

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**ИНТЕГРАЦИЯ МЕТОДОВ КОМПЬЮТЕРНОГО ЗРЕНИЯ В
СОВРЕМЕННЫЕ СИСТЕМЫ И ПРИЛОЖЕНИЯ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 451 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Есина Ильи Дмитриевича

Научный руководитель

зав. каф., к. ф.-м. н., доцент

С. В. Миронов

Заведующий кафедрой

к. ф.-м. н., доцент

С. В. Миронов

Саратов 2023

ВВЕДЕНИЕ

Начиная с 1940 годов в головы ученых крепко и уверенно засело понимание необходимости нейронных сетей. На заре создания люди не могли дать точную оценку тому что в ближайшее будущее полностью изменит ход развития вычислительных систем. В современном мире гиганты IT отрасли делают крупные ставки на развитие нейронных сетей и применение их в реальных бизнес моделях, нанимают лучших экспертов в области машинного обучения и ставят их реализацию в верх приоритета.

Целью моей дипломной работы является изучение возможностей компьютерного зрения, а так же создание на их основе продукта который сможет отразить весь потенциал и показать настоящие возможности машинного обучения.

Проект и тема находятся на острие современных технологий, являются актуальными и могут найти свое применение у людей занимающихся захватом движения, компьютерной графикой с созданием анимаций человеческого тела.

В связи с этим поставлены следующие задачи:

- Анализировать возможности аналогичных продуктов.
- Описать требования которые будут реализованы в будущем продукте.
- Разработать алгоритмы для реализации задачи.
- Разработать UI для взаимодействия с программным обеспечением.
- Разработать отдельное приложение, показывающее использующее уже существующего продукта.

Характеристика материалов исследования. Так как работа с потоковыми видео и обработка их в режиме реального времени дорогостоящая задача, необходимо анализировать системные конфигурации устройств на которых возможен запуск приложения. Работа проекта подразумевает клиент серверное взаимодействие, а значит необходимо анализировать протоколы взаимодействия между устройствами, оценить объемы передаваемых данных, а так же структуру их хранения.

Структура бакалаврской работы. Работа состоит из списка аббревиатур и сокращений, введения, четырех разделов, заключения, списка использованных источников, содержащего 20 наименований и четырех приложений. В первом разделе объясняется цель предстоящей работы, описываются цели и задачи. Приводятся минимальные требования к решению задачи. Во втором

разделе подробно описывается теоретическая часть нейронных сетей. Поднимаются вопросы создания на основе графов, описываются функции активации и метрики качества нейронных сетей. В третьем разделе описывается разработка приложения с использованием API Google MediaPipe, базы данных SQLite и библиотеки для создания графического интерфейса PySide6. В четвертом разделе показывается какими способами можно применять полученную программу в реальных средах разработки. Так же разрабатывается пример демонстрации всех возможностей на платформе Unity.

1 Основное содержание работы

Основой будущего проекта является создание нейронной сети. Попробуем определиться что это такое и зачем это нужно

Для работы с нейронными сетями необходимо полностью отдавать отчет в понимании того с чем предстоит работать. Для этого попытаемся окунуться в математический смысл и попытаемся классифицировать различные виды нейронных сетей.

Нейронная сеть — математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма. Это понятие возникло при изучении процессов, протекающих в мозге, и при попытке смоделировать эти процессы.

Бурное развитие нейронных сетей началось с 1956 года когда американский ученый Розенблатт, Фрэнк изобретает однослойный перцептрон. Использовать данное устройство можно было в элементарных задачах классификации, но основным преимуществом было то, что люди могли понять в случае ускорения прогресса перед ними открывается большое количество задач решения которых не предоставлялось возможным без использования данного или аналогичного ему устройства.

ИНС представляет собой систему соединённых и взаимодействующих между собой простых процессоров (искусственных нейронов). Тем не менее, будучи соединёнными в достаточно большую сеть с управляемым взаимодействием, такие по отдельности простые процессоры вместе способны выполнять довольно сложные задачи. Машинное обучение представляет собой обширную область решения различных задач. Каждая из областей показывает свое видение на нейронные сети:

- С точки зрения машинного обучения, нейронная сеть представляет собой частный случай методов распознавания образов.
- С точки зрения математики, обучение нейронных сетей — это многопараметрическая задача нелинейной оптимизации.
- С точки зрения развития вычислительной техники и программирования, нейронная сеть — способ решения проблемы эффективного параллелизма.

