



## АНОТАЦИЯ

52стр., 13 рисунков, 2 таблицы

### МАШИННОЕ ОБУЧЕНИЕ, ИСКУССТВЕННАЯ НЕЙРОННАЯ СЕТЬ, РАСПОЗНАВАНИЕ ОБРАЗОВ, ОБУЧЕНИЕ

Большинство стран мира заинтересованы в сохранение одного, наверное, из самых важных ресурсов - человека. Население земли стремительно приближается к 8 млрд. человек. И сейчас как никогда остро стоит вопрос о безопасности граждан страны. Угроза жизни или утрата работоспособности в связи нападения людей с преступными помыслами. В этом нам могут помочь информационные технологии.

Объектом исследования работы стал процесс распознавания человека по биометрическим данным. Предметом исследования является выявление проблем распознавания человека по его биометрическим данным.

В данной выпускной бакалаврской работе предложен метод решения поставленной задачи. Решение содержит: стороннюю библиотеку EmguCV и свёрточная нейронная сеть. Работа выполнена на объектно-ориентированном языке программирования с#.

Классификационная модель обучалась на фотографиях актеров сыгравших главную роль в американском сериале друзья и протестирована на видео взятом из интернета.

Внедрение разработанной системы позволит полиции идентифицировать личностей, которые переступили через закон.

## ANNOTATION

52 pages, 13 pictures, 2 tables

MACHINE LEARNING, ARTIFICIAL NEURAL NETWORK, PATTERN RECOGNITION, LEARNING

Most countries of the world are interested in the preservation of one, perhaps the most important resource - a person. The world's population is rapidly approaching 8 billion people. And now more than ever, the issue of the safety of citizens. Threat to life or loss of performance due to the attack of people with criminal thoughts. Information technologies can help us in this.

The object of the research work was the process of recognizing a person from biometric data. The subject of the study is to identify the problems of recognition of a person according to his biometric data.

In this graduation work is proposed a method for solving the problem. The solution contains: a third-party library EmguCV and convolutional neural network. The work is done in an object-oriented programming language c sharp.

The classification model was trained in the photographs of the actors who played the main role in the American TV series friends and tested on video taken from the Internet.

The implementation of the developed system will allow the police to identify individuals who have violated the law.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	5
1 ТЕОРЕТИКО-МЕТОДОЛОГИЧЕСКИЕ ОСНОВЫ РАСПОЗНАВАНИЯ ОБРАЗОВ .....	8
1.1 Ведение в теорию распознавания образов .....	8
1.2 Методы машинного обучения, распознавания образов и компьютерного зрения .....	11
1.3 Архитектура искусственной нейронной сети .....	15
1.4 Обоснование выбора типа нейронной сети .....	18
1.5 Обучение нейронных сетей .....	22
1.6 Современные гибридные архитектуры глубокого обучения .....	24
1.7 Сравнение библиотек глубокого обучения .....	25
1.8 Алгоритм эмоциональной системы кодирования лицевых движений для распознавания эмоций человека .....	26
2 АНАЛИЗ ИНСТРУМЕНТОВ ДЛЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ .....	30
2.1 Архитектура библиотеки EmguCV .....	30
2.2 Архитектура свёрточной нейронной сети .....	31
3 ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ПРОГРАММЫ .....	36
3.1 Разработка нейронной сети программными средствами .....	36
3.2 Обучение и проверка обученности нейронной сети .....	37
3.3 Апробация работы реализованной нейронной сети .....	38
ЗАКЛЮЧЕНИЕ .....	41
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ .....	43
ПРИЛОЖЕНИЕ А Таблица кодов двигательных единиц и двигательных дескрипторов .....	45

## ВВЕДЕНИЕ

В современных система по распознаванию биометрических признаков человека есть ряд не решенных задач. Одна из таких проблем — это их точность распознавания.

Требуется построить нейронную сеть с точностью распознавания более 80%.

Структуру построенной нейронной сети необходимо оптимизировать: для заданного объекта и регулятора найти оптимальное число слоев нейронов в каждом слое и оптимальную функцию активации.

Нейронная сеть должна во время своей работы обеспечивать требуемое качество распознавание объектов.

Для распознавания образов по фотографии в данный момент используют системы использующие технологии искусственного интеллекта (англ. Artificial Intelligence) иногда сокращают до ИИ (англ. Сокращение AI). Алгоритмы имитирующие искусственный интеллект способны запоминать и обучатся так же как это делают живые организмы.

Такие алгоритмы используются в разных сферах деятельности, а не только в задачах по распознаванию образов по фотографиям к примеру, в науке алгоритму нашлось место в нахождении новых материалов. Точность таких систем по последним данным достигает 80% в сравнении у ученых точность нахождения новых материалов составляет около 30%. Система обучалась на данных которые предоставили ученые и смогла не только не хуже работать, а почти утроила точность нахождения по сравнению с человеком. Место для ИИ нашлось и в сфере бизнеса так по данным аналитического агентства Gartner внедрение искусственного интеллекта помогло улучшить качество обслуживания клиентов и заработало компаниям внедривших ИИ к себе в сумме на 700 миллиардов долларов США. Не отстают и системы развлечений где алгоритмы искусственного интеллекта подбирают фильмы музыку или компьютерные игры по вкусу определенного пользователя основываясь на его прошлых запросах.

Способность распознавание образов по фотографиям получила широкое распространение в таких сферах как медицина ИИ используется для распознавания легочных болезней по снимку конечно в данный момент окончательное решение все еще стоит за врачом, но они существенно помогли снизить нагрузку на врачей, в машина строение ИИ встраиваются в системы автопилотирования машин на дорогах общего пользования, в сфере безопасности ИИ помогают идентифицировать человека, пытающегося получить доступ.

Актуальность выбранной тематики заключается в распознавание человека, и попытка определить его эмоционального состояния с целью обеспечение безопасности граждан.

Современные нейронные сети, к сожалению, не способны распознавать образы со 100% точностью. Если и появляются такие сети, то они скорей всего работают с идеальным исходным изображением. Здесь есть ряд подводных камней:

- Объект распознавания виден не полностью, он может быть под углом;
- Изображение содержит шумы, которые сбивают сеть.

В рамках данной выпускной бакалаврской работе стояла задача разработки системы по распознаванию образа человека и предложить методы которые могут помочь в решении повышения эффективности работы системы.

Объектом исследования бакалаврской работы является программное обеспечение по распознаванию биометрических признаков человека. Предмет исследования бакалаврской работы является распознавание человека.

Целью данной работы является создание нейронной сети по распознаванию биометрических признаков человека.

Для достижения поставленной цели нужно требуется решить следующие задачи:

1. Рассмотреть существующие алгоритмы по обучении алгоритма по распознаванию образов в сфере компьютерного зрения, провести анализ уже существующих решений на рынке.

2. Подробно расписать метод, который будет решать поставленную задачу.
3. Реализовать выбранный метод на языке программирования высокого уровня.
4. Проанализировать эффективность работы приложения. Предложить пути решения возможных проблем.
5. Сравнить результаты системы по распознаванию образов с теми, что уже представлены на рынке.

Система по распознаванию человека по биометрическим признакам может работать для обеспечения безопасности отдельных объектов или для поиска преступников на улице.

Классические алгоритмы построения программ не помогут при создании программ по определению биометрических признаков человека. Так как потребовалось бы хранить огромное количество информации, и оно бы не гарантировало работу приложения. Здесь на помощь приходят нейронные сети и их способность абстрагировать данные и выдавать результат, максимально подходящий нам. Остается лишь проблема с точностью определения.

Данная работа содержит: введение, три основные главы и заключение.

Во введении описывается актуальность выбранной темы, формулируются цель и задачи исследования, указываются объект и предмет исследования. Первая глава посвящена исследованию теоретических вопросов, этапам исторического развития нейронных сетей. Во второй главе дается описание архитектуры необходимого инструментария. В заключении подведены итоги и сделаны выводы исследования.

# 1 ТЕОРЕТИКО-МЕТОДОЛОГИЧЕСКИЕ ОСНОВЫ РАСПОЗНАВАНИЯ ОБРАЗОВ

## 1.1 Ведение в теорию распознавания образов

Для реализации программного обеспечения по распознаванию образов человека нужно дать определения разделам, на основе которых будет выстроена нейронная сеть по распознаванию биометрических признаков человека. Основой программного обеспечения выступает возможность ее обучения.

**1. Машинное обучение** (англ. Machine Learning) – обширный подраздел искусственного интеллекта, изучающий методы построения алгоритмов и систем результаты, которых возрастают по мере накопления опыта.

Машинное обучение разделяется на два типа:

1. Обучение по прецедентам или индуктивное обучение основывается на выявлении общих закономерностях эмпирических данных.
2. Дедуктивное обучение основывается на данных предоставляемыми экспертами в виде базы данных.

Дедуктивное обучение принято относить 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 науке о искусственном интеллекте и на методы обучения сетей и виды архитектур сетей. Также машинное обучение вобрало в себя математическую статистику, методы оптимизации и классические математические дисциплины помимо этого машинное обучение имеет и свою собственную специфику, связанную с проблемой вычислительной эффективностью и переобучение сети.

Обучение разделяется на следующие типологии задач:

- Обучение с учителем – каждый прецедент представлен в виде «задача, ответ»;
- Обучение без учителя – система ищет сходства в объектах и образует кластеры из них;



- Частичное обучение – каждый прецедент, как и в обучении с учителем представляется как пара «задача, ответ», но ответ известен лишь для части из них;
- Трансдуктивное обучение – даны данные о предыдущих выборках нужно на основе ее сделать предсказания относительно других данных;
- Обучение с подкреплением – данные представлены в виде пар «ситуация, выбранное решение» ответами являются значения функционала качества;
- Динамическое обучение – данные могут поступать потоком и по каждому прецеденту принимать не за медлительные решения. Обучение может быть с учителем и без учителя.
- Активное обучение – отличается лишь тем, что следующий претендент может быть назначен самостоятельно и быть известным;
- Метаобучение – прецеденты были раньше решеными задачами обучения.

Любой из этих способов может стать основой для машинного обучения и классификации алгоритмов обучения нейронных сетей.

Современные машинное обучение имеют крупный недостаток это их не универсальность и под каждую отдельную задачу нужно тренировать новую сеть.

**2. Теория распознавания образов и компьютерное зрение** относится к разделу информатики и смежных дисциплин. Этот раздел занимается методами классификации и идентификации объектов, процессов, явлений, ситуаций и сигналов и т.п. которые можно охарактеризовать каким-то общим набором признаков и свойств.

Задачи по распознаванию образов не редко попадают в области разработок искусственного интеллекта, а также в области задач компьютерного зрения к которым относится данная задача, рассматриваемая в рамках данной квалификационной работы.

При классической постановке задачи распознавание образов происходит путем использования строго математического языка, в основе которых лежат логические рассуждения и математические доказательства. Альтернативным подходом классическому является использование искусственных нейронных сетей и машинного обучения. Результаты обоих подходов дают примерно одинаковый результат, но в некоторых ситуациях с лучшей стороны себя показывает альтернативный подход к распознаванию образов.

