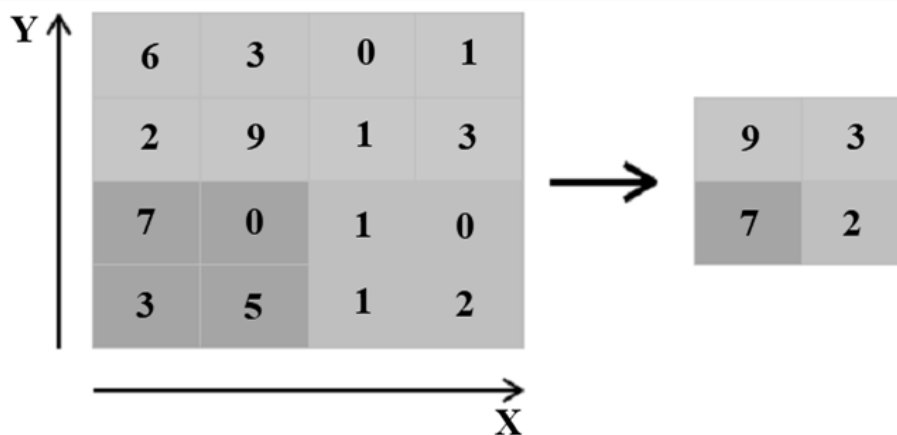


Рисунок 2.5 – Пример работы слоя пулинга

Слой активации – это скалярный результат каждой свёртки, который в свою очередь попадает в функцию активации, которая в свою очередь выглядит, как нелинейная функция. Слой активации в основном логически объединяют со слоем свёртки.

Слой активации является некоторой функцией, где происходит применение к каждому числу входного изображения.

Как правило, функция активации может задаваться такими функциями, как:

- ReLU;
- Sigmoid;
- Tanh;
- Leaky ReLU.

В начале 2000х годов было произведено исследование и предложена новая функция активации, имеющая название rectified linear unit (ReLU), которая в свою очередь позволяет ускорить процесс обучения и заметно упрощает

вычисления. Именно эта функция является самой используемой среди других функций активации.

Слой данной функции ReLU показан на формуле 2.

$$x^l = f(a^l \cdot \text{subsample}(x^{l-1}) + b^l) \quad (2)$$

где x^l – это выход слоя l , $f(x)$ – функция активации, a^l и b^l – это коэффициенты сдвига слоя l , а $\text{subsample}(x^{l-1})$ – это операция выборки локальных максимальных значений.

Последний рассматриваемый слой – это полносвязный слой. Данный слой работает по следующему принципу, где после нескольких проходов свёртки изображения и уплотнения с помощью слоя подвыборки система старается перестроиться от определенной сетки пикселей с высоким разрешением к наиболее абстрактным картам признаков, которые, как правило, с каждым слоем увеличивает число каналов и уменьшает размерность изображения.

В итоге остаётся большой набор каналов, где хранятся небольшие числа полученных данных, которые в свою очередь интерпретируются как самые абстрактные понятия, выявленные из исходного изображения.

После, эти данные объединяются и передаются на обычную полносвязную нейронную сеть, которая состоит из нескольких слоёв. Сами по себе, полносвязные слои утрачивают пространственную структуру пикселей и обладают сравнительно небольшой размерностью.

Теперь можно перейти к выбору тех алгоритмов, которые будут использоваться при разработке программного продукта.

За основу разрабатываемого алгоритма моделирования портрета человек при помощи нейронной сети был взят OpenCV, метод Хаара и API `face_recognition` [6].

`Face_recognition` является самым популярным API с открытым исходным кодом на данный момент во всем мире [15].

Признаки Хаара – это признаки цифрового изображения, которые используются в распознавании различных образов, например, образов человеческого тела или же человеческого лица [6]. На рисунке 2.6 демонстрируется работа признаков Хаара.

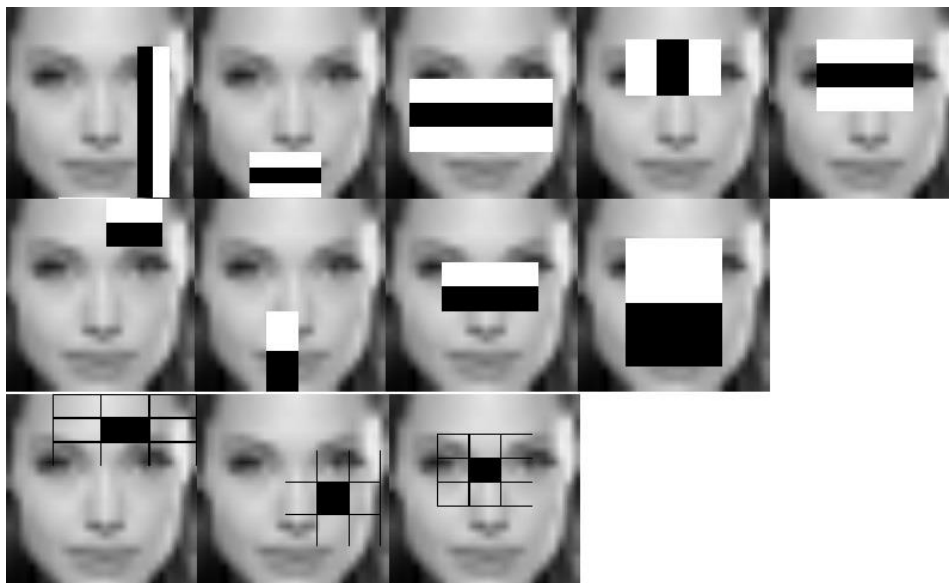


Рисунок 2.6 – Признаки Хаара

Находится общий признак Хаара для лиц, где два смежных прямоугольных региона, лежащих на тех местах, где полученные черты выделяются на фоне других черт лица человека [19].

В результате были рассмотрены и описаны средства реализации программного продукта, разрабатываемого на основе составленного алгоритма моделирование портрета человека с использованием нейронной сети, а также был выбран язык программирования Python в качестве средства для разработки. Теперь перейдем к самой реализации программного продукта.

2.3 Реализация программного продукта моделирования портрета человека при помощи нейронной сети

Для написания программного обеспечения моделирования портрета человека при помощи нейронной сети понадобится признака Хаара, библиотеки

OpenCV и библиотеки NumPy. Было принято решение разработать программное обеспечение с использованием языка программирования Python.

Опираясь на разрабатываемый алгоритм, рассмотрим процесс моделирование портрета человека и отобразим его на рисунке 2.7.

