

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ ГАГАРИНА Ю.А.»

ИНСТИТУТ ПРИКЛАДНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И
КОММУНИКАЦИЙ

Кафедра «Прикладные информационные технологии»

Направление 09.03.02 «Информационные системы и технологии»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
«Разработка программного интеллектуального комплекса
контроля безопасности. Модуль обнаружения оружия»

Студент Антонов Владимир Станиславович
фамилия, имя, отчество

группа МИФСТ-21

Руководитель

к.т.н., доцент кафедры «П
информационные технологии»

29.06.23

А.К. Кузьмин

должность, ученая степень, уч. звание

подпись, дата

Инициалы Фамилия

Допущен к защите

Протокол № ____ от «11» июня 2019 года

Зав. кафедрой «Прикладные информационные технологии»

доктор технических наук, доцент

29.06.23

С.В.Кумова

ученая степень, уч. звание

подпись, дата

Инициалы Фамилия

Саратов 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 АНАЛИЗ РАБОТЫ НЕЙРОСЕТЕВОЙ ОБРАБОТКИ ВИДЕОПОТОКА В РАМКАХ ОХРАННЫХ СИСТЕМ.....	8
2 ВЫБОР ПРОГРАММНЫХ СРЕДСТВ РАЗРАБОТКИ	16
2.1 ВЫБОР СИСТЕМЫ УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ	20
2.1.1 SQLITE.....	23
2.1.2 DB BROWSER FOR SQLITE	24
2.2 ВЫБОР ПРОГРАММНОЙ РЕАЛИЗАЦИИ РАЗРАБОТКИ ПРОГРАММНОГО ИНТЕЛЛЕКТУАЛЬНОГО КОМПЛЕКСА КОНТРОЛЯ БЕЗОПАСНОСТИ. МОДУЛЯ ОБНАРУЖЕНИЯ ОРУЖИЯ.....	26
2.2.1 TYPESCRIPT	26
2.2.2 NODE.JS.....	27
2.2.3 ANGULAR	28
2.2.4 YOLO.....	29
2.2.5 FLASK.....	29
3 ПРОЕКТИРОВАНИЕ НЕЙРОСЕТЕВОЙ СИСТЕМЫ	31
3.1 ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ НЕЙРОСЕТЕВОЙ СИСТЕМЫ.....	31
3.2 ФУНКЦИОНАЛ СИСТЕМЫ В ЗАВИСИМОСТИ ОТ УРОВНЯ ДОСТУПА	32
3.2.1 НЕАВТОРИЗОВАННЫЙ ПОЛЬЗОВАТЕЛЬ	32
3.2.2 АВТОРИЗОВАННЫЙ ПОЛЬЗОВАТЕЛЬ	32
3.3 ФУНКЦИОНАЛ ПОДСИСТЕМ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	33
3.3.1 СЕРВЕРНАЯ ЧАСТЬ.....	33
3.3.2 КЛИЕНТСКАЯ ЧАСТЬ	34

3.3.3	ФУНКЦИИ БД	35
3.3.4	ФУНКЦИИ НЕЙРОСЕТЕВОГО МОДУЛЯ	35
3.4	ВХОДНЫЕ ДАННЫЕ	36
3.4.1	ДАННЫЕ ПРИ РЕГИСТРАЦИИ ПОЛЬЗОВАТЕЛЯ	36
3.4.2	ДАННЫЕ О НЕЙРОСЕТЕВОМ МОДУЛЕ	36
3.4.3	ДАННЫЕ О ПОДПИСЧИКЕ	37
3.4.4	ДАННЫЕ О ПОДПИСКЕ	38
3.5	ВЫХОДНЫЕ ДАННЫЕ	38
3.6	СХЕМЫ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ В СИСТЕМЕ	39
3.7	ИСПОЛЬЗОВАНИЕ СТРУКТУРНЫХ МЕТОДОЛОГИЙ ДЛЯ ПОСТРОЕНИЯ МОДЕЛЕЙ ПРОЦЕССОВ НЕЙРОСЕТЕВОЙ СИСТЕМЫ	39
3.7.1	ДИАГРАММА IDEF0	40
3.7.2	ДИАГРАММА IDEF3	46
3.7.3	ДИАГРАММА DFD	51
3.7.4	ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ ПО МЕТОДОЛОГИИ UML	54
3.7.5	ДИАГРАММА USE CASE	55
3.7.6	ДИАГРАММА АКТИВНОСТИ	56
3.7.7	ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТИ	58
3.7.8	ДИАГРАММА КЛАССОВ	59
3.7.9	ДИАГРАММА ПАКЕТОВ	60
4	ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ	64
4.1	КДО-1 DOMINATION	64
4.2	НЕЙРОСЕТЬ ОТ СПЕЦЛАБ	65
4.3	GUNCHECK	66
4.4	АРСЕНАЛ+	68

4.5	СРАВНИТЕЛЬНЫЙ АНАЛИЗ СУЩЕСТВУЮЩЕГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	69
5	ПРОГРАММНАЯ РЕАЛИЗАЦИЯ НЕЙРОСЕТЕВОЙ СИСТЕМЫ	72
5.1	ПЕРЕЧЕНЬ КОМПОНЕНТОВ СИСТЕМЫ, ИХ НАЗНАЧЕНИЕ И ОСНОВНЫЕ ХАРАКТЕРИСТИКИ	72
5.2	ИСПОЛЬЗУЕМЫЕ API.....	72
5.3	АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	73
5.3.1	КЛАССЫ И ИНТЕРФЕЙСЫ	73
5.3.2	СЕРВЕРНАЯ ЧАСТЬ.....	74
5.3.2.1	API.....	74
5.3.2.2	ROUTES	75
5.3.2.3	CONTROLLERS	76
5.3.3	НЕЙРОСЕТЕВОЙ МОДУЛЬ.....	77
5.3.3.1	НЕЙРОСЕТЕВАЯ МОДЕЛЬ.....	78
5.3.3.2	ПОДГОТОВКА ДАТАСЕТА	83
5.4	ФУНКЦИОНАЛЬНЫЕ ПРОЦЕССЫ СИСТЕМЫ	90
5.4.1	ГОСТЬ.....	91
5.4.2	АДМИНИСТРАТОР.....	92
5.4.3	НЕЙРОСЕТЕВОЙ АНАЛИЗ	98
	ЗАКЛЮЧЕНИЕ.....	103
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	105

ВВЕДЕНИЕ

В стремлении различных компаний к автоматизации и оптимизации различных внутренних процессов, связанных с безопасностью своих сотрудников, в настоящее время все больше и больше обретают популярность нейросетевые системы, позволяющие без участия человека заблаговременно обнаружить угрозу на территории организации и уведомить соответствующие службы. Внедрение подобных систем и их последующее применение зачастую позволяет значительно снизить трудозатраты на повседневные задачи, поддающиеся автоматизации.

Так как большинство систем для безопасности предназначены для работы с непрерывным мониторингом видеопотока, в таких системах важная роль отводится анализу значительных объемов разнородных данных, зачастую не исключая высокую вероятность наличия ошибок из-за человеческого фактора. Исходя из этого, использование нейросетевых систем позволяет также исключить человеческий фактор там, где вмешательство человека не обязательно.

Автоматизация возможна для процессов в самых разных областях, например:

- Мониторинг и анализ видеопотоков;
- Прогнозирование и выявление угроз;
- Оповещение подписавшихся слушателей;

Стремление к автоматизации процессов, связанных с безопасностью, не обошло стороной и учебные заведения. К сожалению, горький опыт колледжа в Благовещенске в 2019-м году наглядно продемонстрировал необходимость внедрения нейросетевых систем в системы видеонаблюдения учебных заведений не только Благовещенска, но и на территории целой страны. С увеличением угрозы возникает потребность в автоматизации процессов, связанных с анализом видеопотока охранных систем и своевременного оповещения соответствующих служб.

Процесс анализа очень похож на стандартный процесс мониторинга видеопотока, выявления образов огнестрельного или холодного оружия, а так же и предметы, напоминающие оружие. Результат анализа должен выводиться на панель администратора в виде светового сигнала, а так же уходить в виде СМС-оповещения на телефон соответствующих служб. Нейросетевой анализ, в отличие от живых сотрудников, может проводить анализ непрерывно, круглосуточно, выключаясь только на время технического обслуживания.

Целью выпускной квалификационной работы является создание программного интеллектуального комплекса для контроля безопасности в постоянно расширяющихся компаниях и учебных заведениях и создание нейросетевого модуля для мониторинга попыток пронести огнестрельное оружие.

Задачи разработки программного интеллектуального комплекса для контроля безопасности включают:

- Автономность;

- Возможность добавлять модули для анализа других разновидностей угроз (например, потерянных/оставленных вещей или анализ критических эмоциональных состояний студента);
- Возможность контролировать процесс, корректировать работу отдельных модулей;
- Предоставление сбоеустойчивости системы (выход из строя, некорректная работа или техническое обслуживание одного отдельного модуля не должно повлечь за собой отказ всей системы);
- Возможность физического разбиения системы на несколько подсистем, которые будут работать на разных компьютерах, подключенных к одной локальной сети.

1 АНАЛИЗ РАБОТЫ НЕЙРОСЕТЕВОЙ ОБРАБОТКИ ВИДЕОПОТОКА В РАМКАХ ОХРАННЫХ СИСТЕМ

Прежде, чем приступить к изучению анализа работы нейросетевой обработки видеопотока, давайте вспомним, что такое нейронная сеть. Нейронные сети — модели биологических нейронных сетей головного мозга, в которых нейроны имитируются относительно простыми, однако часто однотипными элементами (так называемыми искусственными нейронами). Идея нейронных сетей родилась в рамках теории искусственного интеллекта, в результате попыток имитировать способность биологической нервной системы учиться и исправлять ошибки. В 1948 г. вышла книга Н. Винера о кибернетике. Основная идея заключается в представлении сложных биологических процессов математическими моделями[1]. Нейронные сети широко используются для решения различных задач.

Нейронные сети нашли себе применения в различных областях, таких как, например: автоматизация процессов распознавания образов, прогнозирование, адаптивное управление, создание экспертных систем, организация ассоциативной памяти, обработка аналоговых и цифровых сигналов, синтез и идентификация электронных схем и систем.

Нейросетевые модели могут представлять собой программное и аппаратное исполнение. В нейросетевой модели присутствуют искусственные нейроны, которые расположены послойно. В рамках одного слоя нейроны никак не связаны, однако между слоями нейроны связаны друг к другу посредством связи «все ко всем». Таким образом, нейросетевая модель представляет собой n -дольный полный граф,

где n — это количество слоев искусственных нейронов. Информация поступает от первого ко второму слою, от второго к третьему, и так далее.