Компьютерное зрение относится к теории и технологии создания машин, в возможности которых присутствует обнаружение, отслеживание и классификация объектов наблюдения. Данные для такой системы могут быть разными от видео последовательных снимков, снимки с различных камер или быть представлены в виде трех мерных данных.

Компьютерное зрение занимает большую роль при разработке автономных систем высоко технологического уровня и совершающие какие бы то ни было механические движения нуждаются в данных о среде в которой они находятся. По этой причине компьютерное зрение и теория распознавания образов тесно связаны в обработке сигналов. Методы обработки одномерных сигналов могут быть расширены до многомерного уровня. Используя теорию компьютерного зрения, геометрии и методов оптимизации.

Компьютерное зрение также связано с областями обработки изображений и машинного зрения. Обработка изображений занимается анализом и построением изображений на основе других изображений с использованием математических методов. Данные операции подразумевают, что анализ или обработка изображения не зависят от содержания на изображениях. Машинное зрение часто рассматривают как раздел более общей теории компьютерного зрения. Машинное зрение нашло свое применение в промышленности.

По сегодняшний день нет стандартной формулировки как нужно решать проблемы, связанные с компьютерным зрением. Но существуют методы, которые решают проблемы узко направленных задач компьютерного зрения.

## 1.2 Методы машинного обучения, распознавания образов и компьютерного зрения

Основными этапами машинного обучения можно выделить следующие этапы работы с данными:

- Фильтрация исходных данных;
- Анализ данных;
- Обучение системы по исходным данным.

**Фильтрация данных** в рамках задач по компьютерному зрению используется для предварительной обработки изображений перед тем, как провести морфологический анализ.

Чаще всего под фильтрацией понимается очистка исходного изображения от возможных шумов. При этом предполагается что есть копия нашего изображения, которое лишено все возможных шумов, а мы лишь имеем копию, полученную методами зашумления. Таким образом задача фильтрации сводится к тому, чтобы получить копию изображения без так называемых шумов.

Процедура пороговой бинаризации часто используется в обработках изображения – при работе, которой исходное изображение делится на две области. Одна из них содержит все пиксели ниже некоторого порога, а другая напротив содержит все пиксели выше этого порога.

Преобразование Фурье является частным случаем гауссовых функций. На его основе как правило строят более сложные типы фильтров. К недостатку преобразования Фурье можно отнести то, что для обработки изображений как правило не хватает одномерного преобразования, а двумерное преобразование является достаточно ресурсоемким процессом. Это частично решается быстрым преобразованием Фурье за счет однократного выполнения действия для разных пар  $u, v$ . Но такой тип преобразования можно применить лишь к изображениям имеющие размер  $2^m \times 2^m$ . Несмотря на то, что преобразование Фурье распространен при фильтрации изображений он имеет крупный недостаток

того что можно потерять контрастные объекты на изображении. В связи с тем, что алгоритм может посчитать их посторонними шумами и отфильтровать.

Фильтры частот делятся на два типа работы с высокими частотами означающие резкое изменение яркости пикселей и соответственно с низкими частотами означающие более мягкий переход яркости от пикселя к пикселю. Обработка происходит в рамках размера фильтра. Фильтр также получил большое распространение за счет свойств быстрого шума подавления.

Вейвлет-преобразование — это интегральное преобразование главной задачей которой выступает свертка вейвлет-функции с сигналом. Вейвлет — это обобщённое математической функции определенной формы и которые были получены из одной базовой функции посредством ее сдвига и растяжением по оси времени. Вейвлеты используются для классификации и сжатия изображений. Однако при малых масштабах сжатия вейвлет-преобразование начинает уступать в качестве оконному Фурье-преобразованию. В отличие от Фурье-анализа вейвлет-анализ опирается на вейвлеты ограниченные по времени это позволяет анализировать изображение сразу в нескольких масштабах и носит название кратномасштабный анализ.

Так называемые фильтры функций используют математические фильтры для обнаружения на изображении простые математические функции. В следствие чего формируется так называемое аккумулялирующее изображение, где для каждой из точек исходного изображения будут построены множество порождающих ее функций. Одной из известнейших алгоритмов является модификация обобщённое преобразование Хафа дающая возможность нахождения любых параметризуемых функций. Но применять этот алгоритм можно лишь на бинаризованное изображение и имеет высокую чувствительность к качеству бинаризации.

При обработке сложных объектов на изображении к нам на помощь приходят фильтры контуров. Наиболее известный из алгоритмов реализующий фильтр контуров носит название оператор Кенни. В рамках компьютерного зрения при обработке изображений и поиске границ объектов помимо частных

случаев сложно найти детектор, который работал бы лучше, чем оператор Кенни. Помимо оператора Кенни нужно отметить существование таких детекторов как оператор Лапласа, оператор Прюитт, оператор Собеля, и оператор Робертса.

Фильтры эрозии и наращивания - эти фильтры работают по одному алгоритму действия. Фильтр наращивания увеличивает яркость ярких объектов путем выборки пикселя с наибольшей интенсивностью из окрестности. Эрозия для темных объектов из определенной окрестности выбирает пиксель с наименьшей интенсивностью что приводит к его затемнению.

Все вышеописанные фильтры позволяют решить подавляющее большинство задач, связанных с компьютерным зрением. Помимо данных фильтров существуют и десятки других, которые являются более узко специализированными фильтрами. На практике такие фильтры встречаются крайне редко.

Следующим этапом машинного обучения стоит **анализ данных**. При помощи фильтрации мы получаем данные, которые готовы к дальнейшей обработке, но задачи в области компьютерного зрения требуют анализа внутренней структуры изображения и морфологического признака. Для этих целей вводят методы логического анализа изображения.

Математическая морфология - теория и техника в задачу которого входит обработка геометрических структур. В основе математической морфологии лежит теория множеств, топологий и случайных функций. Операция математической морфологии оказались полезными в работе над множеством двумерных точек.

Контурный анализ - позволяет описывать и находить объекты и хранить их внешние контуры. Контурный анализ относится к детерминированным методам распознавания образов и имеет один существенный недостаток — это его чувствительность к шумам на исходном изображении по этой причине важно произвести фильтрацию изображения перед тем, как приступить к контурному анализу. Ранее был описан метод Кенни результатом работы

которого возвращались контуры объектов без необходимости дополнительной бинаризации объектов на изображении. Для контурной задачи обязательной прикладной задачей является бинаризация объектов. При использовании контурного анализа можно перейти от двумерного пространства к пространству контуров так как внутренние точки для нас стали не важны в связи чего алгоритмическая сложность некоторых задач будет упрощена. Доказано что контурный анализ инвариантен к масштабируемости и углу поворота объекта.

Контурный анализ достаточно чувствителен к среде, в которой производится анализ, что накладывает ограничение на их использование в реальных условиях для большинства задач компьютерного зрения, но в задачах машинного зрения где условия среды строго ограничены контурный анализ доказывает свое быстрое действие.

**Метод обучения с учителем** применяется в случае, если входные данные и результаты работы заранее известны. Обучаемая сеть меняет свои параметры работы для достижения необходимых результатов.

**Метод без учителя** применяет реже особенность такого обучения что результаты работы не известны. Задача сети научиться кластеризовать данные по каким-то общим признакам прогнав через себя большой объем входных данных.

**Обучение с подкреплением** применяется тогда, когда можно будет оценить результаты работы нейронной сети. Пока результат максимально приближен к необходимому результату ей дается возможность искать любые пути решения поставленной задачи, до тех пор, пока результаты остаются удовлетворительными.

Разрабатываемое нейронную сеть стоит обучать методом без учителя. Благодаря чему она научится выявлять общие черты для человека с использованием фильтров, которые смогут повысить точность распознавания.

### 1.3 Архитектура искусственной нейронной сети

Искусственная нейронная сеть или просто ИИС – математическая модель, а также программное и аппаратное воплощение имитирующая принцип организации нейронных связей биологических особей. Соединены нейроны в искусственных нейронных сетях при помощи так называемыми синаптическими связями. Сеть принимает какую-то информацию и в процессе своей работы готовит выходную информацию. Нейронные сети не программируются в привычном понимании этого слова, а обучаются по ходу обработки новой информации.

Технология искусственных нейронных сетей за свою историю нередко меняя представление о предмете и проблематики распознавания образов и теории машинного обучения. Также оказывал значительное влияние на теоретический терминологический и методологические аппараты этих дисциплин. За свою историю направление по построению искусственных нейронных сетей переживала как подъемы, так и спады интереса к себе, но сравнительно недавно в связи некоторых прорывов в сфере искусственных нейронных сетях, эта технология получила широкое практическое применение.

Искусственный нейрон (рисунок 1) создан по принципу биологического нейрона, но ученые до сих пор не удалось составить полную математическую модель биологического нейрона в связи его сложности. Не смотря на простоту структуры искусственного нейрона (Формальный нейрон) сеть, построенная из множества таких нейронов, может выдавать произвольную многомерную функцию.

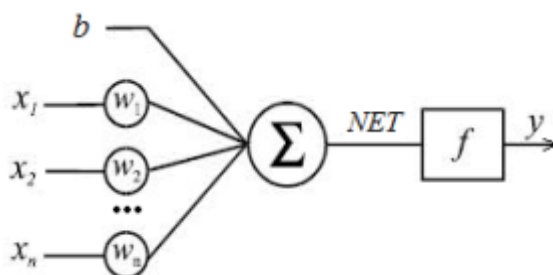


Рисунок 1 - Структура Искусственного нейрона

Модель формального нейрона

$$1. NET = \sum_{n=1}^n w_x * x_n + b \quad (1)$$

$$2. y = \begin{cases} 1, & \text{если } NET \geq T \\ 0, & \text{в остальных случаях} \end{cases} \quad (2)$$

где

$x_n$  – n-й элемент вектора входного сигнала;

$w_n$  - весовой коэффициент n-го входа;

$b$  – смещение;

$T$  – величина порога;

$y$  – выходной сигнал нейрона.

Нейрон имеет несколько входных сигналов  $x$  и один выходной сигнал  $y$ . Параметрами нейрона, определяющими его работу, являются вектора весов  $w$ , пороговый уровень  $T$  и вид функции  $f$ .

Структура большинства архитектур искусственных нейронных сетей имеют фиксированные передаточные функции активаций нейронов, а веса относятся к параметрам сети. Входы некоторых формальных нейронов являются внешними входами совокупной сети, а некоторые выходы формальных нейронов являются выходами совокупной сети. Задачей искусственной нейронной сети стоит в том, чтобы преобразовать входной сигнал в выходной сигнал на основе топологии сети и значений весов.

Все виды нейронных сетей можно условно разделить на сети прямого распространения и на сети с обратными связями. На рисунке 2 представлены самые известные нейронные сети.