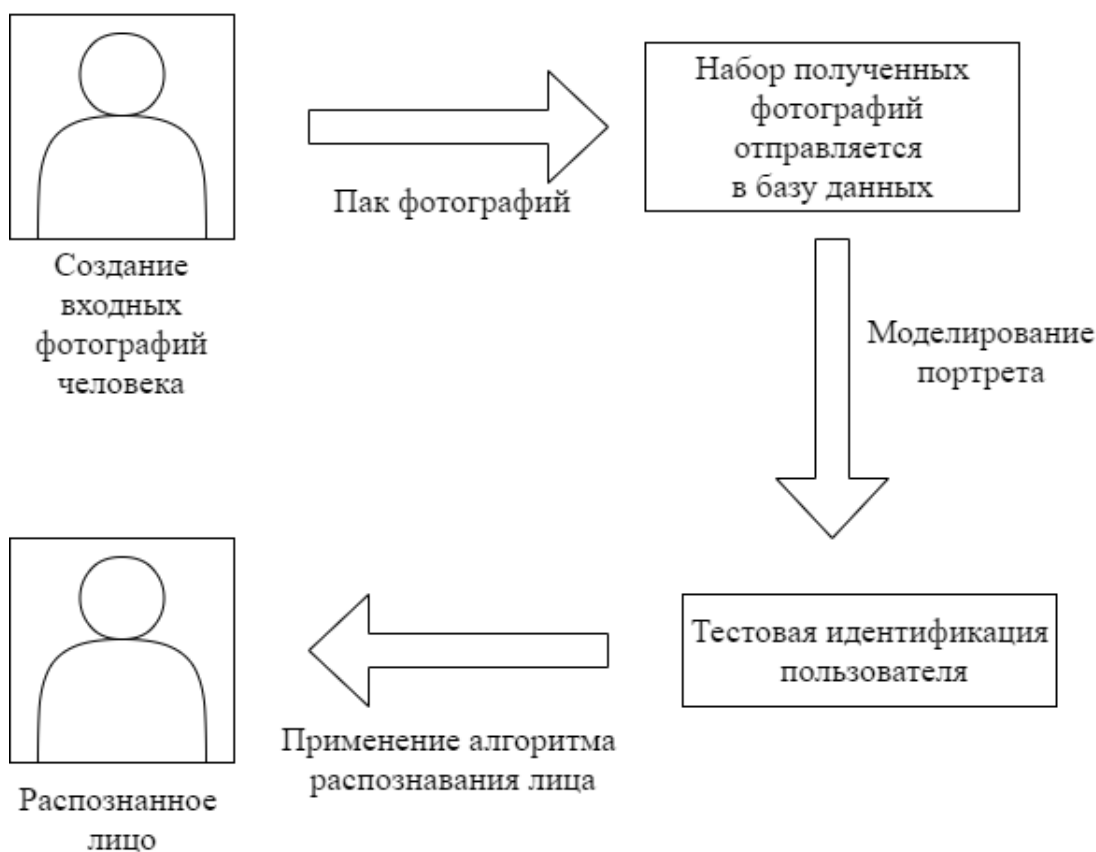


Рисунок 2.7 – Процесс моделирование и распознавание лица человека

Теперь перейдем непосредственно к написанию кода программного обеспечения.

Для начала следует загрузить набор входных изображений лица в базу. Для этого первым шагом задаем ID человека и при помощи вебкамеры создаем набор фотографий пользователя.

На рисунке 2.8. показана инициализация пользователя по ID.

```

# Для каждого человека введите один числовой идентификатор лица
face_id = input('\n enter user id end press <return> ==> ')

print("\n [INFO] Initializing face capture. Look the camera and wait ...")
# Инициализировать лицо
count = 0

```

Рисунок 2.8 – Инициализация по ID

Далее создается 30 фотографий и отправляется в локальную базу. Фрагмент кода для загрузки изображений в локальное хранилище демонстрируется на рисунке 2.9.

```

ret, img = cam.read()
img = cv2.flip(img, 0)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
faces = face_detector.detectMultiScale(gray, 1.3, 5)

for (x,y,w,h) in faces:

    cv2.rectangle(img, (x,y), (x+w,y+h), (255,0,0), 2)
    count += 1

    # Сохранение захваченного изображения в папку наборов данных
    cv2.imwrite("dataset/User." + str(face_id) + '.' + str(count) + ".jpg", gray[y:y+h,x:x+w])

    cv2.imshow('image', img)

k = cv2.waitKey(100) & 0xff # 'ESC' для выхода
if k == 27:
    break
elif count >= 30: # делается 30 фотографий
    break

```

Рисунок 2.9 – Загрузка изображений в локальное хранилище

В ходе Исследования было приятно решение делать 30 фотографий, так как этого достаточно для того, что в будущем нейронная сеть могла определить пользователя.

В итоге у нас получается набор изображений, которые содержат в себе изображения пользователя с разных ракурсов.

Следующим шагом идет обучение нейронной сети, которая моделирует портрет человека [15]. На этом этапе пользователю нужно смотреть на камеру, и нейронная сеть построит модель человека, после сообщит пользователю, что он идентифицирован. Фрагмент кода изображен на рисунке 2.10.

```

# функция для получения изображений и меток данных
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') # преобразовать его в оттенки серого
        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)

    return faceSamples,ids

print ("\n [INFO] Training faces. It will take a few seconds. Wait ...")
faces,ids = getImagesAndLabels(path)
recognizer.train(faces, np.array(ids))

# Сохранить модель в trainer/trainer.yml
recognizer.write('trainer/trainer.yml')

```

Рисунок 2.10 – Тренировка нейронной сети для распознавания пользователя

На этом этапе полученная модель пользователя загружается в trainer.yml, где записывается его лицо и полученный результат находится на рисунке 2.11.



Рисунок 2.11 – Объемная модель пользователя, полученная в результате работы программы

На рисунке 2.12 демонстрируется то, как видит программный код лицо пользователя.

```
opencv_lbphfaces:
  threshold: 1.7976931348623157e+308
  radius: 1
  neighbors: 8
  grid_x: 8
  grid_y: 8
  histograms:
    - !!opencv-matrix
      rows: 1
      cols: 16384
      dt: f
      data: [ 1.02040814e-02, 1.78571418e-02, 0., 0., 2.55102036e-03,
              0., 1.27551018e-03, 1.14795920e-02, 0., 0., 0., 0.,
              1.27551018e-03, 0., 3.82653065e-03, 6.37755077e-03,
              8.92857090e-03, 3.82653065e-03, 0., 1.27551018e-03,
              2.55102036e-03, 0., 0., 7.65306130e-03, 0., 1.27551018e-03,
              0., 0., 1.14795920e-02, 1.27551018e-03, 7.65306130e-03,
              2.55102031e-02, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
              0., 0., 0., 0., 0., 1.27551018e-03, 1.27551018e-03, 0., 0.,
              0., 0., 0., 0., 1.27551018e-03, 0., 0., 0., 2.55102036e-03,
              0., 5.10204071e-03, 6.37755077e-03, 3.82653065e-03,
              1.27551018e-03, 0., 0., 2.55102036e-03, 0., 0.,
              2.55102036e-03, 0., 0., 0., 0., 0., 0., 0., 0.,
              1.27551018e-03, 1.27551018e-03, 0., 0., 0., 0., 0., 0.,
              0., 0., 0., 0., 1.27551018e-03, 0., 1.27551018e-03, 0., 0.,
              0., 0., 1.27551018e-03, 0., 0., 0., 0., 0., 0., 0., 0.,
              0., 0., 3.82653065e-03, 1.27551018e-03, 0., 0., 0., 0.,
              2.55102036e-03, 1.27551018e-03, 0., 0., 0., 6.37755077e-03,
              0., 3.82653065e-03, 5.10204071e-03, 0., 2.55102036e-03, 0.,
              3.82653065e-03, 0., 1.27551018e-03, 0., 1.14795920e-02, 0.,
              0., 0., 0., 0., 0., 0., 2.55102036e-03, 0., 0., 0., 0.,
              0., 0., 1.27551018e-03, 0., 0., 0., 0., 0., 0., 0.,
              1.14795920e-02, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
              0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
              0., 0., 0., 0., 0., 3.82653065e-03, 2.55102036e-03,
              7.90816322e-02, 0., 5.86734675e-02, 1.27551018e-03, 0., 0.,
              2.55102031e-02, 0., 0., 0., 0., 0., 1.27551018e-03, 0.,
              5.10204071e-03, 0., 3.82653065e-03, 0., 0., 1.27551018e-03,
              0., 0., 2.55102036e-03, 0., 0., 0., 0., 0., 0., 0.,
              1.40306121e-02, 5.10204071e-03, 2.32142851e-01, 0.,
              5.61224483e-02, 2.55102036e-03, 0., 1.27551018e-03,
              3.31632644e-02, 0., 0., 0., 0., 0., 0., 0., 6.37755077e-03,
```

Рисунок 2.12 – Определение лица пользователя в trainer.yml

Таким образом, была получена система моделирование человека и идентификация пользователя.

Выводы по второму разделу:

В результате описываемой в данном разделе работы было реализовано программное обеспечение, которое получает с камеры фотографии, по полученным результатам происходит обучение и моделирование человека, а после его идентификация.

Разработанный алгоритм выполняет следующие шаги:

- получает набор изображений с вебкамеры;
- записывает данные на локальный диск;
- с помощью нейронной сети моделирует портрет человека;
- на основе полученных данных моделирует пользователя;
- идентифицирует пользователя и показывает результат в режиме онлайн;

Основным минусом разработанного алгоритма является долгое распознавание человека на слабых процессорах, а также долгое первоначальное обучение нейронной сети.

Описав процесс разработки во втором разделе, перейдём к этапу тестирования разработанного программного продукта.

3 Тестирование разработанного алгоритма моделирование портрета человека с использованием нейронной сети

3.1 Тестирование программного продукта на основе разработанного алгоритма моделирование портрета человека с использованием нейронной сети

В ходе выполнения ВКР было спроектировано, разработано и протестировано программное обеспечение, реализующее описанный алгоритм моделирование портрета человека при помощи нейронной сети.