С точки зрения проектирования архитектуры любую нейронную сеть мож-

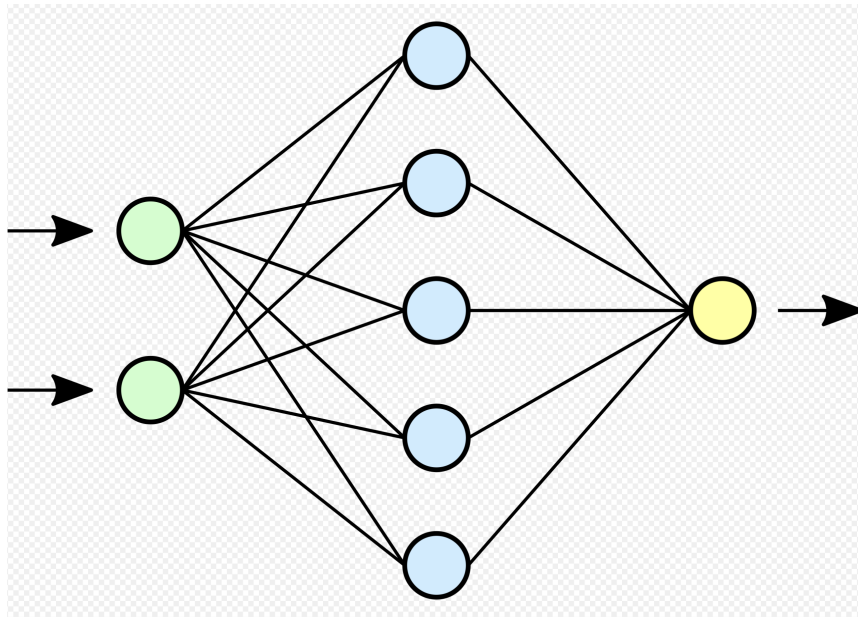


Рисунок 1 – Схема простой нейросети. Зелёным цветом обозначены входные нейроны, голубым — скрытые нейроны, жёлтым — выходной нейрон

но представить в виде графа. Граф — математическая модель представляющая из себя множество вершин и множество пар где каждая пара представляет из себя отрезок от одной вершины к другой. На рисунке (см. Рис ??) представлена нейронная сеть состоящая из 3 слоев.

- Нейроны окрашенные в зеленый цвет называются входными нейронами. Входными данными могут служить пиксели на экране, и любые другие параметры от которых будут зависеть результаты.
- Нейроны окрашенные в желтый цвет являются нейронами выходного слоя и представляют нам значение полученное в ходе обработки нейронной сети.
- Слои расположенные между входным и выходным слоем можно назвать общим слоем внутренние слои. Существует достаточно большое количество различных слоев находящихся между входным и выходным слоем так например в дальнейшем будут применяться слои свертки и развертки для работы с детекцией на фотографиях.

На рисунке (см. Рис ??) можно увидеть пример модели полносвязной нейронной сети. Нейронная сеть называется полносвязной если от каждый нейрон предыдущего слоя связан со всеми нейронами следующего слоя. Пример данной нейронной сети называется глубокой нейронной сетью. Глубокая нейронная сеть — сеть с несколькими слоями между входным и выходным слоями.