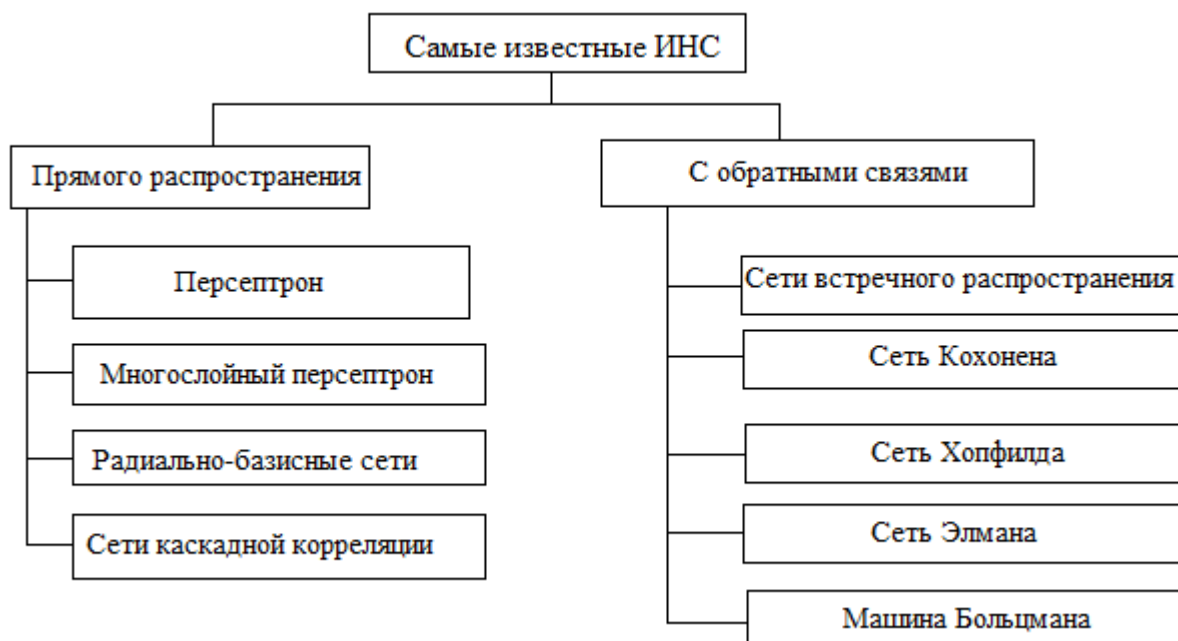


Рисунок 2 - Классификация распространённых видов ИНС

В сетях прямого распространения сигналы от нейрона к нейрону распространяются по однонаправленному направлению от входа к выходу.

Сети второго типа реализовывают рекуррентную структуру с использованием обратных связей, благодаря чему состояние сети зависит от предшествующего состояния.

Перцептрон – это простейшая форма нейронной сети для классификации линейно-разделимых сигналов. Перцептрон состоит из одного нейрона с настраиваемыми весами и порогом. Первая реализация появилась в 1957 году предложил модель американский ученый Фрэнк Розенблатт. Перцептрон стал первой моделью нейронной сети.

Многослойные перцептроны успешно справляются, решая разнообразные сложные задачи.

Нейронная сеть Кохонена – класс нейронной сети основой которой является слой Кохонена. Слой Кохонена состоит из линейных формальных нейронов. Слой Кохонена преобразует самый большой входной сигнал в единичный, а остальные в нулевой. Нейронные сети Кохонена

классифицируются по типам задач решаемые ими, а также по способу настроек весов.

В практическом использовании нейронные сети Кохонена можно встретить при решении задач, связанных с кластерным анализом, задачами моделирования, прогнозирования и сжатия информации.

## **1.4 Обоснование выбора типа нейронной сети**

### **1. Когнитрон, Неокогнитрон и Свёрточная нейронная сеть**

Когнитрон – это искусственная нейронная сеть на основе принципа самоорганизации. Когнитрон имеет иерархическую многослойную структуру где все нейроны между слоями связаны друг с другом лишь локально.

Биологическим прототипом когнитрона послужила зрительная кора, где были обнаружены узлы, реагирующие на такие элементы как линии и углы определенной ориентации. На более абстрактном уровне Узлы способны реагировать на более сложные образы. Абстракция возрастает с каждым уровнем до тех пор, пока не начнет распознавать такие сложные формы как лица.

Когнитрон обучается на основе принципа самоорганизации и в классическом случае реализуется при помощи алгоритма обучения без учителя.

Неокогнитрон является прямым наследником идеи когнитрона и более точно отражает строение зрительной системы, позволяет распознавать образы независимо от их преобразований, вращений, искажений и изменений масштаба. В результате чего неокогнитрон является более мощной парадигмой с точки зрения способности распознавания образов.

Неокогнитрон так же, как и когнитрон обучается без учителя и в основе его обучения лежит самоорганизация, хотя была описана версия, в которой вместо этого использовалась управляемая среда.

Свёрточная нейронная сеть была предложена французским ученым Яном Лекунеом в 1988 году. Свёрточная нейронная сеть продолжает идею неокогнитрона за основу которого была взята особенность зрительной коры.

Основой свёрточной нейронной сети лежит в так называемой операции свертки. Суть свертки лежит в том, что каждый фрагмент изображения умножается на матрицу свертки поэлементно, а результат суммируется и записывается в аналогичную позицию выходного изображения (рисунок 3).

В работа свёрточной нейронной сети происходит переход от конкретных особенностей изображения к более абстрактным деталям и таким образом алгоритм будет повторяться пока не выделит высокий уровень абстракции. В процессе работы сеть само настраивается, выделяя более важные детали и фильтруя при этом менее важные.

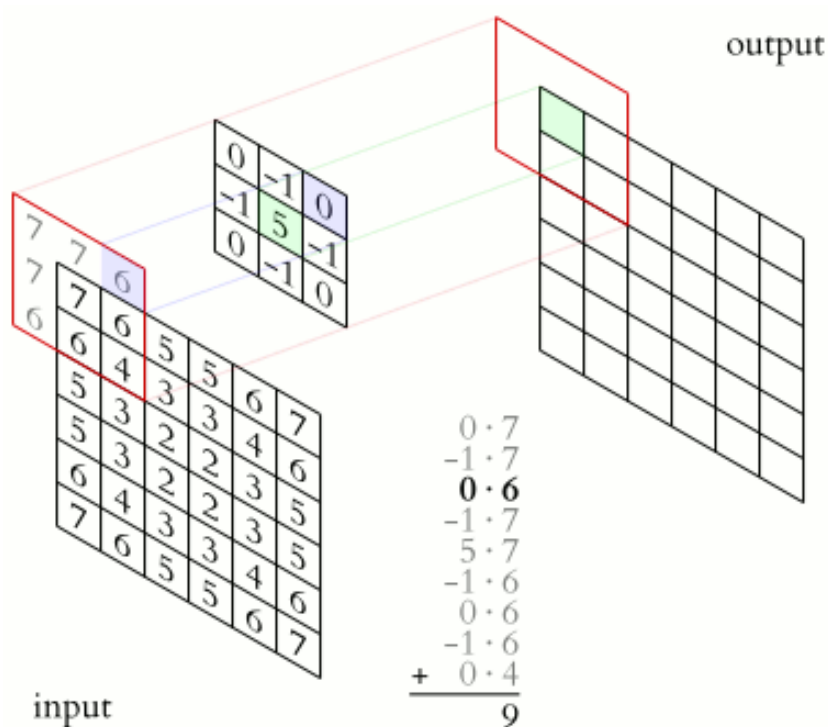


Рисунок 3 - Операция свертки

Свёрточные нейронные сети применяются для решений широкого круга задач к примеру: в промышленности, корпоративных, исследовательских, повседневных и даже в задачах решаемые мобильными устройствами.

## 2. Рекуррентная Нейронная сеть и Современные глубокие нейронные сети

Рекуррентная нейронная сеть – вид нейронной сети, где связи между элементами образуют направленную последовательность. В отличие от многослойных перцептронов, рекуррентные сети могут обрабатывать последовательности любой длины. Образуются в различных типах архитектур и различными видами обратных связей. К наиболее известным архитектурам рекуррентных нейронных сетей относятся: сети Хопфилда, сети Элмана, сети Джордана и сети Коско.

Основой для современных глубоких нейронных сетей выступают архитектура сетей свёрточного типа. Гибридный подход к архитектурным решениям, дополнительные методы защиты от переобучения и развития методов обучения обуславливает их эффективность и стремительное развитие.

Синапсы осуществляют связь между нейронами в сети умножая входной сигнал на вес синапса. Сумматор выполняет сложение всех входящих сигналов поступающих по синоптическим связям от других нейронов сети. Преобразователь реализует функцию одного аргумента результатом которого станет некоторая выходная величина. Такая функция называется функция активации нейрона. Нейрон реализует скалярную функцию векторного аргумента. Выход определяется видом функции и может быть, как целым, так и действительным. Выбор той или иной функции активации часто зависит от поставленной задачи и архитектуры сети. Некоторые из рассматриваемых придаточных функций применяются лишь в устаревающих системах или в целях обучения.

Пороговая функция Хевийса, пожалуй, является самой простой функцией активации. Эта функция использовалась в классическом перцептроне и на данный момент считается устаревшей. Используется только в качестве обучению теории нейронных сетей.

При использовании линейной функции в качестве функции активации выходной сигнал будет линейно зависим от взвешенной суммы входных сигналов. В данный момент линейная функция используется крайне редко.

Сигмоидальная функция активации является одной из самых распространённых функций ее часто можно встретить в современных нейронных сетях. Функция является монотонно возрастающей всюду дифференцируемой S-образной нелинейной функцией с насыщением. Функция активации обеспечивает усиление слабых сигналов и не позволяет насыщению сильных сигналов. Появление сигмоидальных функций было обусловлено недостатком гибкости классификаторов на основе пороговых передаточных функций. Что позволило перейти от жесткой одноразрядной логики к более гибкому поведению и адаптивной параметризации нейронных сетей.

$$x = \frac{1}{1 + e^{-x}} \quad (3)$$

где

$x$  – входной нейрон,

$e$  – основание натурального логарифма.

Гиперболический тангенс отличается от Сигмоидальной функции лишь тем что его интервал находится от  $(-1; 1)$ , что упрощает задачу по обучению искусственным нейронным сетям.

$$x = \frac{e^{2x} - 1}{e^{2x} + 1} \quad (4)$$

где

$x$  – вход нейрона,

$e$  – основание натурального логарифма.

В неглубоких искусственных нейронных сетях используются не линейные функции активации. Часто встречаются функции основанных на сигмоидальных и передаточных функциях являются не линейными, но на практике во время обучения искусственных нейронных сетей могут привести к проблеме затухания или увеличения градиента. Функция ReLU считается гораздо более простой и эффективной с точки зрения вычислительной сложности. Производная от ReLU равна либо 1, либо 0 от чего ее использование предотвращает разрастание или затухание градиента. Функция

ReLU является одной из последних успехов в области методов постройки глубоких нейронных сетей.

$$f(x) = \max(0, x) \quad (5)$$

где

$x$  – вход нейрона.