В обязанности и особенности разработанного программного обеспечения входят:

- автоматическая инициализация и запуск веб-камеры;
- сохранение фотографий человека в базу данных;
- на основе полученных изображений происходит моделирование портрета пользователя;
- обучение нейронной сети для работы с конкретным пользователем;
- возможность идентифицировать как одного пользователя на камере, так и нескольких одновременно.

Разработка и тестирование программного алгоритма производилось на устройстве со следующими техническими характеристиками:

- процессор Intel® Core™ i5-7200U 2x3.10GHz;
- видеокарта NVIDIA GeForce® 940MX 2GB;
- оперативная память KINGSTON VALUERAM KVR24S17S8/8 DDR4 – 2x8GB;
- вебкамера Chicony Panda DC-7144.

При запуске приложение на первом этапе следует выдать нужному пользователю его идентификационный номер. Это демонстрируется на рисунке 3.1.

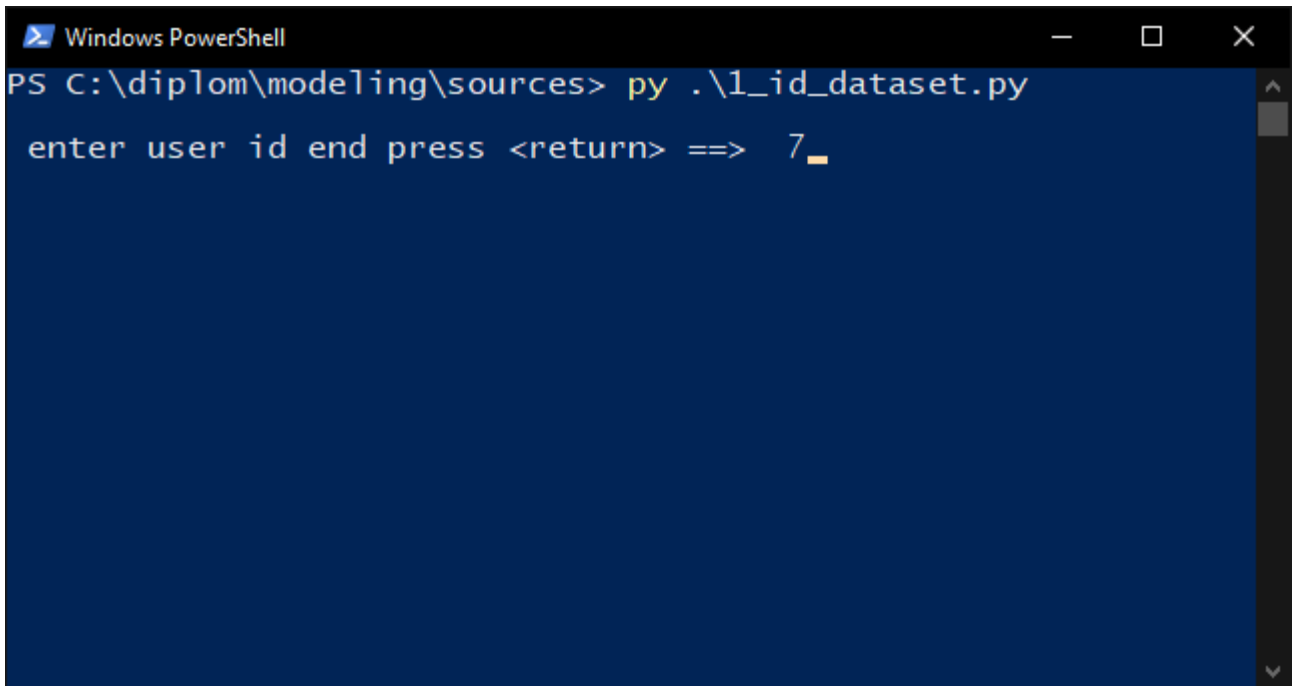


Рисунок 3.1 – Задается идентификатор пользователя

После происходит инициализация веб-камеры. На этом этапе происходит инициализация лица человека и сохранение его лица в базу данных. Этот этап показан на рисунке 3.2.

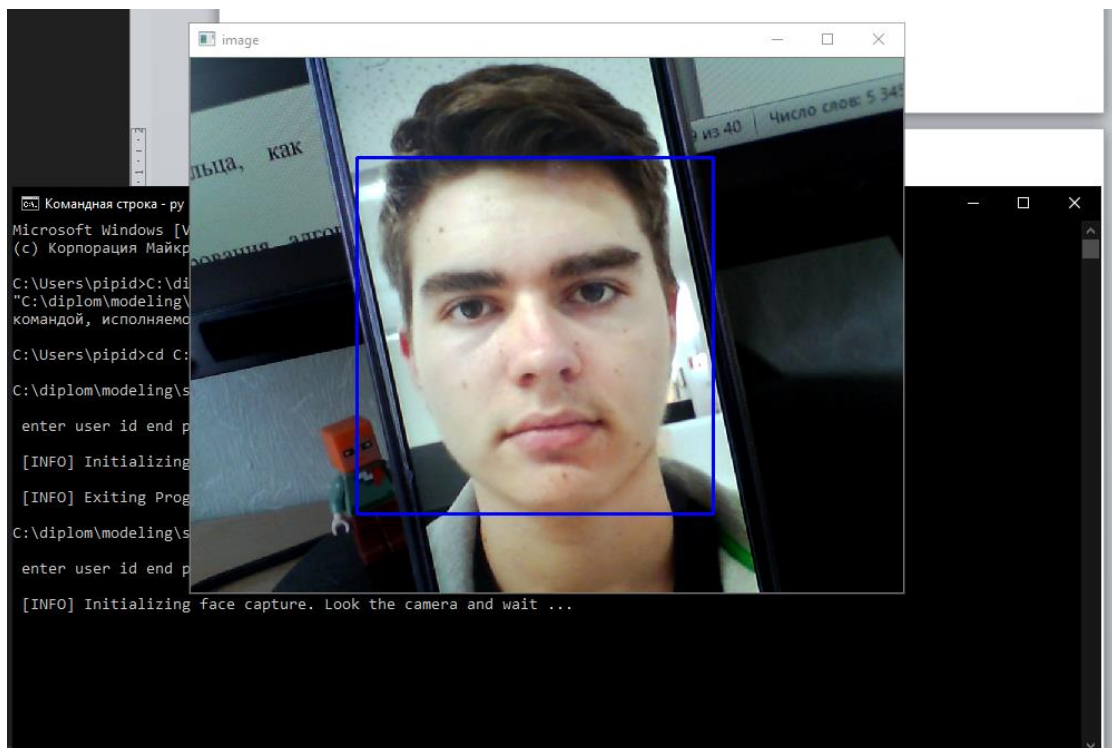


Рисунок 3.2 – Инициализация пользователя

Чтобы убедиться в том, что лицо пользователя занесено в базу данных и распознано правильно, следует перейти в папку dataset, где находятся фотографии всех пользователей. На рисунке 3.3 показана база лиц заранее внесенных пользователей.