Внутри каждого нейрона находится функция активации. Функция акти-

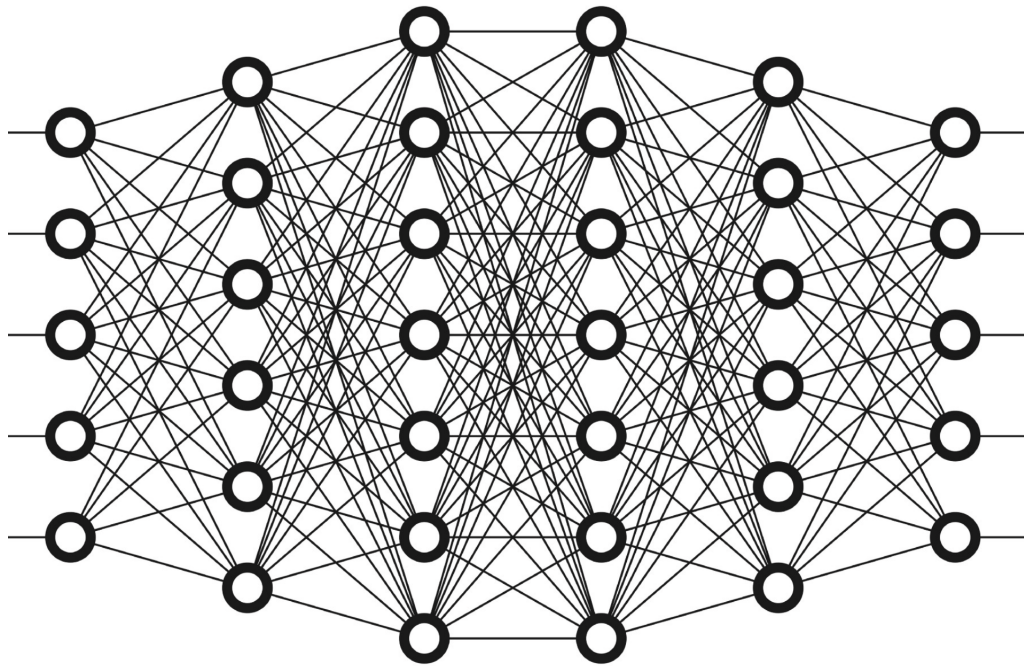


Рисунок 2 – Пример полносвязной нейронной сети.

вазии служит для того что бы определить при каких входных значениях должен активироваться (включаться) нейрон, то есть проводить сигнал. А при каких нет. Нейрон включается, если достигается нужный уровень сигнала, после этого нейрон передает сигнал дальше.

существует 4 основных вида функции активации использующиеся в реальных нейронных сетях и показавшие свою эффективность:

- Ступенька задается уравнением вида: $f(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$
- Сигмоидальная функция задается уравнением вида: $f(x) = \sigma(x) = \frac{1}{1+e^{-x}}$
- Линейно выпрямляющая задается уравнением вида: $f(x) = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases}$
- Функция мягкой сигмойды задается уравнением вида: $f(x) = \sigma(x) = \frac{1}{1+|x|}$

Так как предстоит создать нейронную сеть для обработки фотографии стоит рассмотреть одну из архитектуры нейронной сети и части из которых она состоит. Как правило нейронные сети которые работают с изображениями относятся к глубоким нейронным сетям. На вход для таких сетей должно подаваться изображение где каждому нейрону Соответствует пиксель изображения.

Возникает пара вопросов, а как мы будем работать если все фотографии закодированы в своих кодировках и цветовых стандартах да и к тому же

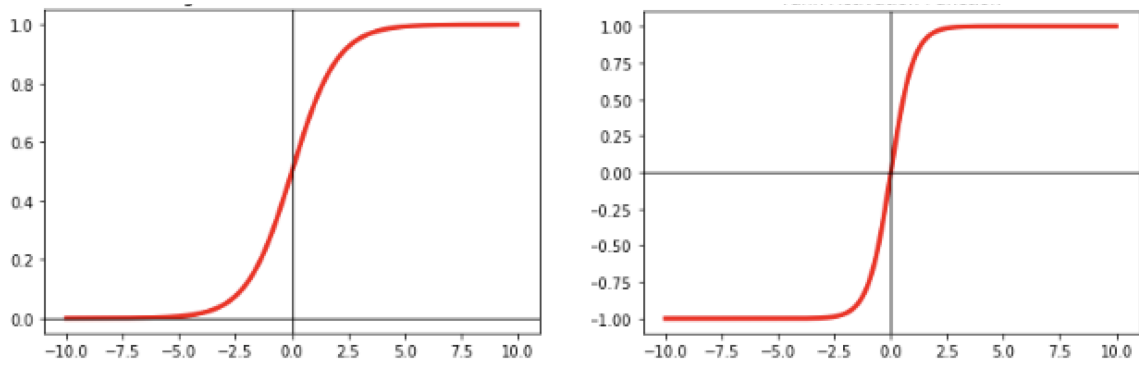


Рисунок 3 – График функций активации Сигмоидальная и функция мягкой сигмойды.