Также существуют разновидности функции ReLU, решающая проблему с надежностью передаточной функции при прохождении через нейрон больших градиентов: Parametric ReLU, Leaky ReLU, Randomized ReLU.

## 1.5 Обучение нейронных сетей

Обучение искусственных нейронных сетей – это настройка архитектуры и поиск наилучшего набора весов для максимизации точности предсказаний. Есть два крупных класса обучения искусственных нейронных сетей: класс детерминированных методов и класс стохастических методов.

К классу детерминированных методов относят методы исследования, в которых зависимость между рассматриваемыми признаками строго задана или детерминирована поэтому исследование приводит к однозначным выводам. Самым распространённым детерминированным методом обучения искусственной нейронной сети является метод обратного распространения ошибки.

К классу стохастических методов относят методы, которые изменяют параметры сети случайным образом и сохраняющие только те параметры, которые привели к улучшению результатов. Стохастические алгоритмы обучения реализуются с помощью сравнения ошибок и некоторые из них связаны с проблемой «ловушки локального минимума», решаемой с помощью некоторых усложнений стохастических алгоритмов.

Алгоритм ОРО основан на настройке сети таким образом, чтобы минимизировать среднеквадратичную ошибку. Веса и смещения изменяются в направлении противоположном направлению градиента. Поэтому базовый алгоритм ОРО еще методом градиентного спуска.

Базовый алгоритм ОРО в виде последовательности шагов:

1. Все веса и смещения инициализируются маленькими ненулевыми величинами обычно от -1 до 1;
2. Представление входного вектора  $p$  и желаемого выходного вектора  $t$ ;
3. Расчет фактического выхода сети  $a$  при подаче на вектор  $p$ ;
4. Настройка весов и смещение по формуле:

$$w_{ij}^k t + 1 = w_{ij}^k t + n\delta_i^k y_i^{k-1} \quad (6)$$

где

$\delta_i^k$  – вычисляется для выходного слоя по формуле:

$$\delta_i^k = f'(y_i^{k+1})(d_i^{k+1} - y_i^{k+1}), \quad (7)$$

а для скрытых слоев по рекуррентной формуле:

$$\delta_i^k = f' y_i^k \sum_j \delta_j^{k+1} w_{ij}^{k+1} \quad k = 1, \dots, K \quad (8)$$

Здесь  $n$  – скорость обучения (небольшая положительная величина, берущая свое начало из интервала от 0 до 1),  $f' y_i^k$  – производная активационной функции  $i$ -го нейрона, находящегося на  $k$ -м слое. Если используются нейроны с сигмоидной функцией активации, то для них  $f' y_i^k = y_i^k (1 - y_i^k)$ . Процедура настройки весов происходит от выходного слоя к входному.

5. Если не удовлетворено условия окончания обучения, то переход ко 2 шагу.

Генетический алгоритм – это эвристический алгоритм поиска, используется для решения задач оптимизации и моделирования путем случайного подбора, комбинирование и вариации искомых параметров с использованием механизмов, аналогичных отбору в природе. Генетические алгоритмы применяются в искусственных нейронных сетях в редких случаях, их применение обусловлено сложностью или невозможностью сформулировать задачу в том виде в котором требуют ее другие более быстрые алгоритмы локальной оптимизации.

Гиперпараметры – это все «степени свободы» алгоритма, которые он напрямую не оптимизирует, но от которых зависит результат. Гиперпараметры можно выставить или на значения по умолчанию или подобрать вручную, но иногда неудачная конфигурация может все испортить и алгоритм либо сильно переобучится или наоборот не сможет использовать информацию до конца.

Исключение или дропаут является простым методом регуляризации искусственных нейронных сетей, предназначен для предотвращения переобучения сети. В процессе обучения сети выбирается слой нейронной сети из которой выбрасывается случайное количество нейронов с определенным шансом (называется коэффициентом дропаут) которые перестают участвовать в дальнейшем обучении. Такой прием улучшает эффективность обучения и качество результата.

Помимо описанных методов регуляции весов существуют методы в основе которых лежит ограничение или штрафы весов.

## **1.6 Современные гибридные архитектуры глубокого обучения**

Softmax-слои современных искусственных нейронных сетей для моделирования вероятностного распределения, включающие  $N$  нейронов по целевому количеству классов. Выход каждого нейрона зависит от сумматоров всех остальных нейронов в слое.

Современные искусственные нейронные сети используют в своей архитектуре субдискретизирующие слои, задача которых состоит в уменьшении входной карты признаков. Чаще всего используют метод выбора максимального элемента. Использование таких слоев позволяет сделать сеть устойчивой к масштабным преобразованиям.

Первая свёрточная сеть была предоставлена в 90-х годах и носит название LeNet (рисунок 4). Разработал ее французским ученым Ян ЛеКун. Обучалась такая сеть распознавать рукописные символы. Последней версией этой сети стала LeNet-5 (рисунок 5) которая демонстрировала впечатляющие результаты на тот момент времени за счет содержания нескольких новаторских идей.



Концептуальным отличием такой сети состояло в наличие субдискретизирующего слоя после каждого свёрточного слоя.

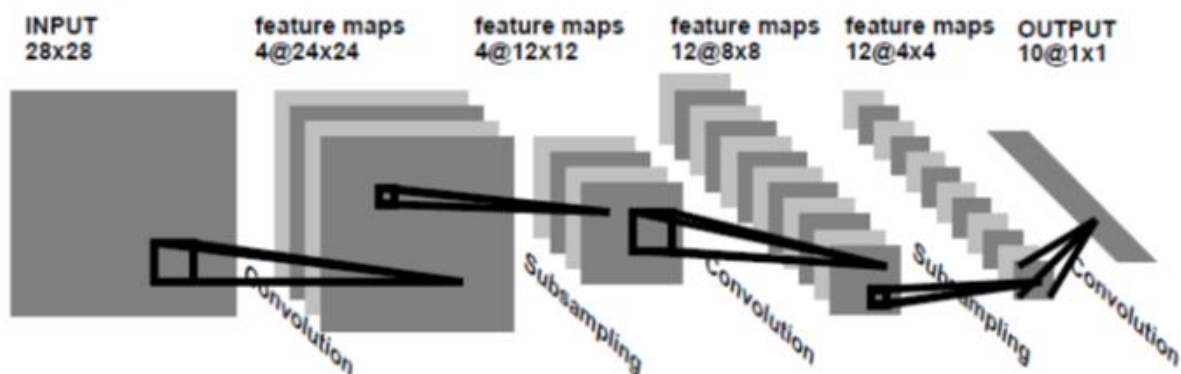


Рисунок 4 - LeNet первого поколения

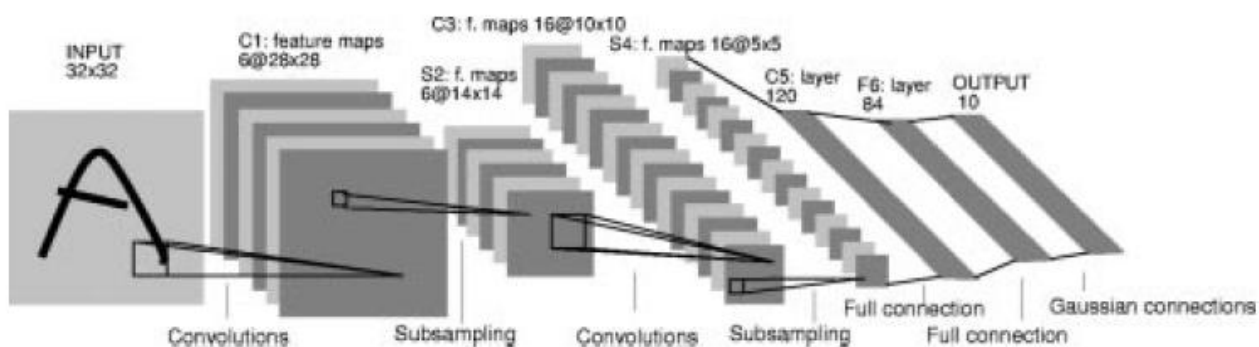


Рисунок 5 - LeNet пятого поколения

В качестве нейронной сети стоит взять сверточную нейронную сеть. Сверточные сети являются современными и эффективными инструментами в области разработок ИИ. За свою историю они ни раз улучшались и дорабатывались.

### 1.7 Сравнение библиотек глубоко обучения

OpenCV – это библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. Разрабатывается библиотека с 2006 год и по сей день (последняя версия на момент написания бакалаврской работы это 4.1.0 датированная 9.04.19). Преимущество библиотеки заключается в том, что ее можно использовать в свободной форме в академических или коммерческих целях. Библиотека

разрабатывается на разных языках программирования, что в купе с долгой поддержкой от разработчиков библиотеки собрало большое комьюнити людей, занимающихся задачами по распознаванию лиц.

EmguCV является кроссплатформенной оберткой над библиотекой OpenCV. EmguCV была полностью написана на языке `c#` с использованием преимуществ платформы .NET.

Вторая библиотека Face API разрабатывается компанией гигантом Microsoft. Face API в отличии от OpenCV является облачной библиотекой для использования которой нужно иметь аккаунт в системе Azure. Библиотека распространяется по подписке имеет довольно подробно и просто написанную документацию. Что снижает порог вхождения для разработчиков систем по распознаванию лиц. Библиотека имеет возможность распознавать не только лицо, но и эмоции человека.

Третья библиотека, которую хотелось бы отметить называется Amazon Recognition, как и предыдущая библиотека она является облачной и распространяется по подписке требующая аккаунт от Amazon.

### **1.8 Алгоритм эмоциональной системы кодирования лицевых движений для распознания эмоций человека**

Для распознания используется так называемая эмоциональная система кодирования лицевых движений или ЭМСКЛиД (англ. EMFACS - Emotion Facial Action Coding System). Разработали эту систему два психологами по имени П. Экман и У. Фризен в 1978 году. Система ЭМСКЛиД является разновидность системы СКЛиД (англ. FACS - Facial Action Coding System) и рассматривает лишь движения лиц, связанные с эмоциями. Для измерений в системе СКЛиД используются так называемые двигательные единицы и двигательные дескрипторы.

Двигательные единицы (ДЕ) это основные движения, совершаемые отдельными мышцами или группой мышц.

Двигательные дескрипторы (ДД) это движения, совершаемые группами мышц (например, выдвижение нижней челюсти вперед). Мышечная основа для этих движений не указана.

Двигательные единицы могут быть разной интенсивности и измеряются по пяти бальной шкале с обозначениями латинских букв от А до Е, где А – это слабо различимое движение, В – это незначительное, С – заметное или ярко выраженное, D –это сильное или крайне заметное, а Е – это предельное.

Цифрами представлены коды двигательных единиц и двигательных дескрипторов таблица которых представлена в приложении.