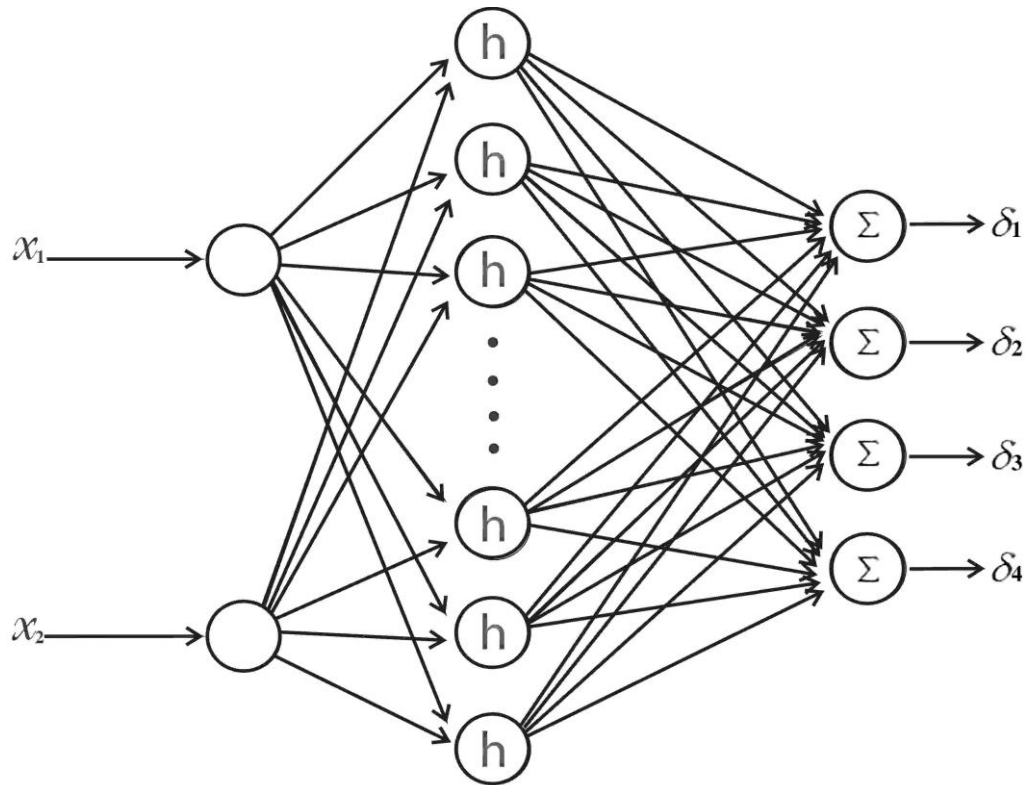


Рисунок 1 - Пример n -дольного полного графа, где $n = 3$

Поступление информации происходит за счет суммирования выходных сигналов всех нейронов с предыдущего нейронного слоя, который позже проходит через активационную функцию. Формула подсчета представлена на формуле 1.

$$\sum_i w_i x_i + f(b) \quad (1)$$

Количество слоев и нейронов в них определяют точность и достоверность результатов, получаемых при решении задач, т.е. чем больше слоев и нейронов на каждом слое, тем меньше ошибок и выше

надежность сети. Однако не стоит забывать что, если вы создадите слишком большую сеть, вы можете столкнуться со снижением производительности и увеличением сложности модели.

Поэтому, при выборе архитектуры сети, следует учитывать условия и контекст решаемой задачи. Поскольку нейронные сети строятся на искусственных нейронах, имеет смысл рассмотреть их структуру и функционирование.

Теперь подробнее рассмотрим, что из себя представляют нейроны и связи между ними. Искусственный нейрон представляют собой отдельный элемент нейронной сети, который имитирует деятельность нейрона головного мозга. Чаще всего нейрон принимает входные сигналы и пропускает их через адаптивный сумматор, после чего сигнал на выходе из сумматора поступает в нелинейный преобразователь для дальнейшего вычисления состояния нейрона, после чего там преобразуется функцией активации, и результат вычисления подается на выход (или в так называемую точку ветвления). Нейрон имеет аксон – выходную связь данного нейрона, при помощи которого сигнал поступает на синапсы нейронов следующего нейронного слоя.



Рисунок 2 - Схема искусственного нейрона

На выходе из нейрона сигнал проходит через активационную функцию, вычисляющую выходной сигнал[2]. Выбор функции определяется спецификой задачи или ограничениями, накладываемыми некоторыми алгоритмами обучения. Дополнительную информацию можно найти в специальной литературе.

Далее стоит рассмотреть применения нейронных сетей. С их помощью успешно решается важная задача в области телекоммуникаций – нахождение оптимального пути трафика между узлами. Чтобы отделить сигналы, генерируемые злоумышленником, от шума и помех, третья и заключительная часть обработки в системе биологической обратной связи выполняет анализ данных по принципу нейронной сети. Использование нейронной сети обеспечивает высокую надежность обнаружения при низком уровне ложных срабатываний.

В многослойной архитектуре нейроны расположены в несколько слоев. Нейроны первого слоя принимают входные сигналы, преобразуют их и передают через точки ветвления нейронам второго слоя.

С улучшением производительности современных систем, усложнением архитектуры нейронных сетей и увеличением качества обнаружения и классификации объектов недолго предположить, что нейронные сети нашли себе применение и в анализе видеопотока и охранных системах. В системах охранной сигнализации нейронная сеть представляет собой вычислительную систему, в которой алгоритм решения задач представлен в виде сети пороговых элементов с динамически настраиваемыми коэффициентами и алгоритмами настройки, не зависящими от размерности сети пороговых элементов и их входного пространства[3].

Классическая видеоаналитика уже не справляется с поставленными задачами: число видеокамер, интегрированных в различные системы безопасности, постоянно растет. Количество поступающих от камер событий тоже многократно увеличивается. Это приводит к большому количеству ложных срабатываний — оператор физически не способен эффективно обработать весь поток событий от камер в реальном времени.

Если заранее настроить типы объектов реагирования камеры, количество ложных срабатываний уменьшится. Но для этого нужно наделить систему видеонаблюдения функциями, присущими человеческому мозгу, научить ее давать оценку происходящему и самостоятельно принимать решения. Именно эту функцию и обеспечивает внедрение нейронных сетей. При грамотной реализации применение нейротехнологий позволяет:

- Избежать ложных срабатываний и сфокусировать внимание операторов на действительно важных событиях;

- Разгрузить сотрудников службы безопасности и повысить эффективность их работы;
- Сортировать информацию в архиве и быстро находить нужные данные при разборе инцидента;
- Решать сложные задачи, не доступные к выполнению классическими алгоритмами[4].

Автоматизация процесса мониторинга с применением нейросетевых технологий не лишена ряда своих преимуществ и недостатков. Начнем, пожалуй, с рассмотрения последних:

- Требования к программному и аппаратному обеспечению. Использование нейросетевого анализа, нейросетевых модулей и подобных технологий очень требовательный к системе процесс. Однако многие крупные фирмы в качестве серверов для своих локальных сетей уже давно использует высокопроизводительные вычислительные системы, которые без особых проблем запустят нейросетевые модули (не стоит забывать, что разрабатываемая система может быть физически разбита между несколькими компьютерами);

- Обслуживание. За работой системы следует следить уже специальным специалистам, которые смогут оказать поддержку программному обеспечению на должном уровне. Следить стоит и за частыми обновлениями фреймворков, используемых в данном приложении, чтобы система даже через несколько лет использовала актуальные технологии. Однако специалисты, проводящие обслуживание оборудованию в любом случае уже давно присутствуют во многих компаниях, где это необходимо.

- Система лишена усталости. Долгая, утомительная и зачастую однообразная работа людей нередко может приводить к искажению восприятия, систематических ошибок. Перечислять их все не имеет смысла, поскольку каждая такая ошибка не именуемо может привести к так называемому «человеческому фактору»[5].

Рисунок 3 - Список когнитивных искажений

Справедливости ради стоит отметить, что нейросетевые технологии не лишены своих когнитивных искажений, а скорее речь идет о том, что данная система не подвергается утомлению, в отличие от человека, что иногда приводит к усилению искажению восприятия.

Проведенный анализ нейросетевых технологий позволяет сделать вывод о том, что необходимо разработать программный комплекс на основе нейросетевого анализа, которое поможет оптимизировать и автоматизировать процесс анализа видеопотока с камер видеонаблюдения охранных систем для компаний и учебных заведений.

Функционала разрабатываемого программного комплекса для анализа и контроля безопасности будет достаточно для полного цикла работ по автоматизации анализа и контроля безопасности. Одной из основных функций является автоматический анализ видеопотока на предмет угроз. Также масштабируемость системы, которая проявляется в возможности добавления новых нейросетевых модулей, которые смогут анализировать на предмет угроз. Кроме того, классификация типа угроз позволяет оценить степень угрозы. Все описанные функции предоставляют возможность для полноценной работы по оценке и анализу видеопотока.

2 ВЫБОР ПРОГРАММНЫХ СРЕДСТВ РАЗРАБОТКИ

Исходя из поставленной задачи и перечня требований к нейросетевой системе, а так же для минимизации трудозатрат для разработки были выбраны следующие программные компоненты и фреймворки:

SQL-Lite – это библиотека на языке C, которая реализует небольшой, но весьма быстрый, автономный, высоконадежный, полнофункциональный механизм базы данных SQL. SQLite – самая используемая СУБД в мире. Данная СУБД может быть встроена во все мобильные телефоны и большинство компьютеров и входит в состав бесчисленного множества других приложений, которые люди используют каждый день[6].

Express.js – минималистичный и гибкий фреймворк для Node.js-веб-приложений, обеспечивающий набор возможностей для построения одно- и многостраничных и гибридных веб-приложений[7]. Один из наиболее привлекательных аспектов Express является его минималистичность. Множество раз разработчики фреймворков забывали, что обычно лучше меньше, да лучше. Философия Express заключается в обеспечении минимальной прослойки между вашими мозгами и сервером. Другим ключевым аспектом философии Express является расширяемость. Express предоставляет вам чрезвычайно минималистичный фреймворк, и вы можете добавлять функциональность в различные части Express по мере необходимости, заменяя все, что вам не подходит.

jQuery – это библиотека JavaScript (т.е., она написана на JavaScript), предназначенная для абстрагирования, выравнивания, исправления и

упрощения скриптинга при работе с узлами HTML-элементов в браузере или для работы в браузере без графического интерфейса[8].

Node.js – это кроссплатформенная среда с открытым исходным кодом, построенная на движке Google Chrome JavaScript V8. Он предоставляет библиотеку JavaScript и среду выполнения для выполнения кода. Node.js обеспечивает преимущество перед другими технологиями в разработке быстрых масштабируемых веб-приложений с интенсивным вводом-выводом, таких как одностраничные приложения (SPA), сайты потокового видео, приложения реального времени с интенсивным использованием данных и т. д. Node.js использует управляемую событиями неблокирующую модель ввода-вывода, которая делает ее легкой и эффективной и идеально подходит для приложений реального времени с интенсивным использованием данных, которые работают на распределенных устройствах. Node.js позволяет запускать код JavaScript на сервере. Не рекомендуется использовать Node.js для приложений, интенсивно использующих процессор[9].

Angular – является одним из фреймворков, реализованных на JavaScript и одним из самых популярных инструментов разработки программного обеспечения на сегодняшний день. Он был представлен Google в 2009 году и получил теплые рекомендации от сообщества разработчиков. Согласно опросу StackOverflow 2022 года, 23% инженеров-программистов применяют фреймворк для создания пользовательских интерфейсов[10].

Python – это мощный язык программирования общего назначения, который разработал Гвидо Ван Россум (Guido van Rossum) в 1989 году. Python классифицируется как язык программирования высокого уровня, в

котором автоматически обрабатывается большинство фундаментальных операций (таких как, управление памятью), выполняемых на уровне процессора («машинный код»). Python считается языком более высокого уровня, чем, например, C, из-за его выразительного синтаксиса (который во многих случаях близок к естественному языку) и богатого разнообразия встроенных структур данных, таких как списки, кортежи, множества и словари. Например, рассмотрим следующую программу на Python, которая выводит список имен в отдельных строках[11].

Flask – минималистичная «микро» веб-инфраструктура, которая уже около десяти лет лежит в основе веб-сообщества Python. Он нашел свое применение в огромном количестве приложений, включая аспекты платформ LinkedIn и Pinterest, а также бесчисленное множество других коммерческих, частных и исследовательских проектов. Эта популярность гарантирует, что у него есть динамичная экосистема расширений, документации и руководств[12].

Tensorflow – это интерфейс для выражения алгоритмов машинного обучения и реализация для выполнения таких алгоритмов. Вычисление, выраженное с помощью TensorFlow, может быть выполняется с небольшими изменениями или без изменений в самых разных разнородных системах, начиная от мобильных устройств, таких как телефоны и планшеты до крупномасштабных распределенных систем из сотен машин и тысячи вычислительных устройств, таких как Карты графического процессора[13].

YoLoy5 – (You Only Look Once) это семейство моделей, которое («PJ Reddie») Джозеф Редмон первоначально придумал в публикации 2016 года. Модели YOLO печально известны своей высокой

производительностью, но невероятно малыми размерами, что делает их идеальными кандидатами для условий реального времени и сред развертывания на устройстве[14].

ultrams.com – это многофункциональный API для WhatsApp и Best Tool для предприятий и программистов, который можно интегрировать в любую систему учета клиентов, CRM, ERP или веб-сайт для отправки сообщений, уведомления пользователей и многого другого[15].