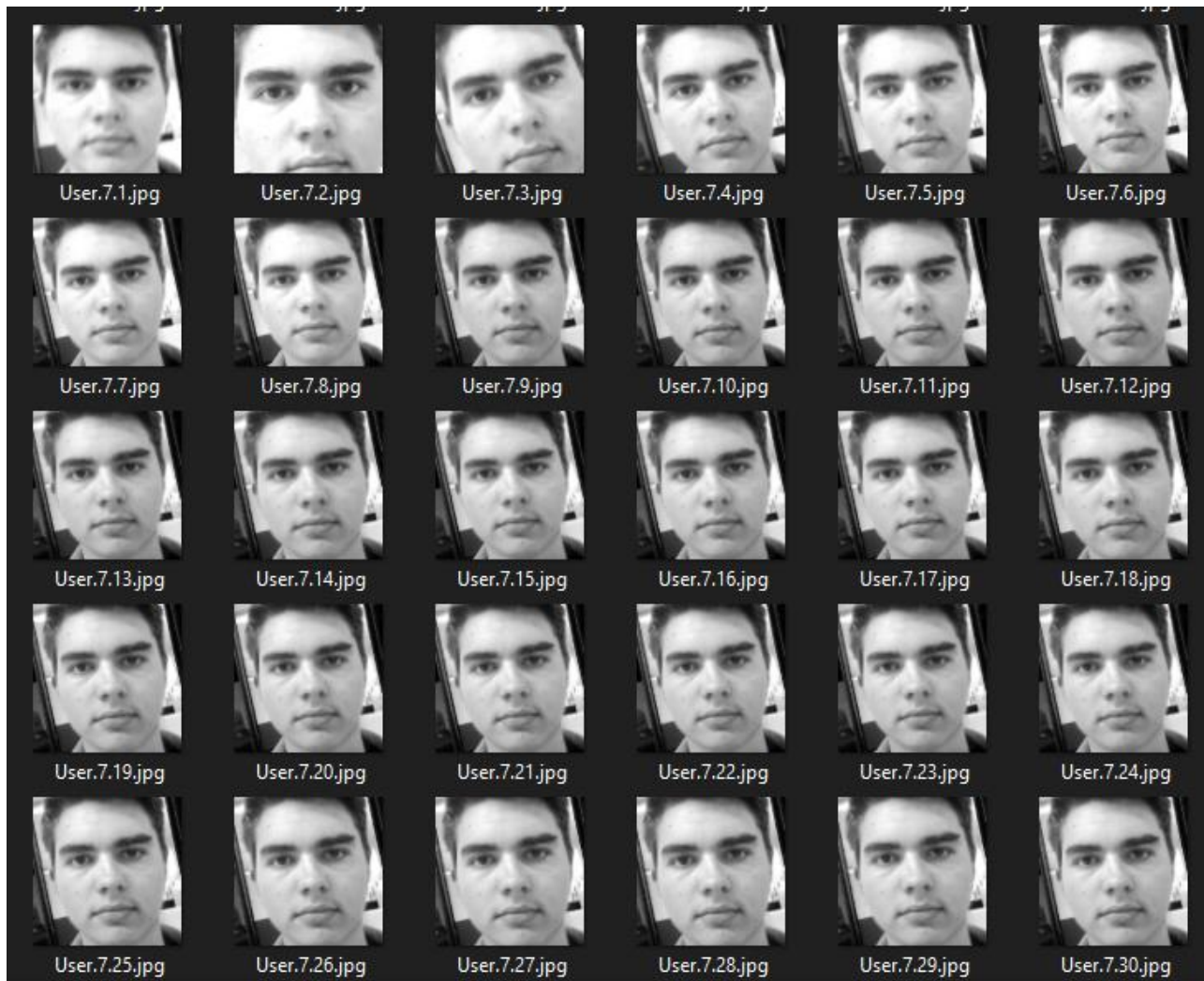
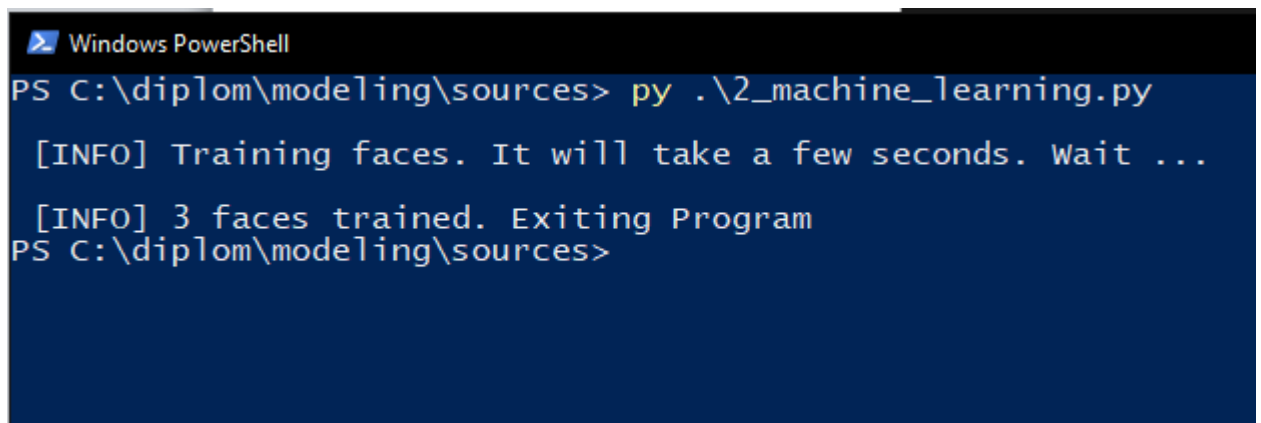


Рисунок 3.3 – Локальное хранилище фотографий пользователя

После этого нужно произвести обучение программы работать с лицом пользователя. Именно на этом этапе происходит моделирование лица человека, которое в будущем будет работать в его идентификации [14]. На рисунке 3.4 демонстрируется машинное обучение распознавания лица человека.

A screenshot of a Windows PowerShell terminal window. The title bar at the top reads "Windows PowerShell". The command prompt shows the current directory as "C:\diplom\modeling\sources". The user has entered the command "py .\2_machine_learning.py". The output of the script is displayed in green text: "[INFO] Training faces. It will take a few seconds. Wait ..." followed by "[INFO] 3 faces trained. Exiting Program". The prompt returns to "PS C:\diplom\modeling\sources>".

```
Windows PowerShell
PS C:\diplom\modeling\sources> py .\2_machine_learning.py
[INFO] Training faces. It will take a few seconds. Wait ...
[INFO] 3 faces trained. Exiting Program
PS C:\diplom\modeling\sources>
```

Рисунок 3.4 – Машинное обучение в работе с лицом пользователя

Как показано на рисунке 3.4, система первым делом производит обучение нейрона, где на этом этапе моделируется лицо, процесс не быстрый, но довольно эффективный. После того, как проходит обучение, система выдает в качестве результата сообщение о том, сколько он лиц он научился различать. На момент тестирования было обучено распознавать лица трех человек.

После выполнения всех манипуляций можно провести тестирование конкретно в работе с распознаванием лиц, заранее заданных в базу данных. Для начала будет произведено тестирование работы программы с 1 человеком на фотографии. Результат показан на рисунке 3.5.