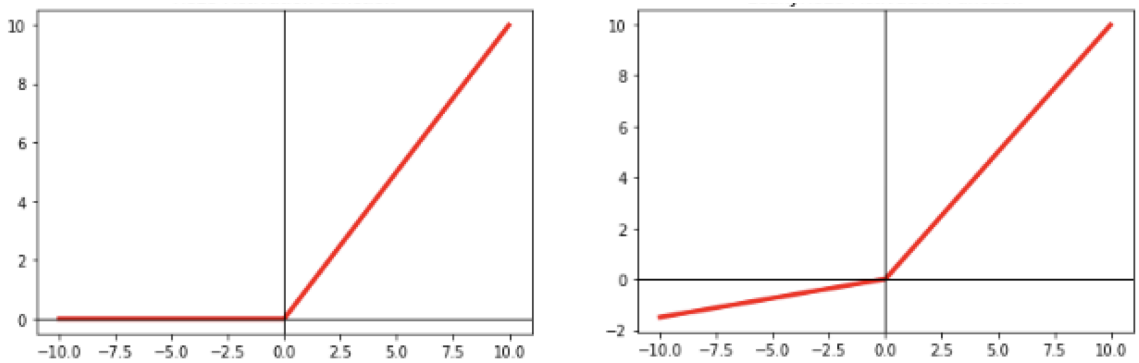


Рисунок 4 – График функций активации Ступенька и Линейно выпрямляющая функция.

в мире не существует единого размера для всех фотографий. Конечно создавать тысячу нейронных сетей где каждая отвечает за свой размер изображения идея не лучшая. Одним из вариантов решения является применения алгоритмов кадрирования изображения для сведения к единому размеру. Очевидно, что чем меньше нейронная сеть тем процесс обучения займет меньшее количество времени, но в таком случае мы можем потерять результат в случае если разрешающая способность изображения резко упадет. Стоит искать баланс между скоростью работы и качеством изображения.

Решая вопрос разных кодировок ученые сошлись на мнении что приведение изображений черно-белому стандарту где значения цвета каждого пикселя нормализованны и находятся в в диапазоне $(0, 1)$ где 0 - цвет черного пикселя, а 1 - белого является оптимальным для работы в нейронных сетях.

После этого необходимо понять, а каким образом можно понять что нейронная сеть или алгоритмы работающие на ней работают адекватно? Для этого существуют критерии качества оценки нейронных сетей.

- Accuracy показывает долю правильных ответов алгоритма.
- Precision называется доля правильных ответов модели в пределах клас-

са — это доля объектов действительно принадлежащих данному классу относительно всех объектов которые система отнесла к этому классу.

— Loss – точность нейронной сети по показателям отклонения, например, среднеквадратического отклонения.

Для реализации необходимо определиться со следующими критериями:

— Какую СУБД будем использовать в ходе создания приложения?

— Какой язык программирования будет использоваться в основе приложения.

— Какой фреймворк будет лежать в основе графического интерфейса?

Для ответа на эти вопросы буду отталкиваться от следующих фактов.

1. Создание нейронной сети будет происходить на языке Python, и для удобства было бы хорошо использовать язык Python в качестве основного языка для написания приложения.

2. Приложение пишется под операционную систему Mac OS, а значит необходимо выбирать из все того что представлено для текущей платформы.

3. Необходимо использовать только бесплатные версии СУБД для реализации приложения.

Исходя из всего вышеперечисленного взгляд упал на СУБД SQLite.

SQLite — встраиваемая Open Source система управления базой данных.

СУБД написана на языке Си которая имеет ряд значительных преимуществ:

— SQLite не использует парадигмы клиент-сервер, то есть движок SQLite не является отдельно работающим процессом, с которым взаимодействует программа, а представляет собой библиотеку, с которой программа компонуется, и движок становится составной частью программы. Таким образом, в качестве протокола обмена используются вызовы функций (API) библиотеки SQLite. Такой подход уменьшает накладные расходы, время отклика и упрощает программу.

— SQLite поддерживает динамическое типизирование данных.

— Есть прямая поддержка Python.

— есть удобный графический интерфейс создания баз данных, а так же формирования запросов.

для создания графического интерфейса воспользуемся библиотекой PySide

6.

PySide6 — open Source API реализованное на основе графической биб-

библиотеки PyQt. Для удобного построения окон и работы с элементами предусмотрена платформа QT Designer.

После завершения работы над хранилищем данных необходимо думать над основным приложением. В качестве языка программирования выбран Python.