Помимо вышеперечисленного, в данной системе будет активно использоваться технология TCP-IP. Системы, основанные на сокетах и TCP/IP, распределяют состояние сеанса по обеим сторонам двухточечного соединения. Распределенное состояние приводит к трем проблемам. Во-первых, аварийное переключение и восстановление соединения затруднено, непрозрачно или и то, и другое, поскольку восстановление потерянного состояния часто является нетривиальной задачей. Сбои веб-сервера, например, может привести к заметному пользователю сбросу соединения. Во-вторых, выделение ресурсов для сохранения состояния провоцирует атаки типа «отказ в обслуживании» (DoS), которые расходуют эти ресурсы. Средства защиты от таких атак часто отключают полезную функциональность: лишь немногие стеки принимают дополнительные данные в пакетах SYN, что увеличивает нагрузку на короткие соединения, а интернет-серверы часто не позволяют запуск постоянных HTTP-соединений, что увеличивает накладные расходы при пакетном доступе[16]. Наконец, состояние в стеках протоколов ограничивает масштабируемость: серверы не могут масштабироваться до большого количества клиентов, потому что им необходимо выделять ресурсы для каждого клиента, и аналогичным образом не могут

масштабироваться до крошечных встроенных устройств, поскольку существует нижняя граница ресурсов, необходимых для соединения[17].

Использование указанных технологий имеет следующие ключевые преимущества:

- Все части системы (база данных, серверная и клиентская части) могут быть распределены между несколькими физическими устройствами, тем самым уменьшить нагрузку на каждый элемент системы по отдельности;
- Данная система использует стандартизированное подключение к модулям, что позволит без особых проблем наращивать функционал системы посредством добавления новых модулей;

2.1 ВЫБОР СИСТЕМЫ УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ

На данный момент была поставлена задача подобрать легкую, простую в использовании и не требующую сложных настроек СУБД. Из обширного списка СУБД выбор пал на несколько, ближе всего подходящих.

Наиболее подходящие к представленным требованиям являются такие СУБД как MySQL и SQLite. Маловесные, простые в использовании и не имеющих высоких требований к системам. Однако Node.js предоставляет инструменты для работы только для SQLite.

СУБД SQLite распространяется бесплатно, но из-за этого проигрывает в функционале большинству СУБД. Однако стоит отметить, что функционала SQLite вполне достаточно для разрабатываемой системы.

Реляционная база данных представляет современная форма хранения и упорядочения данных в базе данных в виде интуитивно понятной таблицы. Каждая таблица представляет собой определенную абстрактную сущность, имеющие свои параметры (например, таблица «Пользователь»). Каждая колонка данной таблицы представляет собой параметры данной сущности (например, «Имя», «Фамилия»). Не стоит только путать таблицу, как сущность, и конкретный объект этой сущности. Объектом сущности уже будет запись или строка в данной таблице.

Такой подход весьма удобен при работе с данными, так как имеет определенную структуру. Все записи расположены в таблице, имеющей соответствующее название.

Таблицы в базе данных могут быть связаны между собой посредством, так называемых, внешних ключей. Например, таблица Подписка будет иметь поля ID_пользователя и ID_модуля. Таким образом, связь многие-ко-многим будет разбита на две связи один-ко-многим, так как на один модуль может подписаться несколько человек, а один человек может подписаться на несколько модулей.

Чтобы получить данные из базы данных, необходимо отправить SQL-запрос, который вернет результат так же в виде таблицы. Примером SQL-запроса рассмотрим выборку всех записей подписок на модули с определенным названием, результат которого представлен в таблице 1.

```
SELECT * FROM subscriptions_number_to_module  
WHERE name_module='weapon'
```

Таблица 1 - Результат запроса

	id	name_module	phone
1	15	weapon	79995391017
2	17	weapon	79372507888
3	18	weapon	79379614107

Инструменты Node.js без лишних действий помогут разработчику превратить подобный результат в JSON-подобное сообщение, которое можно уже передавать между модулями посредством HTTP-запроса.

```
result = [  
  {  
    id : 15,  
    name_module: 'weapon',  
    phone: 79995391017  
  },  
  {  
    id : 17,  
    name_module: 'weapon',  
    phone: 79372507888  
  },  
  {  
    id : 18,  
    name_module: 'weapon',  
    phone: 79379614107  
  },  
]
```

Выше представлен результат того же запроса в виде JSON. JavaScript Object Notation (или сокращенно JSON) – облегченный формат,

основанный на подмножестве синтаксиса JavaScript. Одним из лучших моментов в JSON является то, что его легко читать и писать вручную, в гораздо большей степени, чем что-то вроде XML. Мы можем естественным образом разобрать его с помощью JavaScript, потому что он частично использует тот же синтаксис[18]. JSON формат активно используется в структуре передачи данных в HTTP запросах. Протокол передачи гипертекста является одной из основ Интернета. Каждый веб-разработчик, даже фронтенд-разработчик, должен иметь хотя бы базовое представление о том, что такое HTTP. Это механизм, который позволяет компьютерам общаться друг с другом через Интернет. Он определяет формат, в котором сообщения передаются в Интернете. Это абсолютно необходимо для каждого веб-разработчика[19].

2.1.1 SQLITE

Помимо описанных выше SQLite имеет следующие преимущества:

- Файл *.db можно размещать в любом месте на компьютере (в папке с проектом), что никак не скажется на работе системы;
- СУБД не требует установки на компьютер дополнительных фреймворков или ПО;
- Скорость работы: SQLite отлично подходит для задач малых и средних проектов.

Для администрирования БД разработчиком предлагается программное обеспечение DB Browser for SQLite, написанный в основном на языке программирования C++. Данная программа является

высококачественным визуальным инструментом с открытым исходным кодом для создания, проектирования и редактирования файлов базы данных.

2.1.2 DB BROWSER FOR SQLITE

DB Browser for SQLite – это высококачественный визуальный инструмент с открытым исходным кодом для создания, проектирования и редактирования файлов базы данных, совместимых с SQLite. Он предназначен для пользователей и разработчиков, желающих создавать базы данных, искать и редактировать данные. Он использует знакомый интерфейс, похожий на электронную таблицу, и вам не нужно изучать сложные команды SQL.

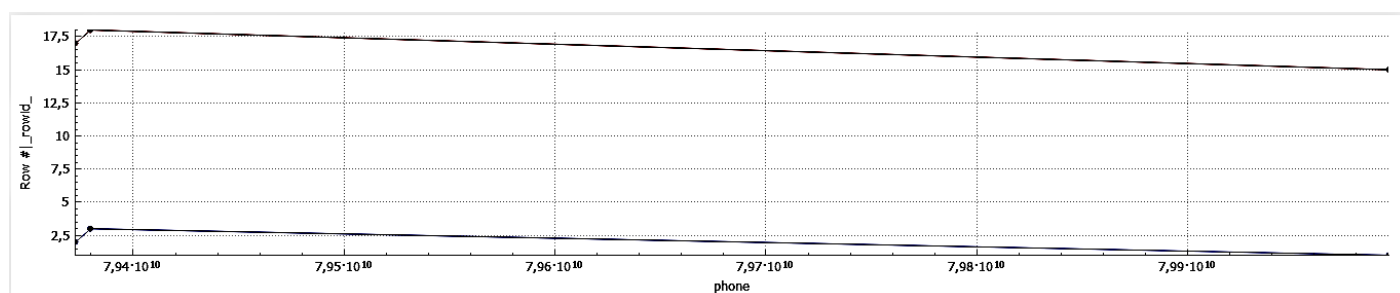


Рисунок 4 - Анализ данных в схеме

Данная программа предоставляет достаточно большой инструментарий анализа операций, проводившихся с таблицами базы данных. На рисунке 4 приведен пример подобного графика, в котором показана зависимость количества столбцов к значению. Таким образом, можно отследить, как меняется значение в определенном столбце.

	name	r_name	ip	port	online_status	ageCooll	message_Attentior
	Фильтр	Фильтр	Фильтр	Фильтр	Фильтр	Ф...	Фильтр
1	weapon	Оружие	127.0.0.1	5003	offline	10	Была зафиксир...
2	emo	Эмоциональ...	10.0.0.113	10201	offline	10	Был зафиксиро...
3	lost	Забывтые/ост...	10.0.0.192	5003	offline	10	Был обнаруже...

Рисунок 5 - Работа с данными, представленными в виде таблицы

На рисунке 5 представлен пример таблицы, хранившей информацию о модулях данной системы. В таблице хранится название модуля, ip-адрес и порт, так как модули могут располагаться на разных физических устройствах, подключенных между собой в локальную сеть первого уровня. В таблице также указан онлайн-статус модуля и количество секунд, которое должно пройти между отправками СМС-оповещение подписчика.

Основные возможности DB Browser for SQLite:

- Визуализация анализа данных, статистика по ним;
- Визуализация данных в виде таблиц и зависимостей;
- Операции с БД, в число которых входят: создание новых таблиц (коллекций), индексов, хранимых процедур и многое другое.

2.2 ВЫБОР ПРОГРАММНОЙ РЕАЛИЗАЦИИ РАЗРАБОТКИ ПРОГРАММНОГО ИНТЕЛЛЕКТУАЛЬНОГО КОМПЛЕКСА КОНТРОЛЯ БЕЗОПАСНОСТИ. МОДУЛЯ ОБНАРУЖЕНИЯ ОРУЖИЯ

Разрабатываемая нейросетевая система предполагает взаимодействие с БД, оперированием данными в ней, взаимодействие с нейросетевыми модулями, представлением результата анализа. Для выполнения вышеперечисленных задач было необходимо реализовать клиентскую, серверную и модульную логики, которые будут передавать данные посредством TCP-IP.

Для написания серверной части, как было уже указано, был выбран Node.js, так как он прост в применении, распространяется бесплатно и превосходная интеграция с SQLite. Для написания интерфейса был выбран Angular, так как подобно Node.js распространяется бесплатно, позволяет написать одностраничные приложения, которые сохраняют состояние даже в режиме offline. Для написания нейросетевых модулей был выбран питон, так как представляет широчайший спектр инструментов в работе с нейронными сетями и машинным обучением.

2.2.1 TYPESCRIPT

TypeScript – это язык программирования, который вхож в семейство JavaScript, который был разработан разработанный и до сих пор поддерживается Microsoft. Язык имеет статическую типизацию данных, которая используется по умолчанию в современных программных продуктах. Помимо этого, TypeScript имеет структурную типизацию. Что

это такое? Это подход, при котором указывают не то, как называется тип или где он определяется, а на то, что он описывает внутри[20].

2.2.2 NODE.JS

Node.js представляет собой среду выполнения, которая может выполнять приложения JavaScript. Значительным преимуществом Node.js, которая привлекает многих разработчиков, так это его управляемая событиями архитектура для асинхронного программирования. При таком подходе функции (также известные как обратные вызовы) могут быть привязаны к определенным событиям; когда происходит прикрепленное событие, выполняется обратный вызов. Таким образом, разработчику не нужно писать основной цикл, так как об этом позаботится среда выполнения.

Node.js так же имеет новые асинхронные функции, которые используют другой синтаксис, так что они слишком хорошо скрывают архитектуру, управляемую событиями, чтобы использовать их в статье с практическими рекомендациями[21].

Преимущества Node.js:

- **Скорость:** Движок V8 постоянно расширяет границы и является одним из самых быстрых интерпретаторов динамических языков на планете;
- **Масштабируемый:** Благодаря простоте и удобству работы с менеджером пакетов для Node.js, который называется npm, любой

разработчик может свободно использовать более полумиллиона опенсорсных пакетов;

- Лицензия: Node.js выпущен под лицензией MIT (открытого программного обеспечения).