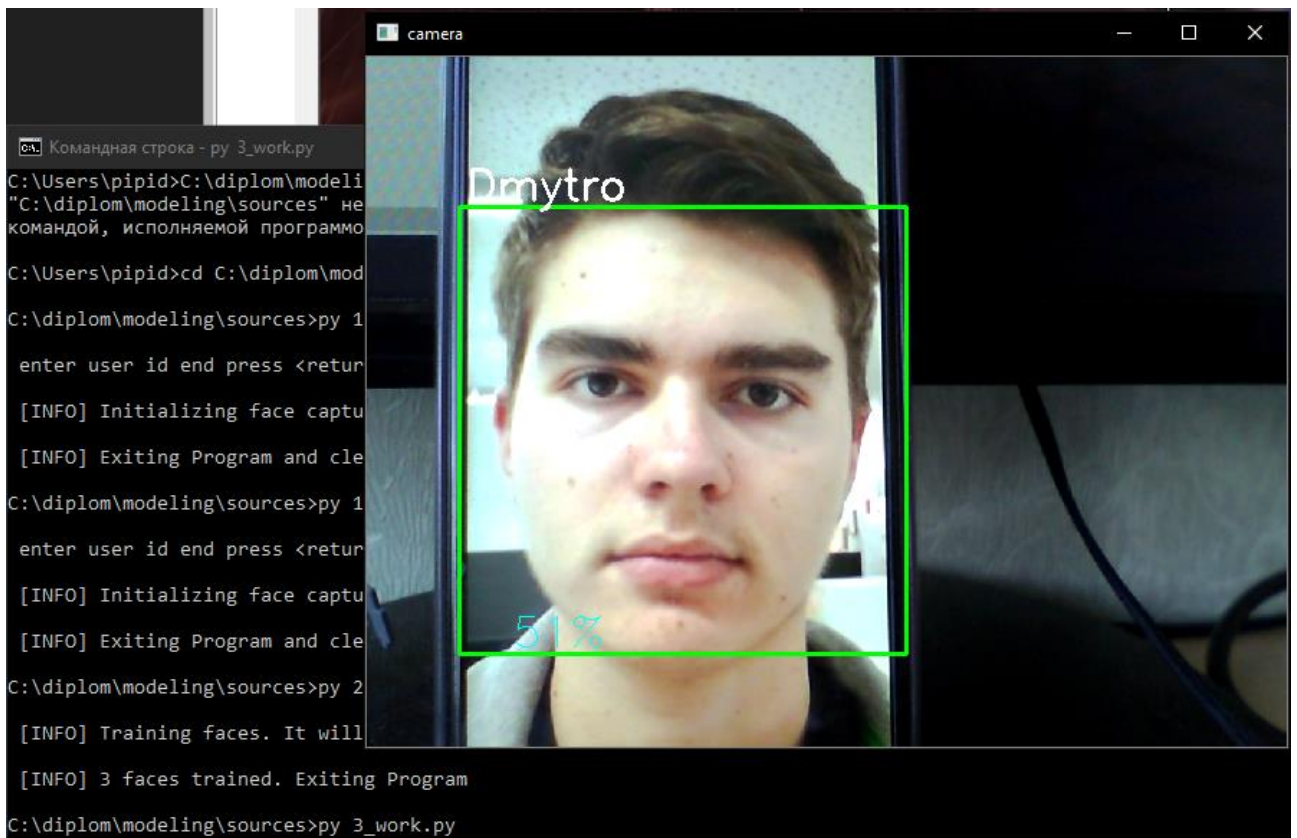


Рисунок 3.5 – Успешная идентификация пользователя

По результатам работы с одним человеком, программный продукт показал себя с отличной стороны. Во время тестирования программы с 1 пользователем в разных условиях освещения, программа практически всегда идентифицировала пользователя с высокой точностью, за несколькими исключениями, но они происходили по вине слабого оборудования, а именно из-за проблем с разрешением вебкамеры.

После нескольких успешных тестирований с одним пользователем, следует провести такое же тестирование, но с несколькими пользователями, которые были заранее добавлены в базу данных. Результат демонстрируется на рисунке 3.6.

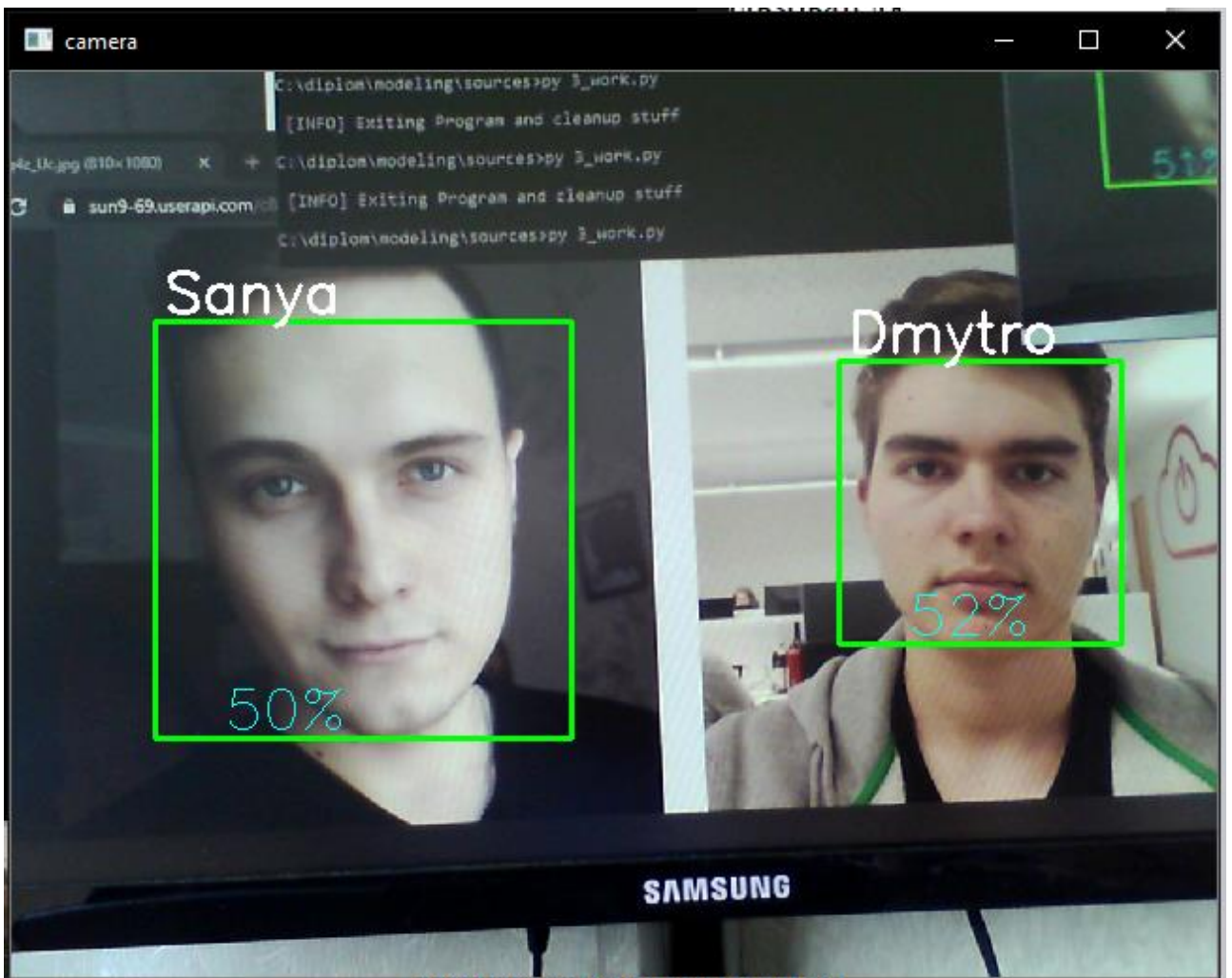


Рисунок 3.7 – Распознавание лиц нескольких пользователей одновременно

Теперь следует рассмотреть варианты с пятью людьми одновременно, но в различных условиях. На рисунке 3.8 демонстрируется пример работы программы при хороших условиях съемки.

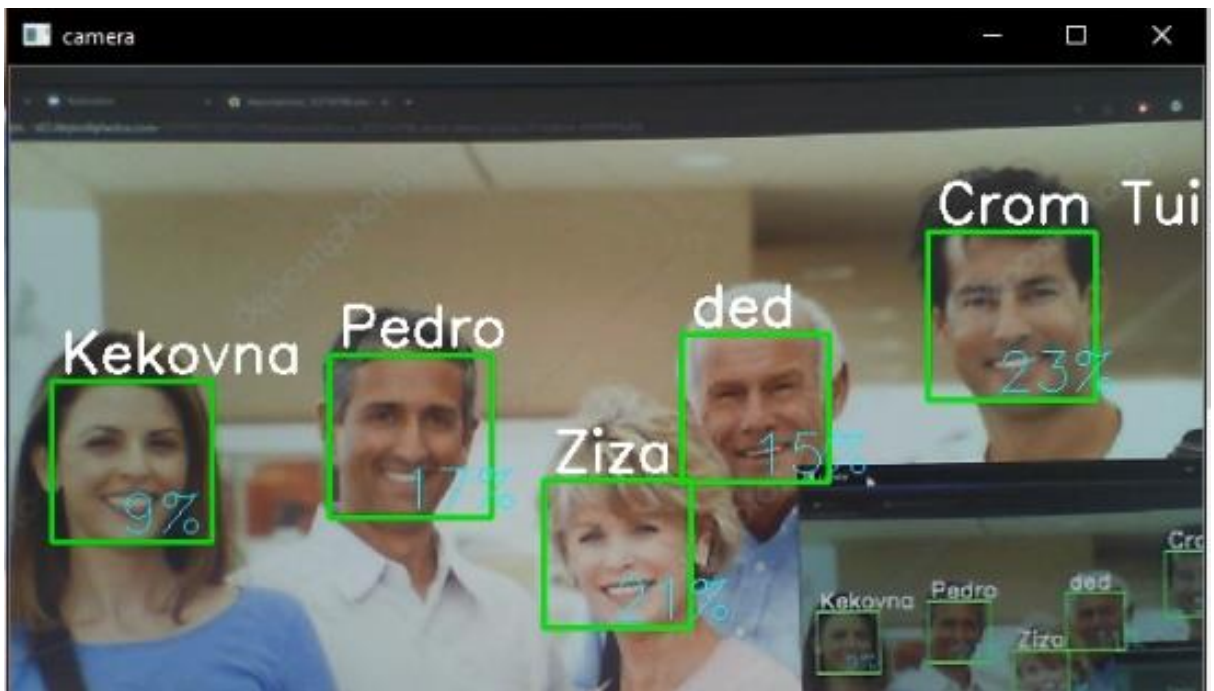


Рисунок 3.8 – Съемка пяти людей в хороших условиях

При данных условиях программа работала с высокой точностью. Теперь следует рассмотреть наихудший вариант, где выбран неудачный ракурс, а также плохое качество съемки. Пример показан на рисунке 3.9.