Python — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью. Язык является полностью объектно-ориентированным в том плане, что всё является объектами. Python является мультипарадигменным языком программирования, поддерживающим императивное, процедурное, структурное, объектно-ориентированное программирование. Для работы с ним будет использоваться среда разработки PyCharm разработанная компанией JetBrains на основе IntelliJ IDEA.

За основу будет выбран язык программирования Python для интеграции библиотеки MediaPipe. Так же необходимо подключить OpenCV.

OpenCV - Open Source Computer Vision Library, библиотека компьютерного зрения с открытым исходным кодом — библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. Реализована на C/C++. Она дает удобные возможности для работы с фотографиями, их демонстрацией, конвертацией и прочими алгоритмами.

Опишем отдельный класс цель которого создавать хэш исходя из входных параметров. Для этого будем использовать библиотеку hashlib для языка Python. В качестве хэш функции будем использовать SHA-256.

SHA-256 - алгоритм использующийся для создания отпечатков данных для сообщений произвольной длины. Применяются в различных приложениях или компонентах, связанных с защитой информации. Алгоритм опубликован Национальным институтом стандартов и технологий в федеральном стандарте обработки информации FIPS PUB 180-2 в августе 2002 года.

Создав весь необходимый комплекс программных модулей осталось ответить на 1 вопрос. Где все это применить? Немного подумав на ум приходит множество различных возможностей и сфер в которые данная программа легко войдет. Интегрируем проект в видео игру.

Создадим 3D модель персонажа и попробуем заставить его двигаться в соответствии с тем как перемещается реальный человек.

MakeHuman — это бесплатное промежуточное ПО для компьютерной 3D-графики с открытым исходным кодом, предназначенное для прототипирования фотореалистичных гуманоидов.

Добавим внутрь объекта так называемые кости для возможности управления. Это создает некоторый уровень абстракции который отделяет человека из реального мира где для движение рук и ног используются мышцы крепящиеся к костям от прототипа в компьютере для получения должной производительности и отказа от ненужных просчетов.

Импортируем объект в формате .bxf. FBX — проприетарный недокументированный формат. Он содержит модель, которая может использоваться и использоваться совместно с различными программами цифрового моделирования и создания контента. FBX может храниться на диске в виде бинарных или ASCII данных, FBX SDK поддерживает оба формата. Файлы FBX часто используются в разработке игр и анимации.

Воспользуемся средой разработки Unity для создания создания прототипа приложения.

Unity — кроссплатформенная среда разработки компьютерных игр, разработанная американской компанией Unity Technologies. Unity позволяет создавать приложения, работающие на более чем 25 различных платформах, включающих персональные компьютеры, игровые консоли, мобильные устройства, интернет-приложения и другие. Основным языком программирования для Unity является C sharp. C sharp - объектно-ориентированный язык программирования общего назначения. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга. C sharp относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов.

Перенесем в проект нашу трехмерную модель человека. В инспекторе можем регулировать основные параметры. Стоит сказать что Unity выстроена по модульному принципу. Каждый объект в сцене хранит лишь параметры класса Transform. Данный класс описывает положение объекта в пространстве. Что бы регулировать добавим свои параметры для этого опишем класс Person. Для каждой движущийся части тела опишем функцию которая отвечает за сохранение в ней вектора положения и углов Эйлера для поворота.

Что бы заставить 3D модель принять позу в соответствии с тем как необходимо необходимо переменным присвоить значения им соответствующие значения. Опишем функции которая на вход принимает объект файла. Пробивавшись по всем массивам выставим значения в необходимые переменные.

ЗАКЛЮЧЕНИЕ

В данной работе были отработаны навыки по применению библиотек компьютерного зрения в коммерческие проекты. Получен опыт работе с такими средами разработки как Unity, PyCharm, PySide Designer. Были отработаны и улучшены навыки написания программного кода, и работы с большим количеством API и интеграции его в проекты. Получено понимание в работе клиент серверных приложений.

Итогом работы стало написание нейронной сети позволяющая производить детекцию множества лиц на фотографии, а так же с использованием фреймворка MediaPipe реализованно приложение которое позволяет не просто производить детекцию людей но и определять местоположения конкретных частей тела в режиме реального времени. Были продемонстрированы варианты использования один из которых может упростить работу людям занимающиеся захватом движения для кинематографа видеоигр и средств дополненной реальности. Второй вариант может упростить жизнь людям которые по тем или иным причинам не могут обычными средствами управлять экраном мобильного телефона или компьютера.