2.2.3 ANGULAR

Для реализации пользовательского интерфейса был выбран Angular, так как этот язык способен реализовать SinglePageApplication (далее SPA), то есть одностраничное решение, что позволит на долгое время сохранять состояние приложения во время длительного сеанса, быстро компилировать интерфейс и предоставлять возможность для интерактивного взаимодействия с элементами UI. Высокое быстродействие, упрощенный поиск ошибок, эффективное кеширование данных и многое другое являются неотъемлемыми преимуществами Ангуляра.

Основные возможности Angular: двустороннее связывание, позволяющее динамически изменять данные в одном месте интерфейса при изменении данных модели в другом, шаблоны, маршрутизация и так далее. Одной из ключевых особенностей Angular является то, что в качестве языка программирования используется TypeScript. Несмотря на то, что другие фреймворки пользуются большей популярностью, разработчики компаний все равно продолжают использовать его в своих проектах[22].

2.2.4 YOLO

You Only Look Once (далее YoLo) популярная модель трекинга объектов и сегментации изображений была создана Джозефом Редмоном и Али Фархади из Вашингтонского университета. Запущенный в 2015 году, YOLO быстро завоевала популярность благодаря своей высокой скорости и точности.

Обнаружение объектов – одна из наиболее важных задач компьютерного зрения. В двух словах, по изображению детектор объектов найдет:

- объекты на изображении или видео;
- их типы (обычно называемые классом);
- ограниченный прямоугольник, представляющий координаты объекта на изображении[23].

2.2.5 FLASK

Для реализации нейросетевого модуля (который разрабатывается на python), необходимо подобрать подходящий фреймворк, который позволит модулю взаимодействовать с клиент-серверной частью разрабатываемой системы.

Flask – это микрофреймворк для создания веб-приложений на языке Python. Flask использует набор инструментов Werkzeug и Jinja 2. Werkzeug - это инструментарий для WSGI - стандартного интерфейса Python между веб-приложениями и различными серверами, предназначен как для

разработки, так и развёртывания. Jinja2 занимается отображением шаблонов[24]. Flask работает при версии Python 2.6 и выше. В основном Flask создали для решения простых задач. Простые задачи должны решаться, не требуя большого объема кода. Для создания веб-приложения используется виртуальное окружение. В виртуальном окружении хранятся используемые библиотеки.

Проанализировав все доступные технологии для реализации программного комплекса, можно сделать следующие выводы: для реализации возможности масштабируемости, необходимо реализовать систему в виде сетевой топологии «звезда» - то есть система должна иметь центральный модуль. Для взаимодействия между модулями можно использовать протокол TCP-IP. Протокол http отлично подходит для этого. Разрабатываемый программный комплекс будет реализован в виде клиент-серверного веб-приложения.

3 ПРОЕКТИРОВАНИЕ НЕЙРОСЕТЕВОЙ СИСТЕМЫ

Подсистема обнаружения фактов проноса оружия – это программное средство для мониторинга и обнаружения попытки проноса оружия на территорию университета. В случае с машинным обучением подсистема автоматизирует процесс мониторинга и обнаружения попыток проноса оружия, а так же обеспечит своевременное оповещение.

Мониторинг начинается с распределения задач между компонентами, которые выполняются на основе анализа нейросетевыми модулями. После этого результат фиксируется в системе и дальше проводится анализ уровня угрозы.

3.1 ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ НЕЙРОСЕТЕВОЙ СИСТЕМЫ

Подсистема обнаружения фактов проноса оружия на территорию организации предоставляет следующие возможности:

- Обеспечение автономности работы системы;
- Разделение обязанностей между модулями системы;
- Хранение данных о состоянии модулей, подписчиков и статусе подписок;
- Управление нейросетевым модулем с отдельного интерфейса;
- Регистрация новых нейросетевых модулей;

- Управление подписками с помощью пользовательского интерфейса;
- Добавление новых подписчиков и проверка их статуса подписки;
- Авторизация и Регистрация;
- Просмотр видеопотока и классификация угрозы;
- Оповещение при помощи SMS-сообщения в WhatsApp;

Нейросетевая система должна обеспечивать безопасность доступа к функционалу системы, поэтому была добавлена возможность авторизации.

3.2 ФУНКЦИОНАЛ СИСТЕМЫ В ЗАВИСИМОСТИ ОТ УРОВНЯ ДОСТУПА

3.2.1 НЕАВТОРИЗОВАННЫЙ ПОЛЬЗОВАТЕЛЬ

Неавторизованный пользователь должен иметь возможность:

- Зарегистрироваться;
- Авторизоваться.

3.2.2 АВТОРИЗОВАННЫЙ ПОЛЬЗОВАТЕЛЬ

Авторизовавшись, пользователь получает в руки все права доступа администратора, который может уже управлять системой при помощи пользовательского интерфейса. Предоставлять доступ стоит только

ограниченному кругу лиц. Зарегистрированный пользователь получает такие возможности, как:

- Управление нейросетевым модулем с отдельного интерфейса;
- Регистрация новых нейросетевых модулей;
- Управление подписками с помощью пользовательского интерфейса;
- Добавление новых подписчиков и проверка их статуса подписки.

3.3 ФУНКЦИОНАЛ ПОДСИСТЕМ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

3.3.1 СЕРВЕРНАЯ ЧАСТЬ

Как было указано выше система состоит из нескольких подсистем, одна из которых является серверная часть, или проще бекенд. В функционал бекенда входит:

- Осуществление авторизации пользователей;
- Осуществление регистрации пользователей;
- Получение и обработка запросов с клиентской части;
- Обеспечение связи с базой данных;
- Получение запросов от нейросетевого модуля;
- Осуществление подписки на нейросетевой модуль;

- Обработка запросов клиентской части;
- Отправление запросов в базу данных;
- Осуществление рассылки SMS-сообщений при помощи мессенджера WhatsApp;
- Отправление ответы на запросы из клиентской части;
- Отправка результатов нейросетевого анализа на клиентскую часть;
- Каждый определенный промежуток времени уточнять статус онлайн-подключения определенного модуля и его доступность системе;
- Обработка ошибки.

3.3.2 КЛИЕНТСКАЯ ЧАСТЬ

Функции бэкенда тесно связаны с функциями фронтенда, или клиентской части. Ниже представлен функционал клиентской части:

- Получение данных о пользователе;
- Отправка запросов на бэкенд;
- Вывод список нейросетевых модулей;
- Вывод список подписчиков;
- Изменение статуса подписки на нейросетевой модуль.

3.3.3 ФУНКЦИИ БД

Так же бекенд осуществляет связь с базой данных. База данных осуществляет следующие задачи:

- Осуществлять создание, чтение, изменение и удаление данных (CRUD-операции) по команде от сервера;
- Хранение данных длительное время;
- Обеспечивать защиту данных.

3.3.4 ФУНКЦИИ НЕЙРОСЕТЕВОГО МОДУЛЯ

Так же бекенд принимает результат анализа видео-потока с нейросетевого модуля. Ниже представлены основные функции нейросетевого модуля:

- Осуществление соединения с модулем захвата видеопотока;
- Принимать запрос на подписку бекенда;
- Предоставлять пользовательский интерфейс для настройки нейронной сети;
- Осуществление анализа видео-потока;
- Отправка результата на бекенд.

3.4 ВХОДНЫЕ ДАННЫЕ

3.4.1 ДАННЫЕ ПРИ РЕГИСТРАЦИИ ПОЛЬЗОВАТЕЛЯ

При регистрации пользователь должен вводить следующие данные:

- Фамилия;
- Имя;
- Номер телефона;
- Логин;
- Пароль.

3.4.2 ДАННЫЕ О НЕЙРОСЕТЕВОМ МОДУЛЕ

Во время создания нейросетевого модуля, система затребует несколько входных данных:

- Ключ или кодовое название – сокращенное название модуля, написанное латинскими буквами для обеспечения взаимодействия между пользователем и конкретным модулем (является уникальным);
- Название – расширенная версия ключа, которая необязательно должна быть уникальным. Поддерживает кириллицу и пробелы. Название должно раскрывать смысл действующего модуля;
- IP – адрес, который хранит в себе тот или иной модуль, грубо говоря, определенное физическое устройство, подключенное к сети,

на котором функционирует модуль. Если бекенд и модуль расположены на одном компьютере, данное поле стоит заполнять стандартными 127.0.0.1 или localhost;

- Port – адрес запущенного модуля уже на физическом устройстве. **ВАЖНОЕ ПРИМЕЧАНИЕ:** порт должен быть открытым для обеспечения взаимодействия между бекендом и модулем, однако в локальных сетях первого уровня порты по умолчанию открыты;
- Частота оповещений (в сек) – время между отправкой сообщениями при выявлении данным модулем угрозы. Указывается в количествах секунд;
- Сообщение – текст сообщения, который отправится в виде SMS подписчику.

3.4.3 ДАННЫЕ О ПОДПИСЧИКЕ

При регистрации подписчика, система потребует следующие данные:

- Фамилия;
- Имя;
- Номер телефона.

3.4.4 ДАННЫЕ О ПОДПИСКЕ

При оформлении подписки на определенный модуль, который происходит в один клик, система на самом деле автоматически собирает все необходимые данные и отправляет их на бекенд. Эти данные представлены ниже:

- Номер телефона;
- Ключ (латинское сокращенное название) модуля.

3.5 ВЫХОДНЫЕ ДАННЫЕ

При регистрации пользователя, выходными данными будет открытый UI-интерфейс системы с пройденной авторизацией.

При создании/изменении/удалении нейросетевого модуля на выходе пользователь увидит обновленный список всех модулей.

При регистрации подписчиков система так же выдаст обновленный список тех самых подписчиков.

При оформлении подписки система выдаст сообщение «*подписаться*» (если на данный момент пользователь не подписан) или соответственно наоборот «*подписан*».

3.6 СХЕМЫ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ В СИСТЕМЕ

Оперируя собранной информацией, необходимо построить модель процессов нейросетевой системы. Для построения моделей процессов нейросетевой системы на практике часто используют структурные методологии, которые помогают оценивать и выбирать технологии разработки программного обеспечения.

3.7 ИСПОЛЬЗОВАНИЕ СТРУКТУРНЫХ МЕТОДОЛОГИЙ ДЛЯ ПОСТРОЕНИЯ МОДЕЛЕЙ ПРОЦЕССОВ НЕЙРОСЕТЕВОЙ СИСТЕМЫ

Методология структурного анализа и проектирования ПО определяет руководящие указания для оценки и выбора технологий разработки для разрабатываемого ПО, шаги работы, которые должны быть выполнены, их последовательность, правила распределения и назначения операций и методов. Структурные методологии предлагают методику трансляции проектных спецификаций в модель реализации, в дальнейшем используемую при кодогенерации.

Можно выделить следующие принципы структурного метода:

- Разбиение задачи на более мелкие подзадачи;
- Построение иерархии подсистем;
- Планирование функционирования подсистем.

Структурные методологии разделяются на стандарты, демонстрирующие функциональную структуру системы и моделирующие структуру данных системы.

3.7.1 ДИАГРАММА IDEF0

Для описания бизнес-процессов, происходящий в разрабатываемом ПО, чаще всего прибегают к одному из наиболее популярных подходов - IDEF0.

IDEF0 - нотация графического моделирования, используемая для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальных объектов, связывающих эти функции. Стандарт IDEF0 (Integration Definition for Function Modeling) утвержден в США в 1993 как Федеральный стандарт обработки информации. В России находится в статусе руководящего документа с 2000 года и в настоящее время в качестве стандарта не утвержден. Тем не менее, методология IDEF0 является одним из популярных подходов для описания бизнес-процессов. К ее особенностям можно отнести:

- использование контекстной диаграммы;
- поддержка декомпозиции;
- доминирование;
- выделение 4 типов стрелок.

Контекстная диаграмма. Самая верхняя диаграмма, на которой объект моделирования представлен единственным блоком с граничными стрелками. Эта диаграмма называется А-0 (А минус ноль). Стрелки на этой диаграмме отображают связи объекта моделирования с окружающей средой. Диаграмма А-0 устанавливает область моделирования и ее границу[25]. На рисунке 6 представлен самый верхний уровень диаграммы подсистемы программного комплекса мониторинга безопасности на территории университета.