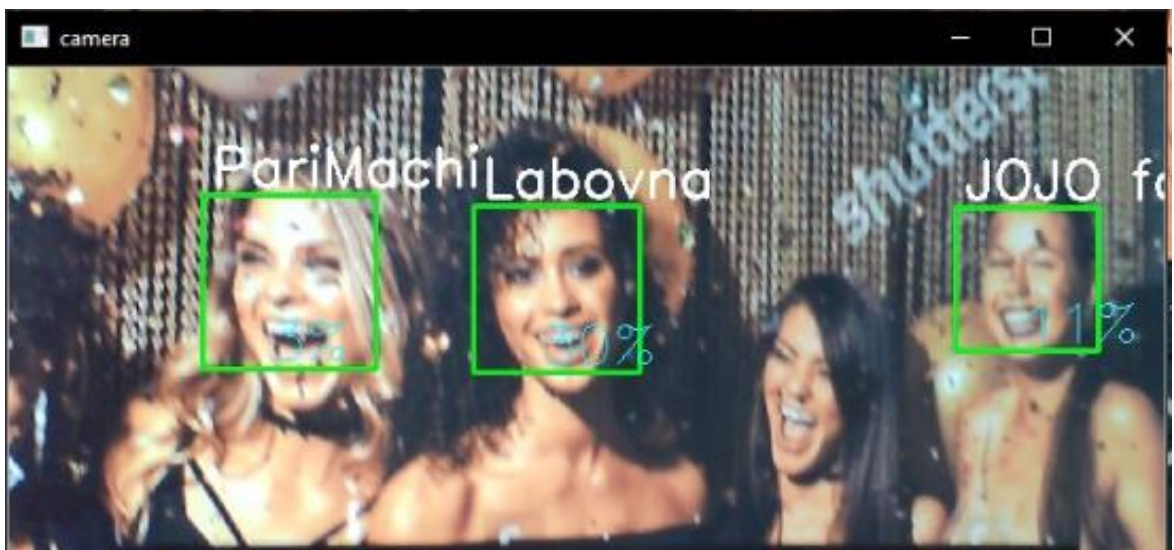


Рисунок 3.9 – Пример работы с неблагоприятными условиями

На данном примере четыре персоны запечатлены в размытой фотографии и подобран неудачный ракурс, из-за этого программа не смогла распознать одну персону.

В результате тестирование с несколькими людьми, программа показала себя с хорошей стороны. Из-за того, что машинное обучение не происходит в одну секунду, редко бывали проблемы с идентификацией пользователей, если их в одном кадре было больше пяти человек, но со временем при продолжительных тестах система давала с каждым разом все меньше ошибок.

В одно из достоинств данной программы можно отнести то, что если человека нет в базе данных, система его все равно распознает, но не сможет идентифицировать. Результат показан на рисунке 3.10.

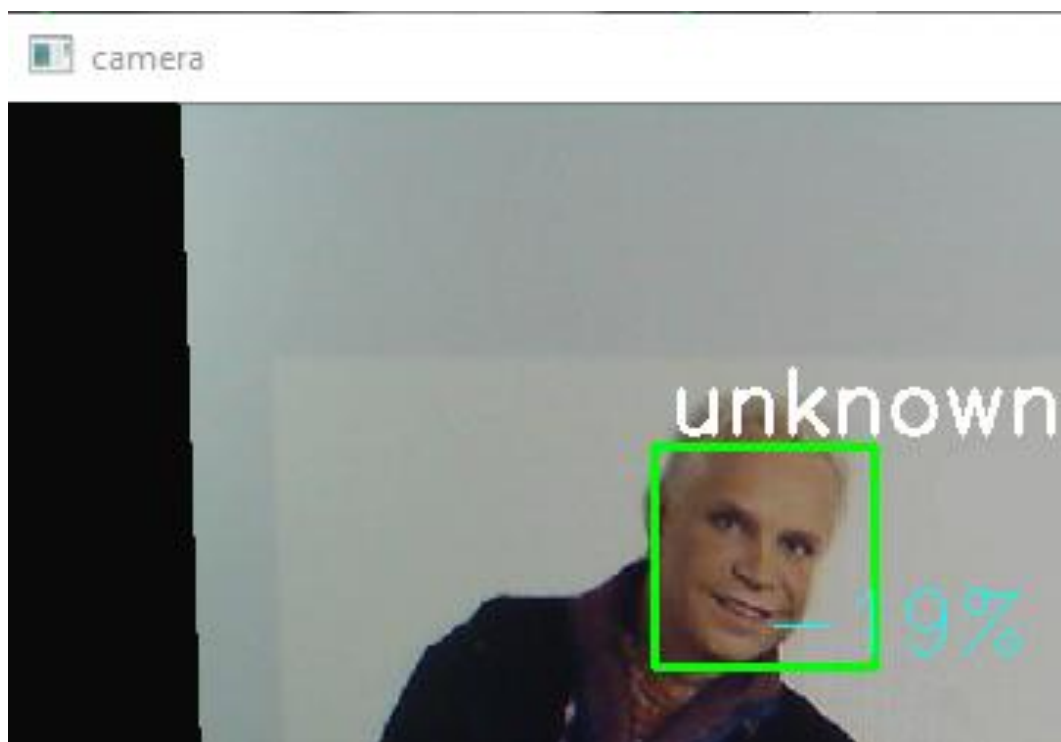


Рисунок 3.10 – Распознавание неизвестного алгоритму человека

Рассмотрев работу разработанного программного продукта, следует провести анализ его эффективности.

3.2 Проведение сравнительного анализа для определения эффективности работы программного продукта

В результате данной бакалаврской работы было реализовано программное обеспечение, которое моделирует портрет человека, а после

распознает его лицо и идентифицирует, применяя к ним разработанный алгоритм с использованием метода Хаара [19].

Тестирование разработанного алгоритма является важной частью реализации программного обеспечения. Для принятия решения о результате разработанного алгоритма необходимо провести ряд тестов, которые позволят проверить его работоспособность в реальных условиях.

В качестве оценки работы алгоритма, были проведены тестирования алгоритма моделирование и идентификация человека [14]. Ключевыми параметрами алгоритма восстановления является точность определения лица.

Для проведения тестирования было загружено в разработанный программный продукт набор изображений, где используется определенное количество людей при разных условиях съемки. Где-то условия съемки идеальны, а где-то либо фон неудачно подобран, где-то лица людей смазаны, а где-то съемка в ночное время. Было принято решение разделить все эти фотографии на три класса в тестировании, а именно:

- хорошее условие съемки;
- плохое освещение, где либо слишком много засветов, либо фотографии слишком темные;
- неудачный ракурс, где лица людей либо смазаны, либо их ракурс сделан так, что лицо плохо видно.

После будет произведен сравнительный анализ полученных результатов, на основе которого будет сделан вывод о проделанной работе.

В качестве тестируемого набора, был взят набор, состоящий из 900 изображений, которые удовлетворяли нужным критериям для теста, а именно наличие на фотографии пяти человек и наличие различных ракурсов и условий для съемки. На основе этих фотографий было произведено тестирование, где все фотографии прогоняли и записывались все удачные попытки, а именно попытки, где все лица были распознаны правильно. В таблице 1 приведены

результаты прогонки 900 фотографий, где 300 фотографий с хорошими условиями съемки, 300 с плохим освещением и 300 с плохим ракурсом.

Таблица 1 – Результаты тестовых прогонов фотографий в разработанном программном продукте в разных условиях.