Таблица 1 - Коды ЭмСКЛид семи основных эмоций

Эмоция	Прототип	Главные варианты
Удивление	1+2+5В+26 1+2+5В+27	1+2+5В 1+2+26 1+2+27 5В+26 5В+27
Страх	1+2+4+5*+20*+25, 26, или 27 1+2+4+5*+25, 26, или 27	1+2+4+5*+L или R20*+25, 26, или 27 1+2+4+5* 1+2+5*, с/без 25, 26, 27 5*+20* с/без 25, 26, 27
Радость	6+12* 12С/D	
Печаль	1+4+11+15В с/без 54+64 1+4+15* с/без 54+64 6+15* с/без 54+64	1+4+11 с/без 54+64 1+4+15В с/без 54+64 1+4+15В+17 с/без 54+64 11+15В с/без 54+64 11+17
Отвращение	9 9+16+15, 26 9+17 10* 10*+16+25, 26 10+17	
25 или 26 могут встречаться со всеми прототипами и основными вариантами		
Презрение	9 или U10 12 U14 или B14	L12+L14 R12+R14

Эмоция	Прототип	Главные варианты
Гнев	4+5*+7+10*+22+23+25, 26 4+5*+7+10*+23+25, 26 4+5*+7+23+25, 26 4+5*+7+17+23 4+5*+7+17+24 4+5*+7+23 4+5*+7+24	Любые из прототипов без любой из следующих ДЕ: 4, 5, 7 или 10.

Звездочками в данной таблице означают что двигательные единицы для данного типа движения могут быть любой интенсивности.

В качестве искусственной нейронной сети была выбрана сверточная нейронная сеть за свои качества распознавания сложных объектов. Также сверточная сеть имеет хорошую производительность. За свои качества на данный момент она является самой распространённой нейронной сетью. Это несомненно должно повысить точность распознавания лица человека.

На основании рассмотренного теоретического материала Распознавания образов было установлено следующее.

Исходя из задачи приложения по распознаванию образов, лучшим вариантом в качестве нейронной сети будет использовать нейронную сеть сверточного типа. За свои качества в распознавании сложных объектов таких как лицо. В качестве активационной функции подойдет алгоритм ReLU за свою простоту без потери качеств.

## 2 АНАЛИЗ ИНСТРУМЕНТОВ ДЛЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

На выпускную квалификационную работу требуется разработать приложение по распознаванию биометрических данных человека. Биометрические данные разделяются на два основных класса:

1. Физиологические - к физиологическим биометрическим данным относят черты, относящиеся к телу человека. Такие как: отпечатки пальцев, распознавание лица, ДНК, ладонь руки, сетчатка глаза, запах, голос.
2. Поведенческие - к поведенческим биометрическим данным относят все что касается поведения. К примеру походка, речь.

В данной системе распознавания биометрических признаков человека было решено создать систему по распознаванию лиц людей.

Библиотека EmguCV выбранная и рассматриваемая ранее позволяет создавать свёрточные нейронные сети, обнаруживать и распознавать объекты на изображениях, проводить распознавание объектов, а также обучать нейронную сеть. EmguCV способна работать с камерой как источником изображений.

### 2.1 Архитектура библиотеки EmguCV

Библиотек Emgu CV имеет два уровня обертки:

1. Основной уровень (*слой 1*) содержит методы, структуры и перечисления, которые отражают что EmguCV это обертка над OpenCV;
2. Второй уровень (*слой 2*) содержат классы, которые отражают преимущества .NET платформы.

На рисунке 6 продемонстрирована архитектура библиотеки EmguCV.

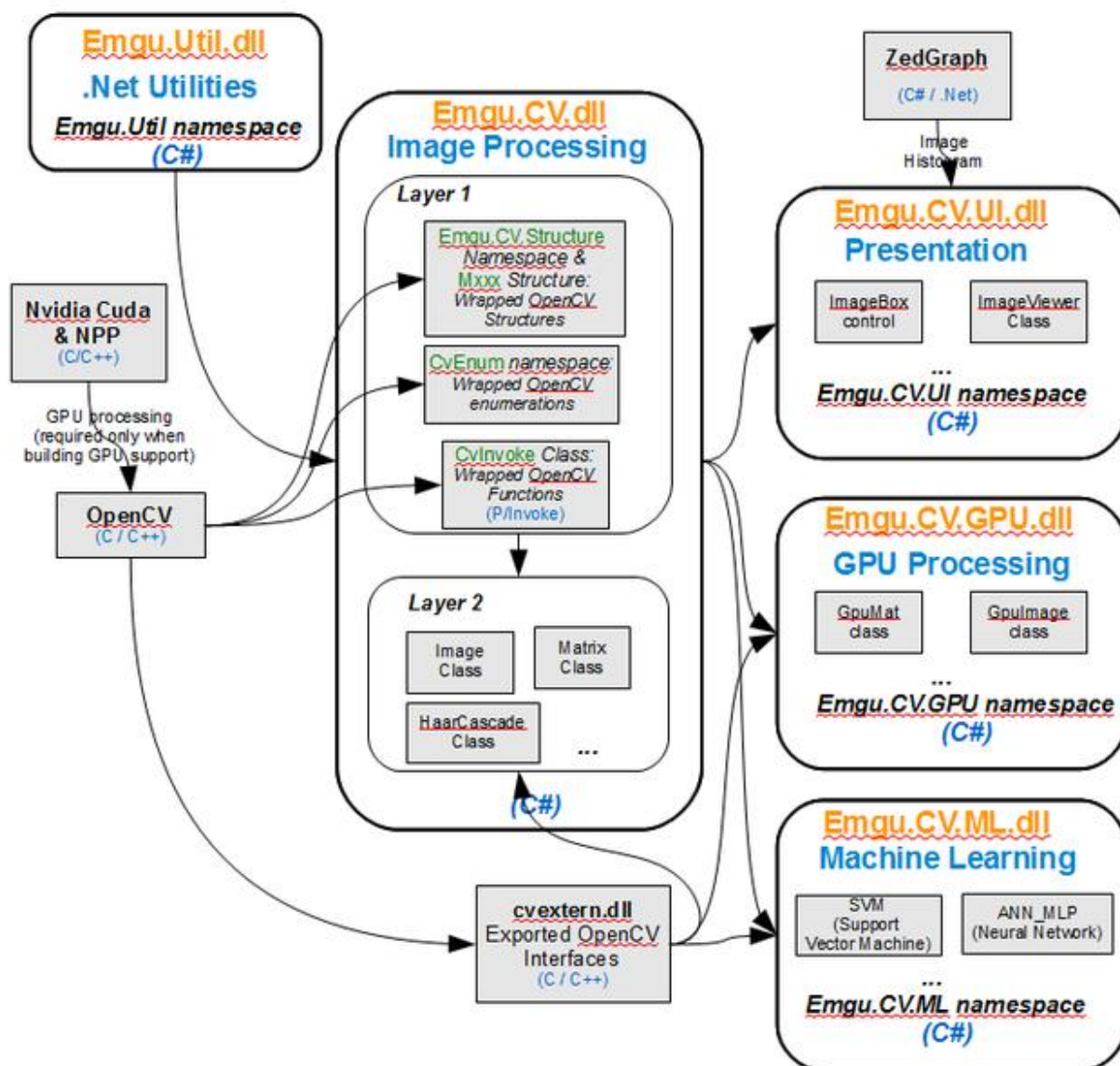


Рисунок 6 - Архитектура EmguCV

## 2.2 Архитектура платформы .NET

Технология .net это разработка компании Microsoft и была выпущена в феврале 2002 года. И разрабатывалась лишь в рамках .net Framework системы. Платформа .net является запатентованной технологией и рассчитана для разработок приложений под операционную систему Windows. Но существовали проекты от независимых разработчиков такие как Mono и Portable.NET. С 2014 года компания Microsoft начало выпускать .net Core ориентированная на кроссплатформенную разработку и эксплуатацию. Не давно компания объявила, что вышедший .net Framework 4.8 в 2019 году станет последней и начиная с 2020 года будет поддерживаться только .net Core.

Одной из основных задач .net состоит в совместимости программных частей, написанных на разных языках. Популярные языки, которые относятся к семейству .net: C#, Visual Basic .NET; JScript .NET; C++/CLI; F#, J#.

Код написанный на этих языках компилируется в так называемый Common Intermediate Language (CIL) его еще называют IL код. Этот код исполняется либо виртуальной машиной Common Language Runtime (CLR) или при помощи утилиты NGen.exe переводится в исполняемый код под конкретный целевой процессор. При использовании виртуальной машины во время исполнения программы JIT-компилятор, встроенный в ней на лету преобразует байтовый код в машинный под конкретный процессор. По мимо этого CLR заботится о базовой безопасности и управлением памяти. На рисунке 7 изображена архитектура .net платформы.

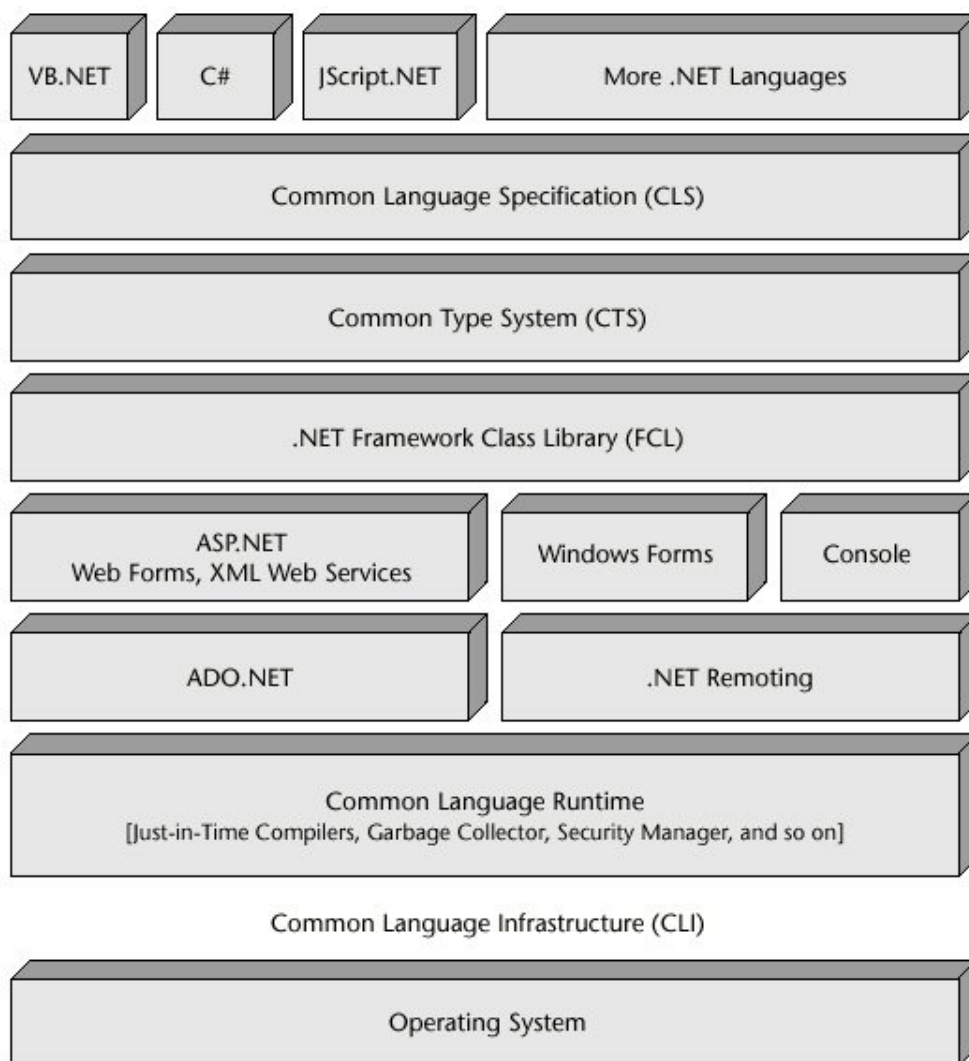


Рисунок 7 - Архитектура .NET



## 2.3 Архитектура свёрточной нейронной сети

Для нахождения объектов нейронной сети достаточно иметь входной слой, на которое подается исходное изображение, следующим должен идти свёрточный слой с функцией активации за тем подвыборочный слой эти действия повторяются до тех пор, пока входная информация из двумерного представления не превратится в одномерное и после этого информация уходит на выходной слой. Каждый из слоев можно представить в виде плоскости и состоящий из нейронов. Каждый нейрон получает сигнал от нейронов предыдущего слоя. Таким образом результат работы предыдущего слоя пропускается через синоптические коэффициенты, а результат отражается на текущем нейроне. Каждая плоскость способна отыскать нужные ей участки изображения в предыдущем поле. Размер локального поля задается самостоятельно в процессе разработки нейронной сети.