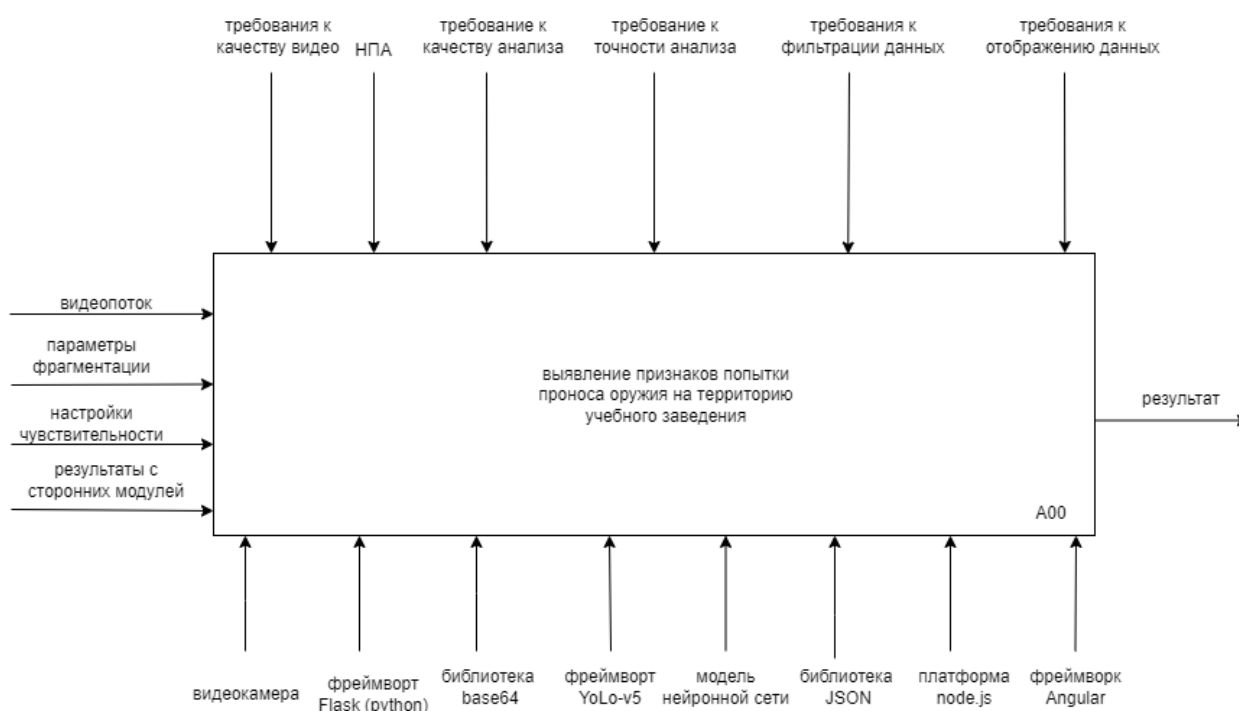


Рисунок 6 - Диаграмма IDEF0 разрабатываемой системы

В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность[26]. Так как система представлена в виде отдельных взаимодействующих компонентов, а диаграмма IDEF

предусматривает декомпозицию на любой уровень, мною была составлена декомпозиция уровня A0, которая представлена на рисунке 7.

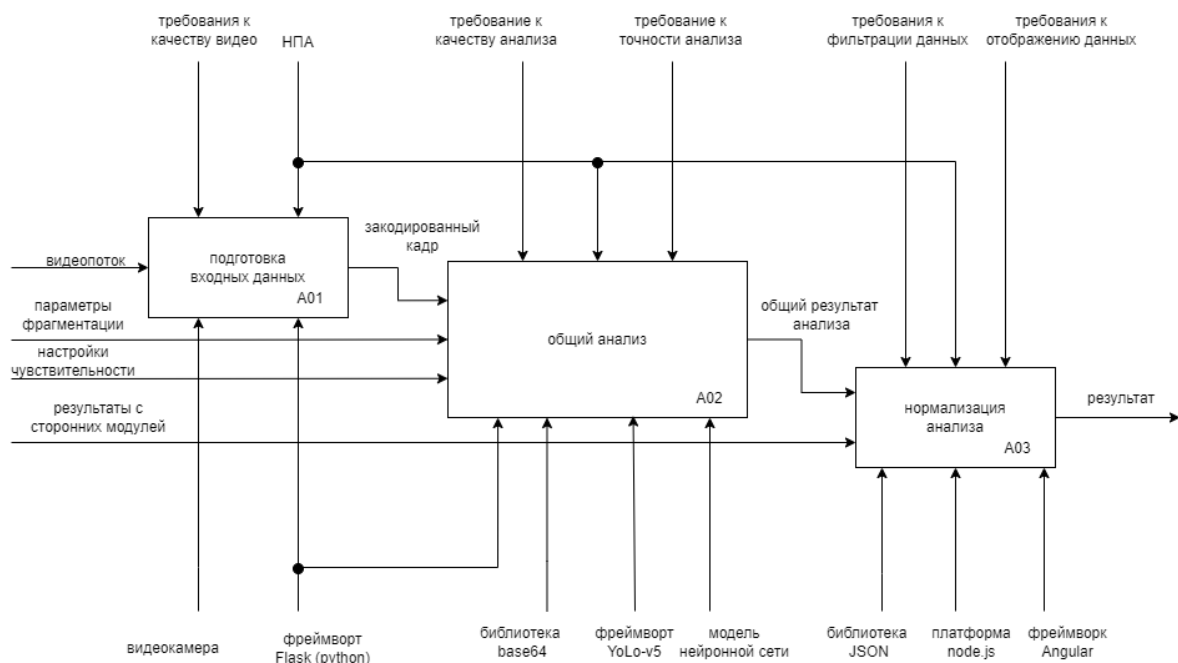


Рисунок 7 - Декомпозиция уровня A0

На рисунке выше представлена декомпозиция системы, на котором каждый блок – это отдельная функциональная единица (фронтенд, бэкенд или нейросетевой модуль). Их функционал так же можно разделить на более мелкие взаимодействия, представленные на рисунках 8-10.

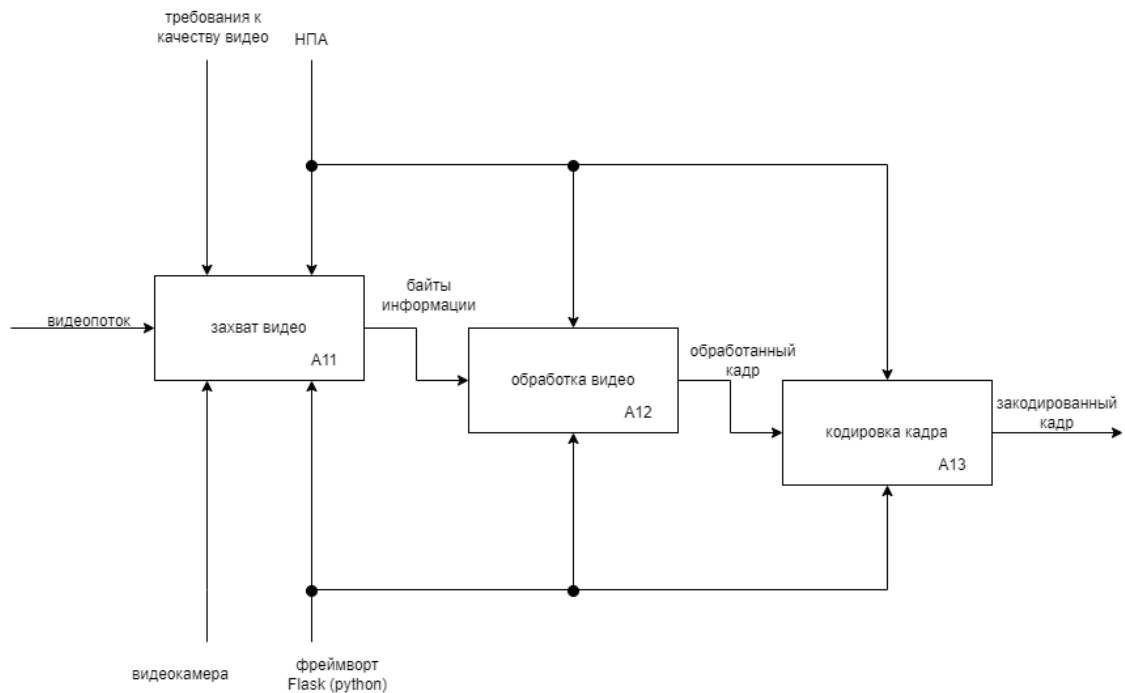


Рисунок 8 - Декомпозиция уровня A01 (представлена работа захвата изображения с камеры)

На рисунке 8 представлена декомпозиция уровня A-0-1, а именно процесса захвата изображения. Данная схема дает наглядное представление о внутренней структуре и процессах, приводящих к выходному результату. Первый этап захвата – непосредственно получение биты информации с физической камеры. Данные биты уже программными средствами преобразуются в изображение (кадр) на втором этапе. Изображение, для дальнейшей передачи нейросетевым модулям, для удобства кодируются в строку при помощи кодировка base64.

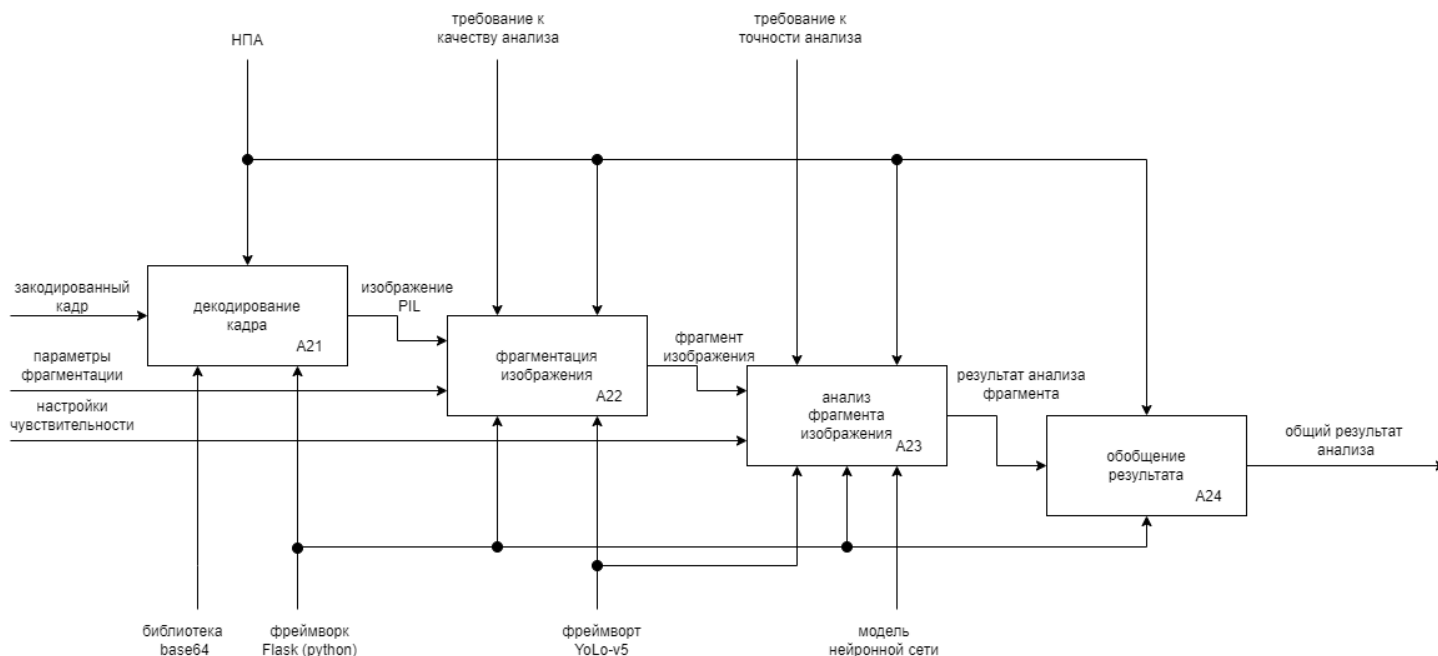


Рисунок 9 - Декомпозиция уровня A02 (представлена работа нейросетевого модуля)

На рисунке выше представлена декомпозиция работы нейросетевого модуля. Первый этап работы модуля (а именно нейросетевой анализ) заключается в получении закодированного кадра от модуля видеозахвата. Закодированный в base64 кадр преобразуется в массив данных, или, проще говоря, изображение в PIL-формате (далее, для упрощения, просто изображение или кадр). На втором этапе изображение подвергается сегментации. Это нужно для того, чтобы анализировать отдельные области изображения, так как нужно не только классифицировать объект в кадре, но и установить его положение на изображении. На третьем этапе происходит обычная классификация объекта, но взятого уже непосредственно с фрагмента кадра. Четвертый анализ нужен исключительно по требованию заказчика, так как подробная и более точная классификация огнестрельного оружия не нужна, поэтому все

возможные результаты, такие как автомат, пистолет и так далее, подаются под одним названием – оружие.