	Количество прогонок									
	100	200	300	400	500	600	700	800	900	1000
Хорошее условие	189	228	249	261	273	279	280	288	292	294
Плохое освещение	163	172	177	192	206	233	252	269	278	288
Неудачный ракурс	113	132	144	169	182	214	238	247	259	269

На данной таблице видно, что с каждым запуском программа начинала определять лица людей на фотографии все точнее и точнее. При хороших условиях съемки программа довольно быстро смогла научиться определять нужные лица на фотографиях, а на 1000 прогон программа смогла определить всех людей на всех 300 входных изображениях, что говорит о том, что программа имеет высокую точность распознавания при хороших условиях.

Однако если фотографии слишком темные или же выбран неудачный ракурс, то точность распознавания становится ниже. Это видно в случае, если ракурс выбран неудачно. Здесь из 300 входных изображений программа смогла лишь на 269 определить всех людей, на остальных программа либо не смогла определить человека, либо определила неправильно и не смогла идентифицировать его.

Основываясь на полученных данных таблицы, следует составить график зависимости определения лиц участников в различных условиях съемки.

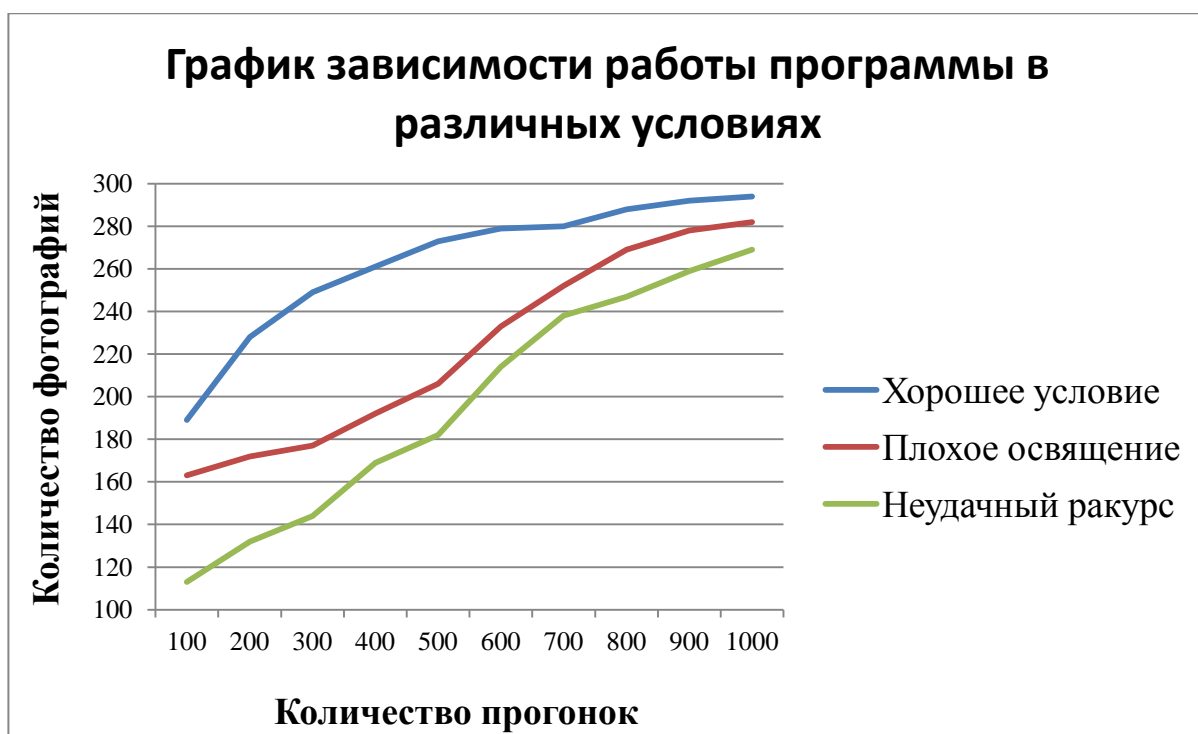


Рисунок 3.9 – График зависимости работы программы в различных условиях

Эффективность работы предложенного алгоритма оценивалась путем подсчета отношения количества правильно распознанных фотографий к общему числу всех фото. Эксперимент показал, что точность распознавания в хороших условиях после обучения составляет 98%, то есть после 1000 запусков программы 294 фотографии из 300 идентифицировали всех людей правильно с первого раза, в остальных были либо задержки, либо ошибки. При плохом освещении точность 96%, то есть 288 фотографий идентифицировали всех людей, а при неудачном ракурсе точность составляет 269 фотографий из 300 или же точность 89,67%.

При этом стоит отметить, что программа тестировалась в различных условиях, а именно:

- задний фон различной сложности;
- различное время суток;
- различное качество фотографий, полученных заранее или непосредственно с вебкамеры.

Как видно из результатов тестирования, предложенный алгоритм позволяет успешно решать поставленную задачу моделирования портрета человека при помощи нейронной сети. При этом, как видно, алгоритм анализа и идентификации человека успешно справляется с поставленной задачей даже в условиях недостаточной освещенности и с задним фоном различной сложности.

Выводы по третьему разделу:

В ходе тестирования программного продукта в реальных условиях были сделаны следующие выводы:

- программа успешно справляется с идентификацией как одного человека, так и нескольких одновременно;

- для более точного моделирования портрета человека нейронной сети нужно на 1 этапе получить не менее 30 фотографий одного пользователя, но, к сожалению, программа работает немного дольше, чем ожидалось;

- программа может распознавать лица и идентифицировать пользователей с задним фоном различной сложности;

- программа способна различать пользователей с различным освещением, все зависит только от рабочей техники.

Проведя тестирование на реальных примерах, можно перейти к заключению, где будут подведены итоги работы.

Заключение

Выполненная работа посвящена исследованию нейронных сетей в работе с распознаванием лица и его идентификации. Цель работы определила её основное направление – разработка алгоритма моделирования портрета человека с использованием нейронной сети, получение максимально возможной производительности одновременно выполняемых функций данного алгоритма, а также разработки программного продукта, использующего данный алгоритм.

Для этого были рассмотрены и проанализированы способы использования нейронных сетей в современных реалиях, которые работают с распознаванием.

В ходе анализа рассмотренных методов моделирования портрета человек с использованием нейронной сети были сформированы задачи для разрабатываемого алгоритма и программного продукта на его основе, а также сформированы критерии.

В процессе разработки были реализованы:

- алгоритм распознавания лица;
- блок схема для разработанного алгоритма;
- функция моделирование портрета человека с использованием нейронной сети;
- идентификация пользователя при заранее полученных данных;
- программный продукт, использующий разработанный ранее алгоритм.

К тому же были выявлены недостатки разработанного алгоритма, предложены способы по их устранению с описанием возможных последствий.

Данная система позволяет идентифицировать пользователя, моделируя заранее полученные фотографии и обучаясь, идентифицировать пользователей. Система достаточно требовательна к вычислительной мощности компьютера, так как подразумевает выполнение различных математических операции и

операций с моделированием, поэтому на этапе разработке и тестирования использовалось довольно мощное устройство.

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

Разработанный алгоритм был протестирован, были продемонстрированы результаты как в идентификации одного пользователя, так и нескольких одновременно, а также выявлены недостатки созданного программного продукта.

Для тестирования программного продукта на основе разработанного алгоритма был взят набор фотографий с разным количеством людей и с разными условиями съемки.

Все фотографии были протестированы через программную реализацию разработанного алгоритма. Разработанный программный продукт на основе составленного алгоритма показал лучшие результаты при хороших условиях съемки, определив людей с высокой точностью.

Полученные результаты разработки алгоритма, а также на его основе программного продукта, можно считать успешными, так как при тестировании не было выявлено никаких ошибок, которые могли бы привести процесс работы приложения и алгоритма к непригодности.

В результате работы над выпускной квалификационной работы был спроектирован алгоритм моделирования портрета человека с помощью нейронной сети и продемонстрирована его работа.