В свёрточной нейронной сети слои делятся на два типа свёрточные и подвыборочные. В свёрточных слоях при сканировании рецептивные поля наслаиваются друг на друга. В подвыборочных слоях уменьшается масштаб плоскостей путем усреднения значений выходных нейронов. Последующие слои извлекают все более и более общие характеристики, на которые все меньше и меньше влияют шумы изображения.

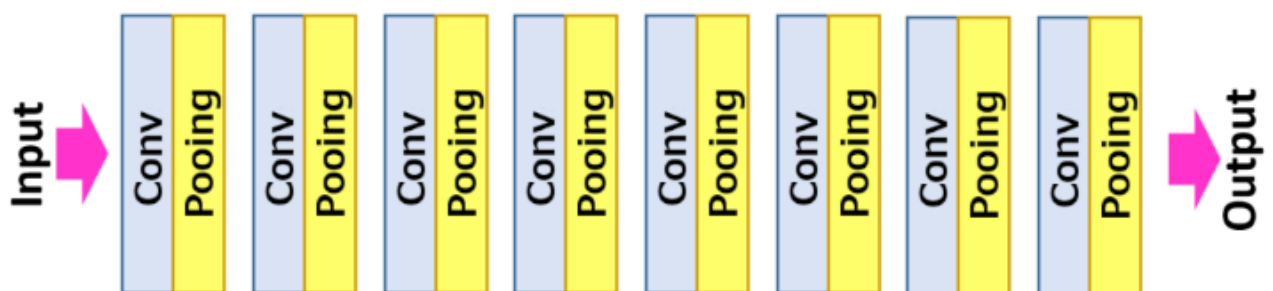


Рисунок 8 - Архитектура свёрточной нейронной сети

Входной слой имеет размерность  $[100 \times 100 \times 1]$  (картинка черно-белая). Следующим идет слой свертка где значение каждого пикселя умножается на соответствующий ему значение в фильтре имеющий размер  $[5 \times 5 \times 1]$ . Размер фильтра имеет размер 5 по причине повышения точности распознавания, но

нужно быть осторожным так как большая размерность фильтра повышает требования к вычислительной мощности. Значения фильтра изначально устанавливаются на значение 0 и в ходе обучения могут изменяться от 0 до 1.

Размер сверточного слоя вычисляется по формуле:

$$sW, sH = W - sW + 1, H - sH + 1, \quad (9)$$

где:

$W$  – ширина входного изображения;

$sW$  – ширина фильтра;

$H$  – высота входного изображения;

$sH$  = высота фильтра.

Из этого следует что размеры сверточного слоя на первом уровне должны быть [ 96 x 96 x 1], как правило поле свертки содержит в себе активационную функцию. В данной ситуации в роле активационной функции выступает ReLU  $f(x) = \max(0, x)$ . Формула значений в поле свертки вычисляется следующим образом:

$$x_{ij}^l = \sum_{a=0}^{sW} \sum_{b=0}^{sH} w_{a*b}^l * \max(0, x_{(i*s-a)(j*s-b)}^l) + b^l, \quad (10)$$

где

$i, j, a, b$  – это индексы элементов в матрицах

$s$  – величина шага;

$l, l-1$  – индексы слоев сети;

$w^l$  – ядро свертки;

$b^l$  – смещение;

$x^l$  - скалярный результат

Следующий слой за слоем свертки идет пулинг. На этом этапе слой становится в два раза меньше и имеет значения [48 x 48 x 1], так как фильтр, который советуется имеет размеры [2 x 2]. На этом слое мы просматриваем область [2 x 2] с шагом 2 и находим максимальное число.

Дальше используется опять сверточный слой и такой алгоритм повторяется до тех пор, пока массив не превратится в [ sW x sH x 1] (5 x 5 x 1). На этом этапе формируется выходной массив размером sW \* sH \* 1. В итоге

выходной массив должен будет равняться 25. Слоев в программе должно быть 8.

[INPUT - CONV - POOL – CONV - POOL - CONV – POOL - FC] – схема нейронной сети при входных данных 100 x 100

INPUT – входной слой

CONV – сверточный слой с функцией активации

POOL – пулинг слой

FC – результат работы нейронной сети

Из 10000 элементов на входе нейронная сеть выдает 25 элементов это очень мало. Для увеличения этого количества стоит разбивать на несколько уровней сразу после входного слоя. Это поможет повысить точность распознавания готовой системы за счет большего числа выходных элементов. Зависимость чисел на выходе прямо пропорционально количеству уровней. В сети будут располагаться четыре уровня что даст на выходе 100 элементов.

На основании проведенных вычислений по анализу инструментов для разработки программного обеспечения можно сделать следующие выводы.

Сверточная нейронная сеть должна обладать восьмью слоями и четырьмя уровнями. При размерности входного потока [100 x 100] и черно-белого типа данных.

Библиотека EmguCV обладает полным функционалом библиотеки OpenCV, но в отличие от OpenCV обладает преимуществами, которые дает ей платформа .net.

Платформа .net позволяет разработчику не беспокоиться за использованием памяти и переноса своего кода на другую машину. Также платформа дает возможности использовать сильные стороны языков, входящих в ее состав.

## 3 ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ПРОГРАММЫ

### 3.1 Разработка нейронной сети программными средствами

Для реализации программы по распознаванию биометрических признаков человека использовался объектно-ориентированный язык высокого уровня C#. Язык C# обладает кроссплатформенными качествами при работе на .NET Core. C# относится к языкам семейства Си подобных. Язык имеет строгую статистическую типизацию. C# не поддерживает множественное наследование в отличие от предшественника в лице C++. Библиотека EmguCV написана на C#, что позволяет вызывать её непосредственно без каких-либо дополнительных затрат. Основной средой разработки является Visual Studio 2019.

Первым делом нужно создать проект выбрав WPF App (.NET Framework). Следующим шагом нужно добавить в проект из области управления пакетами NuGet библиотеку EmguCV. Для того что бы вызвать Nuget нужно в Solution Explorer найти строчку References щелкнуть по ней ПКМ и выбрать Manage NuGet Packages.

На рисунке 9 изображен результат поиска библиотеки EmguCV.

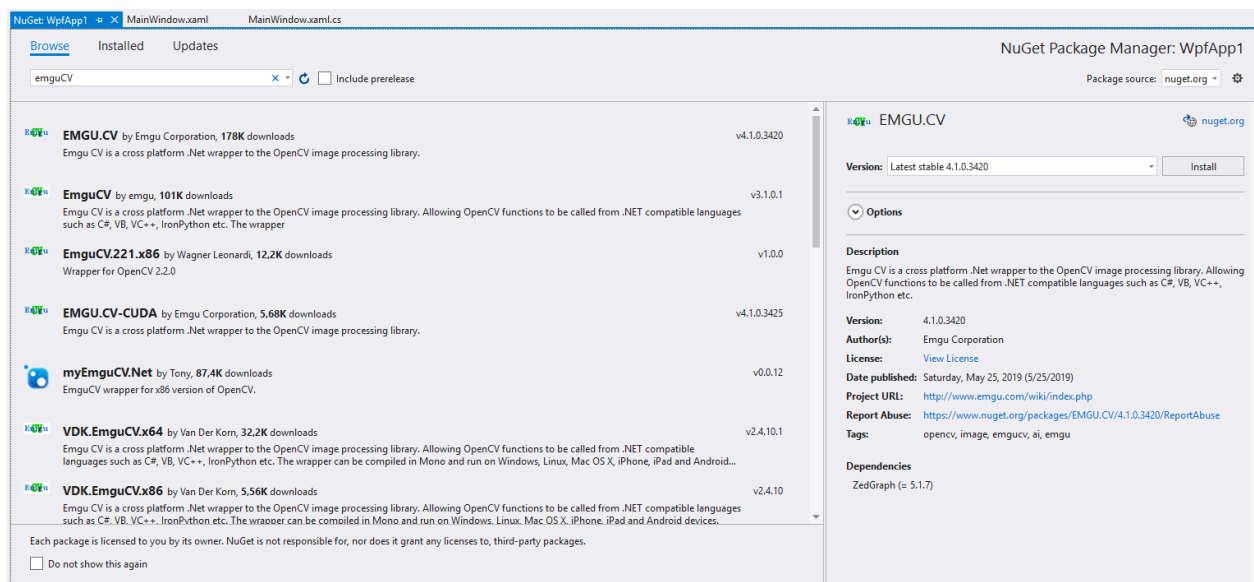


Рисунок 9 - NuGet Packages и найденной библиотекой EmguCV

На изображение 10 показана код реализации алгоритма свёрточной сети на языке высокого уровня с#.

```
private void GenerateNSCov()
{
    int NeuronCountInHiddenLayer = 8;
    HiddenLayer = new List<Neuron>();
    for (int i = 0; i < NeuronCountInHiddenLayer; i++)
    {
        List<int> Root = new List<int>();
        for (int j = 0; j < 5; j++)
        {
            Root.Add(j + i);
        }
        List<int> Leaf = new List<int>();
        Leaf.Add(0);
        Leaf.Add(1);
        double AP = int.Parse(textBox4.Text);
        HiddenLayer.Add(new Neuron(Root, Leaf, 1, 0, AP, P));
        HiddenLayer.Add(new Neuron(Root, Leaf, 1, 0, AP, P));
    }
    for (int i = 0; i < Svertka1.Length; i++)
    {
        Svertka1[i] = P.NextDouble() / 10.0;
        Svertka2[i] = P.NextDouble() / 10.0;
    }
    List<int> Root2 = new List<int>();
    for (int j = 0; j < 2 * NeuronCountInHiddenLayer; j++)
    {
        Root2.Add(j);
    }
    List<int> Leaf2 = new List<int>();
    Leaf2.Add(0);
    GoodOut = new Neuron(Root2, Leaf2, 0, 0,
        int.Parse(textBox4.Text), P);
    BadOut = new Neuron(Root2, Leaf2, 0, 0,
        int.Parse(textBox4.Text), P);
}
```

Рисунок 10 - Реализация свёрточной сети

### 3.2 Обучение и проверка обученности нейронной сети

Тестирование результатов обучения происходило на основе американского сериала Друзья. Видео было скачено из сети интернет и при помощи библиотеки EmguCV были вытащены кадры с шагом в одну секунду. Вручную были отсортированы все изображения, по папкам, которые были созданы для каждого из шести главных героев сериала. Также мною были

скачены по десять фотографий актеров для увеличения выборки. Файлы содержали в себе имя персонажа и порядковый номер файла. В среднем на каждого героя пришлось около 500 кадров и того 3000 кадров. Было также решено при помощи фильтров наложить некоторые шумы на изображения. Таким образом база была увеличена еще в три раза и составляла уже по 1500 кадров на персонажа с общим объемом в 9000 кадров.