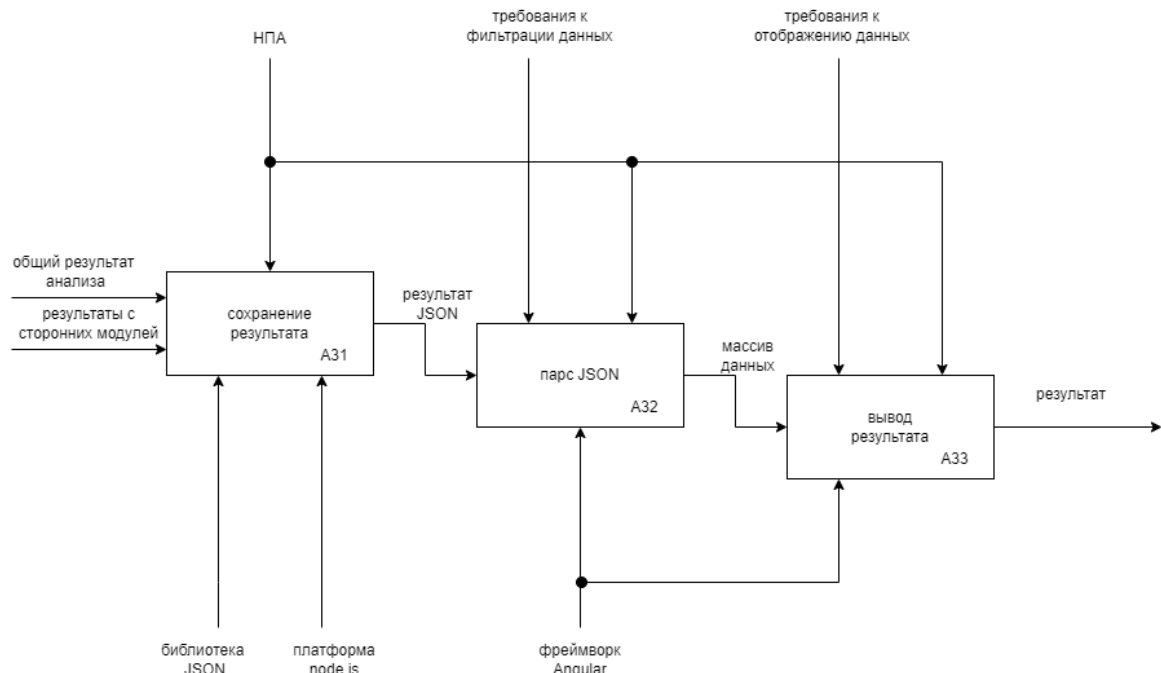


Рисунок 10 - Декомпозиция уровня A03 (представлена работа панели администратора)

С точки зрения работы с нейросетевого анализа данный модуль практически имеет самый простой функционал. На рисунке 10 можно заметить всего три шага визуализации результатов анализа. Первый шаг – это принять общий результат анализа и преобразовать его в JSON-формат для удобства дальнейшей работы на языке angular. JSON уже прекрасно обрабатывается при помощи данного языка, который преобразует определенный раздел пакета ответа с бэкенда в массив данных на втором этапе. Третий блок или шаг уже отвечает за визуализацию данных, а именно если в массиве есть значения – значит, индикатор угрозы окрашивается в красный цвет, если массив был пустым – в зелёный.

3.7.2 ДИАГРАММА IDEF3

Нотация IDEF3 была создана для описания одновременно технологических и бизнес-процессов. Чтобы понять, чем отличаются бизнес-процессы от технологических, вы можете ознакомиться с моей статьей «Что такое бизнес-процесс». Здесь же я кратко напомню, что в бизнес-процессе может быть несколько вариантов финала. Технологический представляет собой алгоритм, где результат всегда один: создание продукта (услуги). Кроме того, в бизнес-процессах всегда задействованы не только технологии, но и люди. Технологический процесс может быть автоматизирован полностью.

Нотация IDEF3 может быть использована:

- Как инструкция для сотрудников, которые будут работать в рамках бизнес-процесса;
- Как иллюстрация предлагаемых решений для заказчика (если есть возможность работать с компьютером, то можно также пользоваться декомпозицией, что особенно удобно);
- Как алгоритм для настройки IT-системы[27].

Плюсы IDEF3:

1. Простота. Элементов не так много, как в некоторых аналогичных решениях. В процессном моделировании используется всего 5 основных объектов;
2. Ограничение нотации определенными рамками (формат А-4). При моделировании важно, чтобы диаграмма не «расползалась» на

огромный лист, не была перегружена слишком большим количеством элементов, но при этом она должна быть информативной.

Для реализации сочетания лаконичности и максимальной информативности в IDEF3 присутствует возможность использования декомпозиции. Именно через нее объединяют в общей системе бизнеспроцессы и технологические. Например, вы можете создать нотацию бизнеспроцесса работы компании, где одним из элементов будет технологический процесс производства. Далее в рамках декомпозиции вы сможете развернуть технологический процесс в виде отдельного подпроцесса. Внешне нотация IDEF3 представляет собой набор из прямоугольников и стрелок внутри какого-то элемента. Это может быть: основной процесс; UOB (один из элементов основного процесса, для которого выполнена декомпозиция); нотация IDEF0.

Важные параметры диаграммы IDEF3: Графическая нотация всегда выполняется в черно-белом формате, цветовые решения IDEF3 не использует. Нотация (основная или декомпозиция элемента) должна читаться при распечатке в формате А-4. Элементы диаграммы выполняются в соответствии с правилами, которые определяют возможности взаимодействия между ними.

Техпроцесс можно рассматривать с разных точек зрения, так как пользоваться системой могут пользователи с разным уровнем доступа.

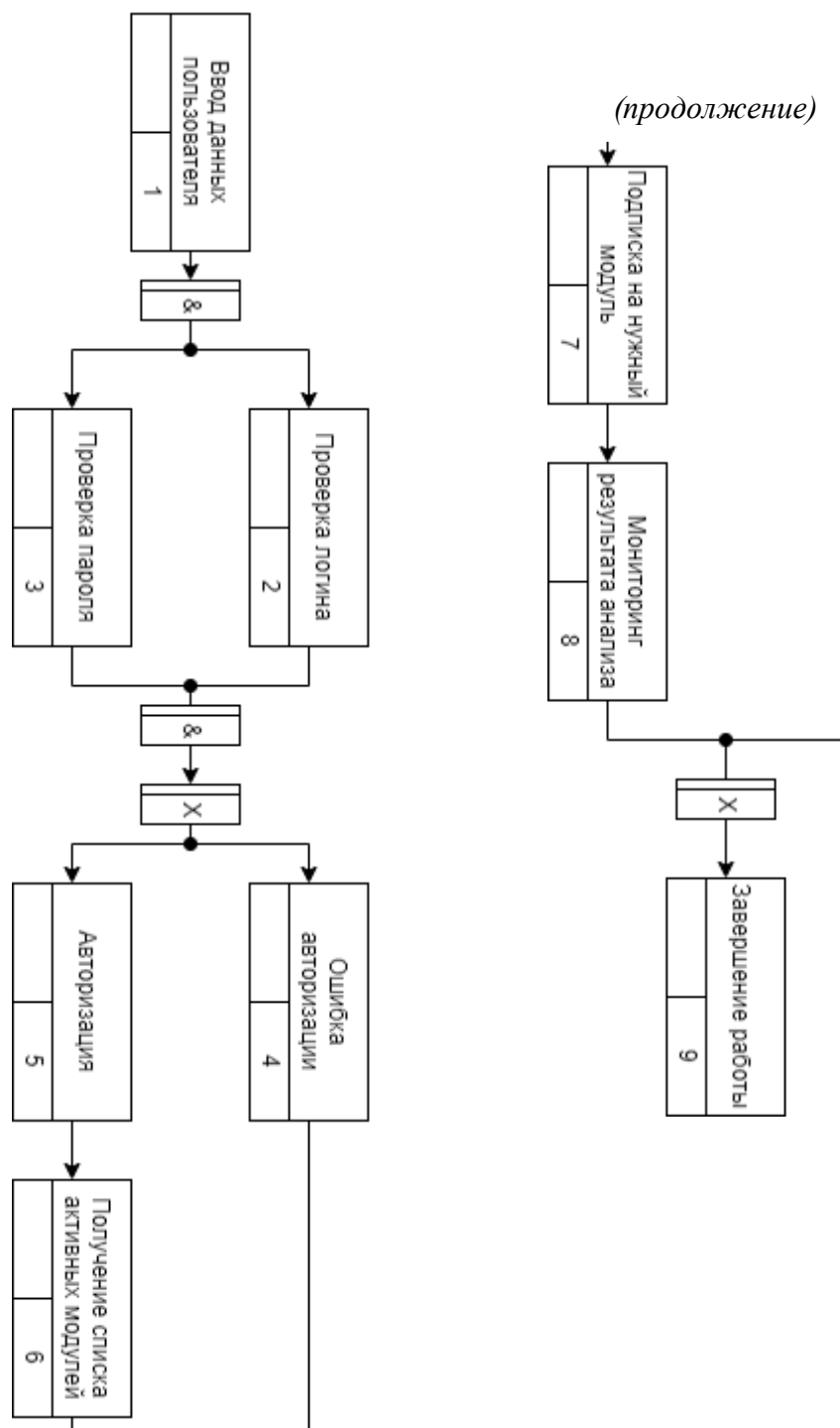


Рисунок 11 - Диаграмма IDEF3 с точки зрения пользователя.

На рисунке 11 представлена IDEF-3 диаграмма потока действий с точки зрения пользователя. Для начала работы пользователь должен

ввести данные для авторизации в системе. Это необходимо для того, чтобы начать работу с нейросетевыми модулями. Неавторизованный пользователь не сможет настроить систему и подписать кого-либо на рассылку СМС-уведомлений. Авторизация пользователя заключается в обязательной проверке логина и пароля. Первым на проверку идет логин – в базе данных проводится поиск все пользователей с таким же логином. Так как логин уникален для каждого пользователя, то результат будет либо определенный пользователь, либо пустым. При возвращении пустого результата на интерфейс просто выводится сообщение, что такой пользователь не найден. Если результат не пустой, то проводится проверка пароля. Для защиты данных, пароль хранится в неявном виде, только ключ к нему. Дешифровать пароль просто не возможно – шифровка является односторонней операцией. Однако по данному ключу можно зашифровать пароль, введенный при попытке авторизоваться. Если результат совпадает с шифром из базы данных – значит, пароль верен, и можно выдавать пользователю Beaver-токен.

После успешной авторизации пользователь получает список зарегистрированных в системе модулей нейросетевого анализа. Все они представляются в виде JSON-данных, которая система уже отображает на пользовательском интерфейсе. Каждый модуль показывает свой ip-адрес, порт и результат анализа (если совершена подписка).