Список используемых источников

1. Бэрри, П. Изучаем программирование на Python / П. Бэрри ; [перевод с английского М.А. Райтман]. – Москва : Эксмо, 2017. – 624 с. : ил. ; Библиогр.: с. 182–228. – Заказ экз. – ISBN 978-5-699-98595-1. – Текст : непосредственный.
2. Элбон, К. Машинное обучение с использованием Python. Сборник рецептов / К. Элбон ; [перевод с английского СПб]. – Санкт-Петербург : БХВ-Петербург, 2019. – 384 с. : ил. ; Библиогр.: с. 139–238. – 1500 экз. – ISBN 978-5-9775-4056-8. – Текст : непосредственный.
3. Николенко, С. И. Глубокое обучение. Погружение в мир нейронных сетей / Николенко, С. И., Кадуринов А. А., Архангельская Е. О. – Санкт-Петербург : Питер, 2020. – 480 с. : ил. ; Библиогр.: с. 6–129. – 2500 экз. – ISBN: 978-5-4461-1537-2. – Текст : непосредственный.
4. Рашид, Т. Создаем нейронную сеть / Т. Рашид ; [перевод с английского А. Г. Гузикевич]. – Санкт-Петербург : Вильямс, 2018. – 272 с. : ил. ; Библиогр.: с. 129–212. – 1000 экз. – ISBN 978-5-9909445-7-2. – Текст : непосредственный.
5. Хайкин С. Нейронные сети. Полный курс / С. Хайкин ; [перевод с английского Н.Н. Куссуль]. – Москва : Вильямс, 2019. – 1104 с. : ил. ; Библиогр.: с. 42–71. – 200 экз. – ISBN 978-5-8459-2069-0. Текст : непосредственный.
6. ХАБР : Использование каскада Хаара для сравнения изображений : [сайт]. – Москва, 2020 – . URL: <https://habr.com/ru/post/198338> (дата обращения: 08.01.2020). – Текст : электронный.
7. OpenCV: OpenCV Tutorials : [сайт]. – USA, 2020 –. URL: https://docs.opencv.org/master/d9/df8/tutorial_root.html (дата обращения: 10.01.2020). – Текст : электронный.
8. ХАБР : Нейронные сети: практическое применение : [сайт]. – Москва, 2020 –. URL: <https://habr.com/ru/post/322392/> (дата обращения: 11.01.2020). – Текст : электронный.

9. DATАСИСТЕМ : Распознавание лиц : [сайт]. – Москва, 2020 –. URL: <https://www.hardbroker.ru/pages/recognition> (дата обращения: 15.01.2020). – Текст : электронный.
10. Hi-News.ru : Как это работает? | Система распознавания лиц : [сайт]. – Москва, 2020 –. URL: <https://hi-news.ru/eto-interesno/kak-eto-rabotaet-sistema-raspoznaniya-lic.html> (дата обращения: 15.01.2020). – Текст : электронный.
11. RUSBASE : Что можно узнать, отсканировав ваше лицо? : [сайт]. – Москва, 2020 –. URL: <https://rb.ru/story/face-recognition/> (дата обращения: 15.01.2020). – Текст : электронный.
12. ХАБР : Обучение машины — забавная штука: современное распознавание лиц с глубинным обучением : [сайт]. – Москва, 2020 –. URL: <https://habr.com/ru/post/306568/> (дата обращения: 20.01.2020). – Текст : электронный.
13. Neurohive : FaceNet — пример простой системы распознавания лиц с открытым кодом Github : [сайт]. – Санкт-Петербург, 2020 –. URL: <https://neurohive.io/ru/tutorial/raspoznvanie-lica-facenet/> (дата обращения: 02.03.2020). – Текст : электронный.
14. CyberLeninka : Моделирование на основе искусственных нейронных сетей как метод исследования в психологической науке : [сайт]. – Москва, 2020 –. URL: <https://cyberleninka.ru/article/n/modelirovanie-na-osnove-iskusstvennyh-neyronnyh-setey-kak-metod-issledovaniya-v-psihologicheskoy-nauke> (дата обращения: 05.03.2020). – Текст : электронный.
15. PYimageSearch : Face recognition with OpenCV, Python, and deep learning : [сайт]. – Baltimore, 2020 –. URL: <https://www.pyimageSearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/> (дата обращения: 15.03.2020). – Текст : электронный.
16. OpenFace : OpenFace : [сайт]. – Pittsburgh, 2020 –. URL: <https://cmusatyalab.github.io/openface/> (дата обращения: 17.03.2020). – Текст : электронный.

17. Dmitriy Azarov : Метод распознавания лиц Виолы-Джонса (Viola-Jones) : [сайт]. – Санкт-Петербург, 2020 –. URL: <https://oxozle.com/2015/04/11/metod-raspoznavaniya-lic-violy-dzhonsa-viola-jones/> (дата обращения: 20.03.2020). – Текст : электронный.

18. KostyaKulakov : Обучение каскада Хаара на примере поиска символов автомобильного номера OpenCV : [сайт]. – Москва, 2020 –. URL: <https://kostyakulakov.ru/opencv-обучение-каскада-хаара/> (дата обращения 20.03.2020). – Текст : электронный.

19. ХАБР : Работа каскада Хаара в OpenCV в картинках: теория и практика : [сайт]. – Москва, 2020 –. URL: <https://habr.com/ru/company/recognitor/blog/228195/> (дата обращения 20.03.2020). – Текст : электронный.

20. Joseph Howse. Learning OpenCV 4 Computer Vision with Python 3: Get to grips, techniques, and algorithms for computer vision and machine learning 3rd Edition / Joseph Howse, Joe Minichino. – USA : Packt Publishing, 2020. – 372 с. : ил. ; Библиогр.: с. 15–189. – ISBN: 978-1789531619.

21. Alberto Fernández Villán. Mastering OpenCV 4 with Python: A practical guide covering topics from image processing, augmented reality to deep learning with OpenCV 4 and Python 3.7 1st Edition / Alberto Fernández Villán. – USA : Packt Publishing, 2019. – 532 с. : ил. ; Библиогр.: 49–93. – ISBN: 978-1789344912.

22. Dr. Gary Rost Bradski. Learning OpenCV: Computer Vision with the OpenCV Library 1st Edition / Dr. Gary Rost Bradski. – USA : O'Reilly Media, 2008. – 580 с. : ил. ; Библиогр.: 228–322 с. – ISBN: 978-0596516130.

23. Jake VanderPlas. Python Data Science Handbook: Essential Tools for Working with Data 1st Edition / Jake VanderPlas. – USA : O'Reilly Media, 2016. – 550 с. : ил. ; Библиогр.: 21-87 с. – ISBN: 978-1491912058.

24. Aurélien Géron. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 2nd

Edition / Aurélien Géron. – USA : O'Reilly Media, 2019. – 849 с. : ил. ; Библиогр.: 190–839 с. – ISBN: 978-1492032649.

25. Andreas Müller. Introduction to Machine Learning with Python: A Guide for Data Scientists 1st Edition / Andreas Müller. – USA : O'Reilly Media, 2016. – 402 с. :ил. ; Библиогр.: 77–199 с. – ISBN: 978-1449369415.

26. Sebastian Raschka. Python Machine Learning - Second Edition: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2nd Edition / Sebastian Raschka, Vahid Mirjalili. – USA : Packt Publishing, 2017. – 624 с. :ил. ; Библиогр.: 298–579 с. – ISBN: 978-1787125933.

27. François Chollet. Deep Learning with Python 1st Edition / François Chollet. – USA : Manning Publications, 2017. – 384 с. :ил. ; Библиогр.: 25–140 с. – ISBN: 978-1617294433.

28. Michael Beyeler. Machine Learning for OpenCV: Intelligent image processing with Python / Michael Beyeler. – USA : Packt Publishing, 2017. – 382 с. :ил. ; Библиогр.: 201–360 с. – ISBN: 978-1783980284.

29. Ivan Idris. Numpy Beginner's Guide 2nd Edition / Ivan Idris. – USA : Packt Publishing, 2013. – 310 с. :ил. ; Библиогр.: 60–99 с. – ISBN: 978-1782166085.

30. Marc Peter Deisenroth. Mathematics for Machine Learning / Marc Peter Deisenroth. – UK: Cambridge University Press, 2020. – 398 с. :ил. ; Библиогр.: 87–256 с. – ISBN: 978-1108455145.

Приложение А

Исходный код разработанного программного обеспечения

Режим доступа к программному обеспечению, разработанному на основе составленного алгоритма восстановления расфокусированных и смазанных изображений. Исходный код программы доступен для скачивания в электронном виде.

Режим доступа:

<https://drive.google.com/open?id=16YbBul6MzmPQ6ldU1Cn-sI6oj9UMykrM>