Изображения были отданы на обработку нейронной сети, таким образом сеть должна была запомнить главных героев сериала.

Для проверки обученности нейронной сети в корне где лежат папки с фотографиями была добавлена еще одна папка куда должны складываться все кадры где не удалось распознать человека.

Для теста была скачена одна серия сериала Друзья из интернета. После обрабатывания программы было получено 32784 кадра из них 21408 кадр нейронная сеть распределила. Осталось 11376 кадров, помещённых в специально отведенную папку это примерно 24,7% от общего числа кадров. При просмотре этих кадров было выяснено что в основном там находятся кадры без главных героев или кадры где было смазано лицо, что даже человек не смог бы определить кто находится в кадре. Но были кадры где лица героев не были достаточно хорошо освещены таких кадров набралось около 3590 что составило чуть более 10,95% от общего числа кадров, распознанных программой. Из этого следует что точность распознавание программы составило 89%.

### **3.3 Апробация работы реализованной нейронной сети**

Для теста программы была выбрана актриса Courteney Cox и скачена фотография из интернета. На изображении 11 приведен пример того как выглядит GUI программы.

Кнопка Start Detection and Recognise создает окно, в которое транслируется видео поток в реальном времени с web-камеры. Зеленым квадратом программа отметила лицо которая нашла при сканировании

изображений с камеры устройства. Также есть кнопка Save Face которая сохранит в документ с расширением .txt JSON строку. Красным подсвечивается имя человека, под которым он сохранен в системе.

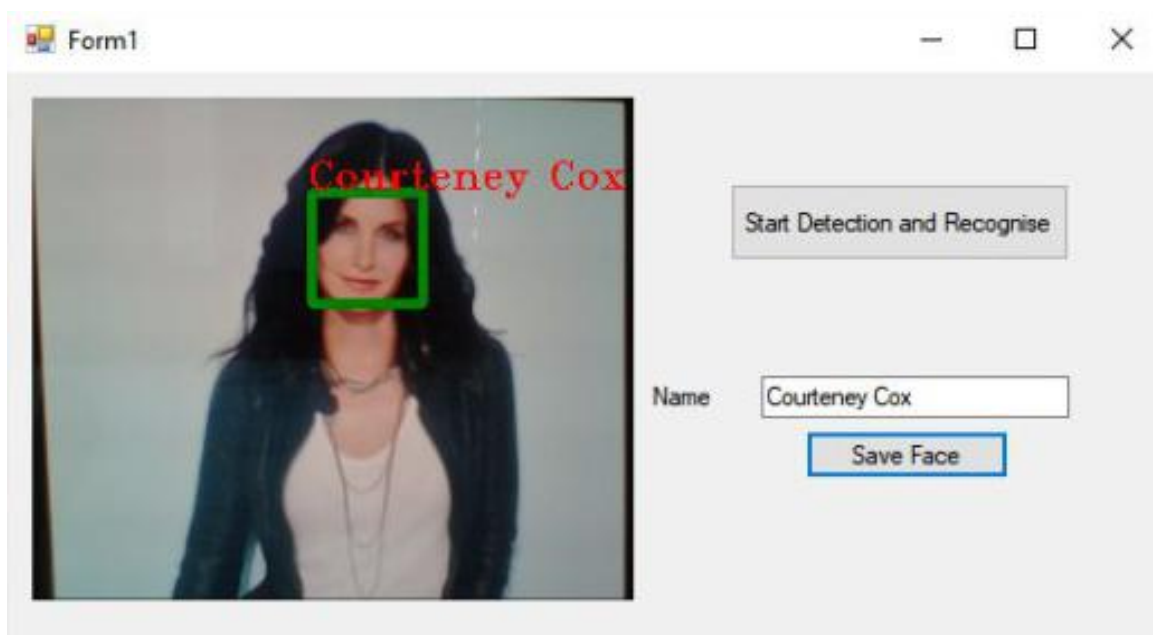


Рисунок 11 - Работа программы

Результаты работы программы предоставлены в форме JSON файла для возможности удобного хранения или при надобности отправить результатов работы во внешние службы программного обеспечения.

Результаты распознавания представлены на рисунке 12.

```
[{"faceId": "Courteney Cox", "faceRectangle": { "top": 181, "left": 434, "width": 177, "height": 177 }, "faceAttributes": { "hairColor": [ { "color": "black", "confidence": 0.99 }, { "color": "brown", "confidence": 0.93 }, { "color": "gray", "confidence": 0.42 }, { "color": "other", "confidence": 0.27 }, { "color": "blond", "confidence": 0.04 }, { "color": "red", "confidence": 0.04 } ], "smile": 0.957, "gender": "female", "age": 36.0, "emotion": { "anger": 0.0, "contempt": 0.0, "disgust": 0.0, "fear": 0.0, "happiness": 0.957, "neutral": 0.042, "sadness": 0.0, "surprise": 0.0 },}}]
```

Рисунок 12 - JSON строка, отформатированная вручную

Поле faceId - это единственный параметр, который можно заполнить вручную, когда при сохранении нового лица в поле Name вести имя человека. Все остальные значения программа заполняет на основе входного изображения.

В поле faceRectangle содержится информация в какой области расположено лицо человека. HairColor содержит в себе цвет и вероятность.

Emotion содержит в себе шесть эмоций с их вероятностью. Все числа в этом JSON строке имеют диапазон значений от 0 до 1.

При разборе Json строки видно имя человека которого распознала система. По фотографии система не смогла точно определить цвет волос и предполагает, что они либо черного цвета с вероятностью 99% или коричневые с вероятностью 93%. Также система определила, что человек улыбается, является девушкой и предположила, что человеку 36 лет на момент съемки. Также система предположила, что человек испытывает радость с шансом в 98,7%.

Была совершена попытка улучшить точность за счет наложения фильтра Blur. Изображение 13 демонстрирует применение фильтра в коде.

```
if (MaskBlurFactor > 0)
    Cv2.GaussianBlur(alphaMask, alphaMask, new Size(MaskBlurFactor, MaskBlurFactor), MaskBlurFactor);
```

Рисунок 13 - Применение фильтра Blur

Применение фильтра помогло повысить точность до 89,97% ~ 90%. От применения фильтра размытия ожидал улучшение качества на 2 - 4 %, но результат показал выигрыш лишь в 1%.

В будущем стоит добавить окна куда будет выводиться зашумленность сигнала для понимания специфики проблемы при не возможности определения лица человека. Также стоит пересмотреть архитектуру нейронной сети для повышения ее эффективности.

На основании проведенного анализа готовой системы по распознаванию образа человека выяснилось, что программа со своей основной задачей по распознаванию образов справляется хорошо при хорошей освещенности. Удалось достичь точности в распознании в 90% и распознавания образов в режиме реального времени используя Web-камеру.



## ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы была выполнена задача по реализации программного обеспечения распознавания образов человека. Было произведено обучение и распознаны лица человека. В рамках бакалаврской работы так же было продемонстрировано распознавание человека с web-камеры устройства.

Система была протестирована и обнаружено что при хорошем освещении программе не составляет особо большого труда определить лицо человека, но в случае недостатка света программа теряет лицо. Также было выяснено что лицо обнаруживается если система видит два глаза нос и губы если человек повернут в профиль система не может определить на изображении такое лицо.

Была написана программа в основе которой лежит сверточная нейронная сеть, работающая с черно- белыми изображениями. Программа также имела методы по переводу изображения из цветного в черно-белое. Такой подход позволил обрабатывать в три раза меньше данных, так как в случае цветного изображения нужно было бы работать с каждым цветом отдельно. Сеть была успешно пред обучена на выборке изображений количество которых превышало 9000 экземпляров. Для подбора нужных значений весов. Результатом обучения стала точность распознавания в 90%. При изучении вопроса точности сверточных нейронных сетях мною были обнаружены цифры от 90 до 95% точности. Точность программы в 90% считаю достаточно приемлемым результатом. Также для повышения точности можно создавать 3D модель векторного представления лица и поворачивать угол наклона фотографии с целью выравнивание угла наклона головы если таков присутствует.

Время обработки одного изображения в среднем составляет 1,2-1,3 секунды. При поиске информации по сверточным нейронным сетям. Была обнаружена информация об обработке одного изображения раз в секунду. Для повышения этого результата можно опробовать ввести слой дропаут в нейронную сеть.

Для повышения качества распознавания человека можно дополнительно реализовать систему по распознаванию голоса. По причине того, что звук также, как и изображение можно собирать дистанционно. Основой системы по распознаванию голоса также может служить сверточная нейронная сеть. Такую систему предлагается реализовать на основе проверки амплитуды звука.

В системе которой присутствуют две разные проверки по идентификации должны помочь повысить точность с превышением значения в 90%.

В ходе работы над бакалаврской работой были обнаружены проблемы с точностью распознавания объектов и предложены способы их решения.

Поставленные задачи в рамках выпускной квалификационной работы считаю выполненными успешно.

## СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Ворона В.А., Костенко В.О. Биометрические технологии идентификации в системах контроля и управления доступом // Computational nanotechnology. 2016. №3. URL: <https://cyberleninka.ru/article/n/biometricheskie-tehnologii-identifikatsii-v-sistemah-kontrolya-i-upravleniya-dostupom> (дата обращения: 20.02.2019).
2. Габриелян Б. А. Нейросетевой метод выделения и распознавания лиц // Известия ЮФУ. Технические науки. 2006. №9-1. URL: <https://cyberleninka.ru/article/n/neyrosetevoy-metod-vydeleniya-i-raspoznavaniya-lits> (дата обращения: 20.02.2019).
3. Dixon J. Mastering .NET Machine Learning. USA: Packt Publishing, 2016. ISBN 978-1785888403.
4. Bradski G, Kaehler A. Learning OpenCV: Computer Vision with the OpenCV Library. Издательство: O'Reilly Media, 2008 г. ISBN: 0596516134
5. C# Guide | Microsoft Docs [Электронный ресурс]. Режим доступа: <https://docs.microsoft.com/en-us/dotnet/csharp/>
6. Хайкин С. Нейронные сети: полный курс. 2 издание. Издательство: Вильямс, 2016 г. ISBN: 978-5-8459-2069-0
7. Jeffrey Richter CLR via C#: Developer Reference, US.: Published by Microsoft Press, 2012.- 896с. - ISBN-13: 978-0735667457
8. Allen Jones, Adam Freeman Технология программирования и программная инженерия: Developer Reference, US.: Published by Microsoft Press, 2002.- 576с. - ISBN-13: 978-0735617797
9. From Zero to Hero in Json with C# [Электронный ресурс]. Режим доступа: <https://www.c-sharpcorner.com/article/from-zero-to-hero-in-json-with-c-shar/>
10. OpenCV: OpenCV Tutorials [Электронный ресурс]. Режим доступа: [https://docs.opencv.org/master/d9/df8/tutorial\\_root.html](https://docs.opencv.org/master/d9/df8/tutorial_root.html)
11. What is the Azure Face API? [Электронный ресурс]. Режим доступа: <https://docs.microsoft.com/id-id/azure/cognitive-services/face/overview>

12. Екатерина Лаврищева Технология программирования и программная инженерия: учебник для вузов, М.: Издательство Юрайт, 2017.- 432с. - ISBN 978-5-9916-8275-6
13. Биометрия [Электронный ресурс]. Режим доступа: <https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%BE%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%8F>
14. Система распознавания лиц Face-Интеллект [Электронный ресурс]. Режим доступа: <https://www.itv.ru/products/intellect/faceintellect/>
15. Facial Recognition System [Электронный ресурс]. Режим доступа: [https://en.wikipedia.org/wiki/Facial\\_recognition\\_system](https://en.wikipedia.org/wiki/Facial_recognition_system)
16. Amazon Rekognition [Электронный ресурс]. Режим доступа: <https://aws.amazon.com/rekognition/>
17. Face-Recognition Using OpenCV: A step-by-step guide to build a facial recognition system [Электронный ресурс]. Режим доступа: <https://hackernoon.com/face-recognition-using-opencv-a-step-by-step-guide-to-build-a-facial-recognition-system-8da97cd89847>
18. Face Detection and Recognition in C# using EmguCV 3.0 (OpenCV Wrapper) – Part 1 « Ahmed's Blog [Электронный ресурс]. Режим доступа: <https://ahmedopeyemi.com/main/face-detection-and-recognition-in-c-using-emgucv-3-0-opencv-wrapper-part-1/>
19. Paul Kimmel LINQ Unleashed: for C# (Volume 1): Developer Reference, US.: Published by Sams Publishing, 2008.- 552с. - ISBN-13: 978-0672329838
20. Jon Skeet C# in Depth, 3rd Edition Third Edition: Developer Reference, US.: Published by Manning Publications, 2013.- 616с. - ISBN-13: 978-1617291340
21. Биометрия [Электронный ресурс]. Режим доступа: <https://en.wikipedia.org/wiki/Biometrics>

## ПРИЛОЖЕНИЕ А

**Таблица кодов двигательных единиц и двигательных дескрипторов**

№ ДЕ	Оригинал	Перевод	Мышечная основа
0	Neutral face	Нейтральное лицо	
1	Inner Brow Raiser	Подниматель внутренней части брови	лобная мышца (медиальная часть)
2	Outer Brow Raiser	Подниматель внешней части брови	лобная мышца (латеральная часть)
4	Brow Lowerer	Опускатель брови	мышца гордецов; мышца, опускающая бровь; мышца, сморщивающая бровь
5	Upper Lid Raiser	Подниматель верхнего века	мышца, поднимающее верхнее веко
6	Cheek Raiser	Подниматель щеки	круговая мышца глаза (глазничная часть)
7	Lid Tightener	Натягиватель века	круговая мышца глаза (вековая часть)
8	Lips Toward Each Other	Губы навстречу друг другу	круговая мышца рта
9	Nose Wrinkler	Сморщиватель носа	мышца, поднимающая верхнюю губу и крыло носа
10	Upper Lip Raiser	Подниматель верхней губы	мышца, поднимающая верхнюю губу (также известна как квадратная мышца верхней губы), нижнеглазничная головка
11	Nasolabial Deepener	Углубитель носогубной складки	малая скуловая мышца
12	Lip Corner Puller	Подниматель уголка губы	большая скуловая мышца
13	Sharp Lip Puller	Острый подниматель уголка губы	мышца, поднимающая угол рта (также известна как собачья мышца)
14	Dimpler	Ямочка	щёчная мышца (также известна как

			мышца трубачей)
15	Lip Corner Depressor	Опускатель уголка губы	мышца, опускающая угол рта (также известна как треугольная мышца рта)
16	Lower Lip Depressor	Опускатель нижней губы	мышца, опускающая нижнюю губу (также известна как четырёхугольная мышца нижней губы)
17	Chin Raiser	Подниматель подбородка	подбородочная мышца
18	Lip Pucker	Сморщиватель губ	резцовая мышца верхней губы и резцовая мышца нижней губы
19	Tongue Show	Показ языка	
20	Lip Stretcher	Растягиватель губ	мышца смеха с/без подкожной мышцей шеи
21	Neck Tightener	Натягиватель шеи	подкожная мышца шеи
22	Lip Funneler	Губы воронкой	круговая мышца рта
23	Lip Tightener	Натягиватель губ	круговая мышца рта
24	Lip Pressor	Сжиматель губ	круговая мышца рта
25	Lips Part	Губы разведены	мышца, опускающая нижнюю губу или расслабление подбородочной мышцы или круговой мышцы рта
26	Jaw Drop	Челюсть опущена	жевательная мышца, расслабленные височная мышца и медиальная крыловидная мышца
27	Mouth Stretch	Рот широко открыт	медиальная крыловидная мышца и латеральная крыловидная мышца, двубрюшная мышца
28	Lip Suck	Втягивание губ	круговая мышца рта
29	Jaw Thrust	Нижняя челюсть вперёд	
30	Jaw Sideways	Челюсть вбок	
31	Jaw Clencher	Сжиматель челюстей	жевательная мышца
32	Lip Bite	Покусывание губы	

33	Cheek Blow	Выдувание	
34	Cheek Puff	Раздувание щёк	
35	Cheek Suck	Втягивание щёк	
36	Tongue Bulge	Язык высунут	
37	Lip Wipe	Облизывание губ	
38	Nostril Dilator	Расширитель ноздрей	носовая мышца (внутренняя, или крыльчатая часть)
39	Nostril Compressor	Суживатель ноздрей	носовая мышца (наружная, или поперечная часть) и мышца, опускающая перегородку носа
41	Glabella Lowerer	Опускающий надпереносья	Отдельная часть ДЕ 4: мышца гордецов
42	Inner Eyebrow Lowerer	Опускающий внутреннюю часть брови	Отдельная часть ДЕ 4: мышца, опускающая бровь
43	Eyes Closed	Глаза закрыты	Расслабление мышцы, поднимающей верхнее веко
44	Eyebrow Gatherer	Сведение бровей	Отдельная часть ДЕ 4: мышца, сморщивающая бровь
45	Blink	Моргание	Расслабление мышцы, поднимающей верхнее веко; сокращение круговой мышцы глаза (вековая часть)
46	Wink	Подмигивание	круговая мышца глаза
51	Head Turn Left	Поворот головы влево	
52	Head Turn Right	Поворот головы вправо	
53	Head Up	Голова вверх	
54	Head Down	Голова вниз	
55	Head Tilt Left	Наклон головы влево	
M55	Head Tilt Left	Наклон головы влево	Начало симметричной ДЕ 14 немедленно предваряется или сопровождается наклоном головы влево.

56	Head Tilt Right	Наклон головы вправо	
M56	Head Tilt Right	Наклон головы вправо	Начало симметричной ДЕ 14 немедленно предваряется или сопровождается наклоном головы вправо.
57	Head Forward	Голова вперед	
M57	Head Thrust Forward	Толчок головы вперед	Начало ДЕ 17+24 немедленно предваряется, сопровождается либо завершается толчком головы вперед.
58	Head Back	Голова назад	
M59	Head Shake Up and Down	Кивок головой	Начало ДЕ 17+24 немедленно предваряется, сопровождается либо завершается кивком головы.
M60	Head Shake Side to Side	Голова из стороны в сторону	Начало ДЕ 17+24 немедленно предваряется, сопровождается либо завершается движением головы из стороны в сторону.
M83	Head Upward and to the Side	Голова вверх и в сторону	Начало симметричной ДЕ 14 немедленно предваряется или сопровождается движением головы вверх и в сторону и/или наклоном головы влево или вправо.
61	Eyes Turn Left	Отведение глаз влево	
M61	Eyes Left	Глаза влево	Начало симметричной ДЕ 14 немедленно предваряется или сопровождается движением глаз влево.
62	Eyes Turn Right	Отведение глаз вправо	
M62	Eyes Right	Глаза вправо	Начало симметричной ДЕ 14 немедленно предваряется или сопровождается движением глаз вправо.
63	Eyes Up	Глаза вверх	
64	Eyes Down	Глаза вниз	
65	Walleye	Расходящееся косоглазие	



66	Cross-eye	Сходящееся косоглазие	
M68	Upward Rolling of Eyes	Закатывание глаз	Начало симметричной ДЕ 14 немедленно предваряется или сопровождается закатыванием глаз.
69	Eyes Positioned to Look at Other Person	Глаза на другом человеке	ДЕ 4, 5, 7, по отдельности или в сочетании, происходят в момент, когда взгляд сосредоточен на другом человеке в разговоре.
M69	Head and/or Eyes Look at Other Person	Голова и/или глаза на другом человеке	Начало симметричной ДЕ 14 или ДЕ 4, 5, 7, по отдельности или в сочетании, немедленно предваряется или сопровождается движением глаз или движением головы и глаз для взгляда на другого человека в разговоре.
70	Brows and forehead not visible	Брови и лоб не видны	
71	Eyes not visible	Глаза не видны	
72	Lower face not visible	Нижняя часть лица не видна	
73	Entire face not visible	Всё лицо не видно	
74	Unscorable	Оценивание невозможно	

Эти коды зарезервированы для записи информации о поведении, которая может быть значимой для оцениваемых лицевых движений.

№ ДЕ	Оригинал	Перевод
40	Sniff	Втягивание носом
50	Speech	Речь
80	Swallow	Глотание
81	Chewing	Жевание
82	Shoulder shrug	Пожатие плечом
84	Head shake back and forth	Движение головой назад и вперед
85	Head nod up and down	Кивок головой вверх и вниз
91	Flash	
92	Partial flash	
97*	Shiver/Tremble	Дрожь/Озноб
98*	Fast up-down look	Быстрый взгляд вверх-вниз