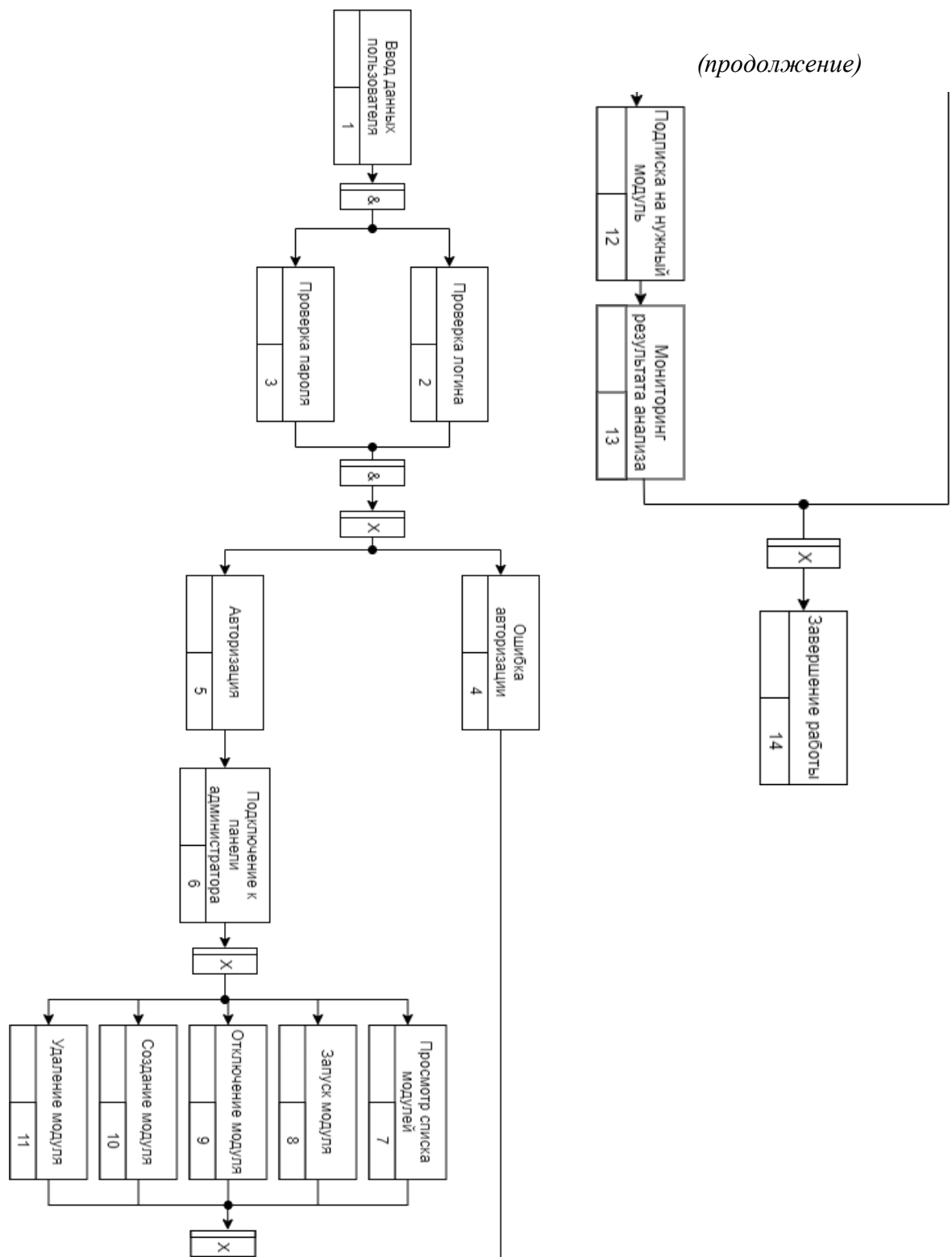


Рисунок 12 - Диаграмма IDEF3 с точки зрения администратора

С точки зрения администратора авторизация проходит аналогичным образом, за исключением того, что администратор получает сразу в распоряжение инструментарий для работы с нейросетевыми моделями. Например в системе реализован стандартный CRUD (Create, Read, Update, Delete), базовые операции которые можно делать с объектами, хранимые в базе данных. Таким образом, система реализует, наверно, одну из основных требований к разрабатываемому комплексу приложений – а именно масштабируемость. На данный момент система содержит три нейросетевых модуля – распознавание оружия, распознавание негативных эмоций, распознавание забытых/оставленных вещей. Однако, и самое главное, масштабируемость позволяет в будущем создавать и подключать новые нейросетевые модули, которые будут распознавать другие виды угроз.

3.7.3 ДИАГРАММА DFD

DFD – общепринятое сокращение от английского data flow diagrams – диаграммы потоков данных. Диаграмма DFD наглядно отображает течение информации в пределах процесса или системы. Для изображения входных и выходных данных, точек хранения информации и путей ее передвижения между источниками и пунктами доставки в таких диаграммах применяются стандартные фигуры, такие как прямоугольники и круги, а также стрелки и краткие текстовые метки. Диаграммы DFD варьируются от простейших набросков процессов (включая нарисованные вручную) до подробных многоуровневых схем с глубоким анализом способов обработки данных. диаграммы DFD применяются для анализа существующих и моделирования новых систем. В лучших традициях

визуализации данных диаграммы DFD часто наглядно «рассказывают» о процессах, которые сложно объяснить словами, и позволяют эффективно донести информацию и до «физиков», и до «лириков», то есть до всех участников организации — от разработчиков до генеральных директоров. Вот почему диаграммы DFD не утратили популярности за долгие годы существования. Однако стоит упомянуть, что хотя диаграммы DFD отлично подходят для программ и систем потоков данных, в наши дни они далеко не всегда отвечают требованиям ПО и систем, ориентированных на интерактивность, работу в реальном времени и базы данных [28].

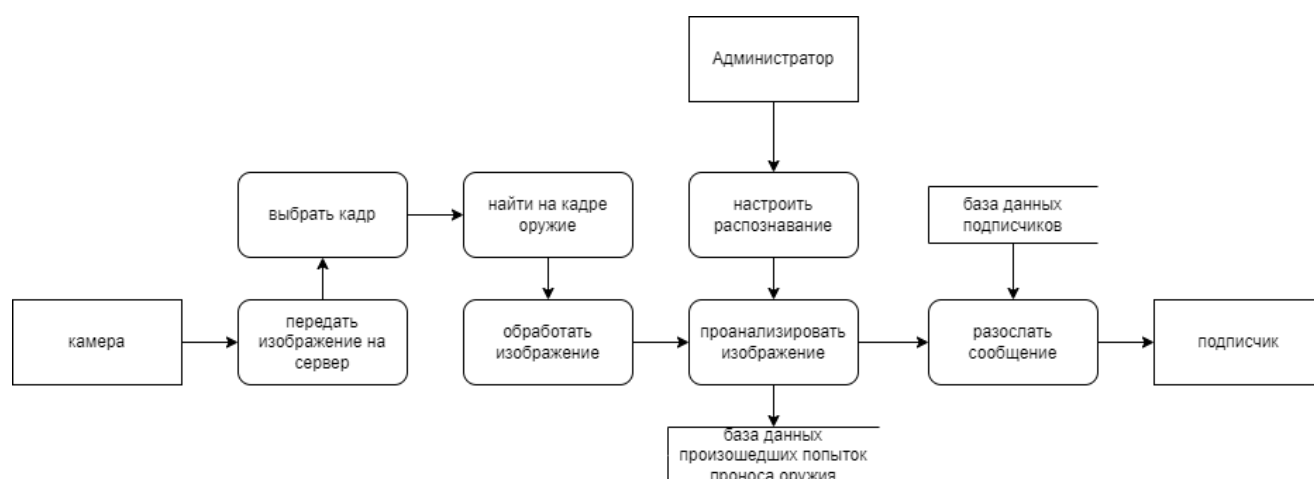


Рисунок 13 - Диаграмма DFD целиком приложения

На рисунке 13 видно общий маршрут, который проходят данные, начиная от физической камеры, заканчивая результатом в виде цветового индикатора. На этом маршруте данные постоянно преобразуются, принимают разные формы. На первых трех блоках данные представлены в виде изображения, Но после фрагментирования, происходит анализ, результатом которого будут координаты обнаруженного оружия на кадре. Эти координаты отправляются основному подписчику – бэкенду панели администратора. Бэкенд уже понимает, что с этими данными сделать, в

зависимости от настроек приложения. Например, если у модуля оружия есть подписчик – отправляет запрос модулю оповещения, после чего подписчику приходит СМС-оповещение.

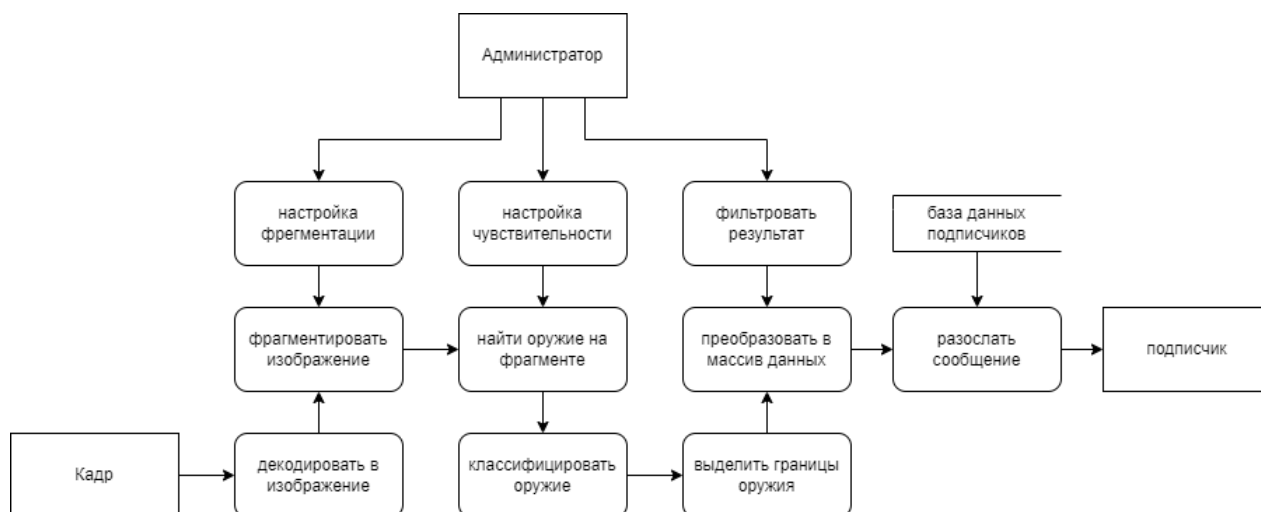


Рисунок 14 - Диаграмма DFD нейросетевого модуля

Рисунок 14 демонстрирует поток данных, проходящий внутри нейросетевого модуля. Как видно Администратор устанавливает степень фрагментации или, проще говоря, размеры фрагментов для анализа. Вторая настройка – это чувствительность нейронной сети. То есть минимально допустимый уровень уверенности анализа, который принято считать за верный. Название фильтра результатов говорит само за себя: этот фильтр указывает, какие объекты являются угрозой, а какие – нет. После анализа результат, независимо от того, нашла ли нейронная сеть что-нибудь или нет, рассылает его подписчику.

3.7.4 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ ПО МЕТОДОЛОГИИ UML

Унифицированный язык моделирования (Unified Modeling Language, UML) является графическим языком для визуализации, специфицирования, конструирования и документирования систем, в которых большая роль принадлежит программному обеспечению. С помощью UML можно разработать детальный план создаваемой системы, содержащий не только ее концептуальные элементы, такие как системные функции и бизнес-процессы, но и конкретные особенности, например классы, написанные на специальных языках программирования, схемы баз данных и программные компоненты многократного использования[29].

Язык представляет словарь и правила 17 комбинирования входящих в него слов в целях коммуникации. Язык моделирования – это язык, словарь и правила которого сосредоточены на концептуальном и физическом представлении системы. UML – стандартное средство представления «чертежей» программного обеспечения.

Моделирование необходимо для понимания системы. При этом ни одна модель не является абсолютно достаточной. Напротив, чтобы понять

большинство систем, кроме самых тривиальных, часто требуется множество взаимосвязанных моделей. В отношении программных систем это означает, что необходим язык, средствами которого можно описать архитектуру системы с различных точек зрения, причем на протяжении всего жизненного цикла ее разработки.

3.7.5 ДИАГРАММА USE CASE

Диаграмма прецедентов (англ. use case diagram, диаграмма вариантов использования) в UML — диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

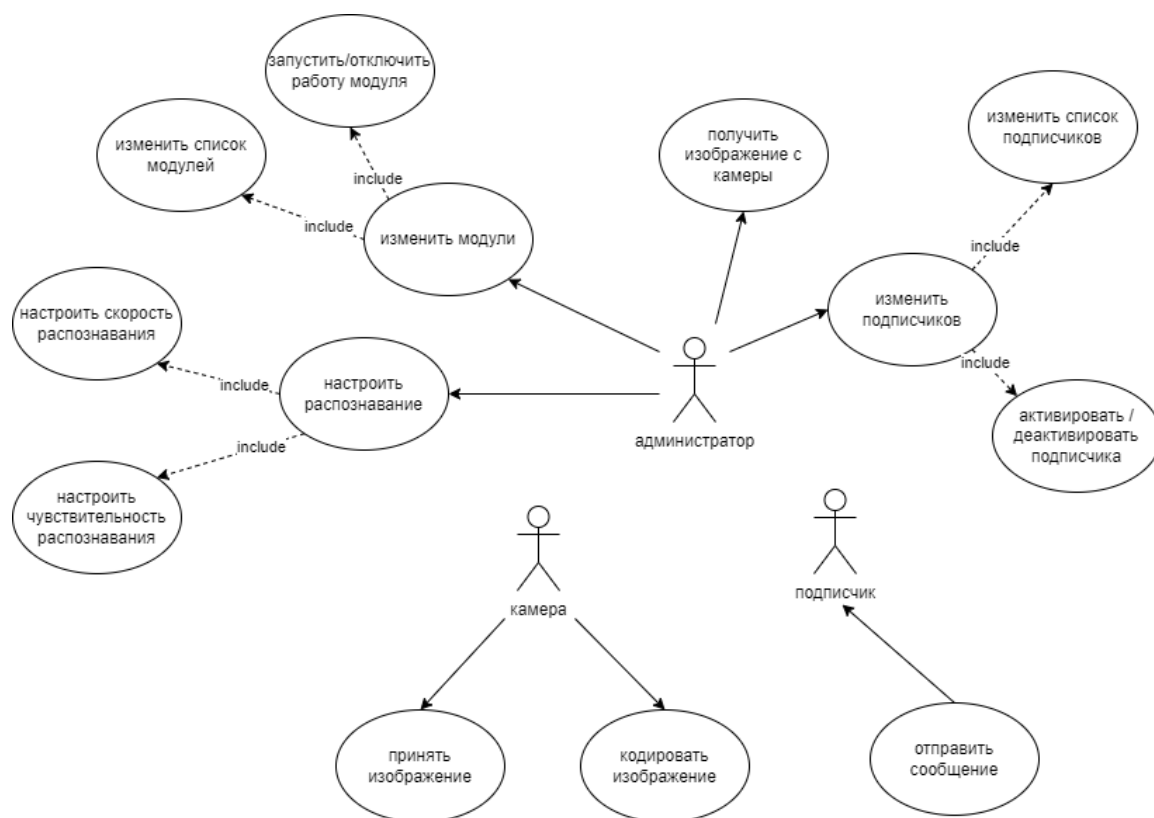


Рисунок 15 - Диаграмма прецедентов

Прецедент — возможность моделируемой системы (часть её функциональности), благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Прецедент соответствует отдельному сервису системы, определяет один из вариантов

её использования и описывает типичный способ взаимодействия пользователя с системой. Варианты использования обычно применяются для спецификации внешних требований к системе[30].

На рисунке 15 показаны основные сущности системы и их роль в общем программном модуле. Камера отвечает за захват и кодирование изображения. Подписчик отвечает только за рассылку СМС-оповещений. Больше всего задач у администратора: настройка система, управление нейросетевыми модулями, управление подписчиками и подписка на определенные модули.

3.7.6 ДИАГРАММА АКТИВНОСТИ

Диаграмма активности UML позволяет более детально визуализировать конкретный случай использования. Это поведенческая диаграмма, которая иллюстрирует поток деятельности через систему. Диаграммы активности UML также могут быть использованы для отображения потока событий в бизнес-процессе. Они могут быть использованы для изучения бизнеспроцессов с целью определения их потока и требований[31]. Активности на диаграмме “разбросаны” по беговым дорожкам, каждая из которых соответствует поведению одного из объектов (например, клиента, менеджера, веб-сервера, сервера БД и т.п.). Благодаря этому легко определить, каким из объектов выполняется каждая из активностей. Дорожка – часть области диаграммы деятельности, на которой отображаются только те активности, за которые отвечает конкретный объект. Предназначены дорожки для разбиения 19 диаграммы в соответствии с распределением ответственности за действия. Имя дорожки может означать роль или объект, которому она соответствует.

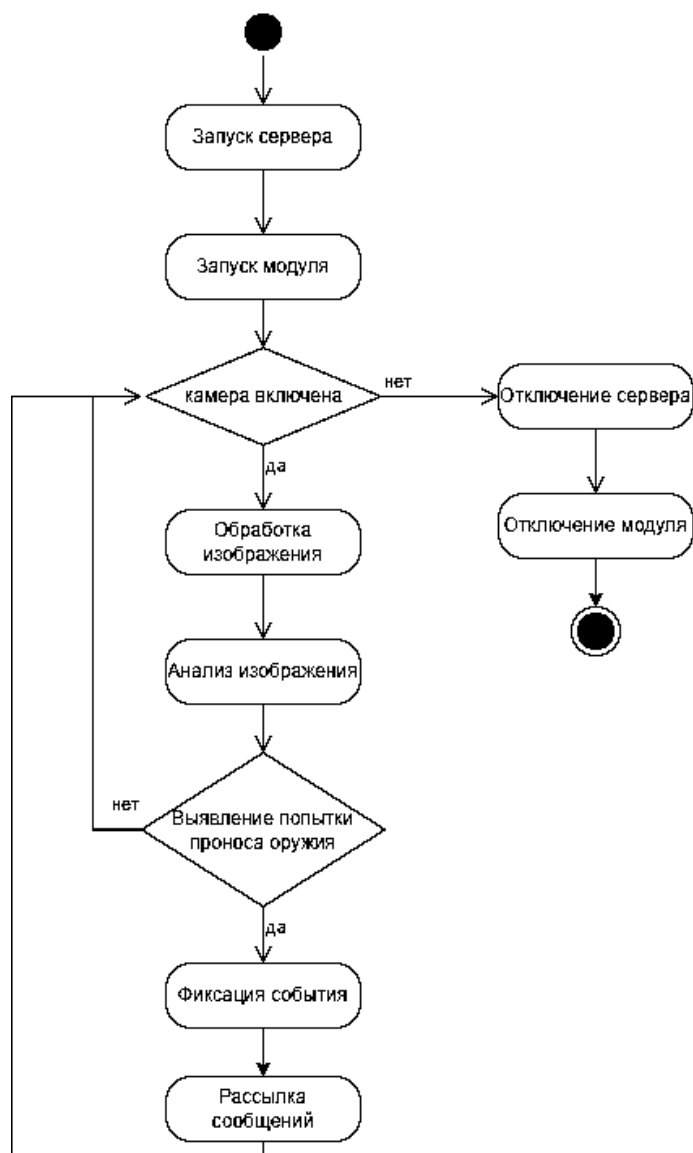


Рисунок 16 - Диаграмма активности

Активность или порядок событий – важная часть системы, так как неверный порядок может привести к сбоям в работе системы. Начнем с главного: запуск сервера. Сервер, или бэкенд, является связующим звеном между нейросетевым модулем и подписчиками. После идет запуск модуля и проверка на наличие изображения на входе системы. При наличии кадра изображение анализируется и, при выявлении угрозы, происходит рассылка сообщений.

3.7.7 ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТИ

Диаграммы последовательности (sequence diagram) являются видом диаграмм взаимодействия языка UML, которые описывают отношения объектов в различных условиях. Условия взаимодействия задаются сценарием, полученным на этапе разработки диаграмм вариантов использования[32].

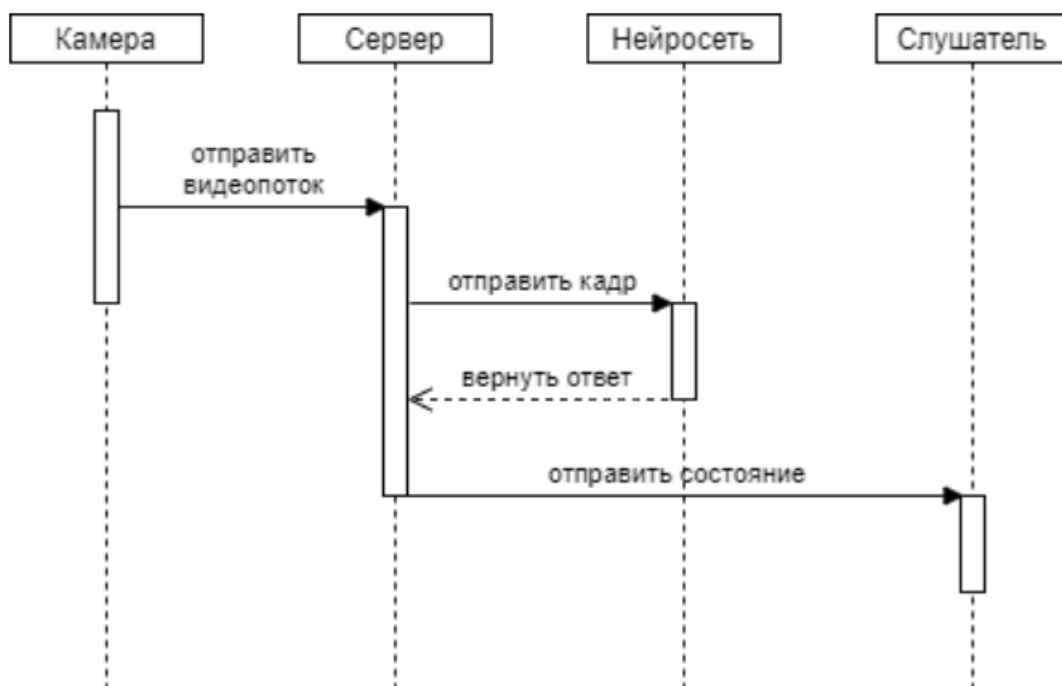


Рисунок 17 - Диаграмма последовательности

В данном случае под последовательностью имеется ввиду последовательность взаимодействий между отдельными компонентами. Пакеты данных передаются от компонента к компоненту для надежности по протоколу TCP-IP. На рисунке 17 показан порядок взаимодействий.

3.7.8 ДИАГРАММА КЛАССОВ

Диаграмма классов определяет типы классов системы и различного рода статические связи, которые существуют между ними. На диаграммах классов изображаются также атрибуты классов, операции классов и ограничения, которые накладываются на связи между классами. Вид и интерпретация диаграммы классов существенно зависит от точки зрения (уровня абстракции): классы могут представлять сущности предметной области (в процессе анализа) или элементы программной системы (в процессах проектирования и реализации)[33].

Основными элементами являются классы и связи между ними. Классы характеризуются при помощи атрибутов и операций.

Атрибуты описывают свойства объектов класса. Большинство объектов в классе получают свою индивидуальность из-за различий в их атрибутах и взаимосвязи с другими объектами. Однако, возможны объекты с идентичными значениями атрибутов и взаимосвязей. Т.е. индивидуальность объектов определяется самим фактом их существования, а не различиями в их свойствах. Имя атрибута должно быть уникально в пределах класса. За именем атрибута может следовать его тип и значение по умолчанию.

Операция есть функция или преобразование. Операция может иметь параметры и возвращать значения.

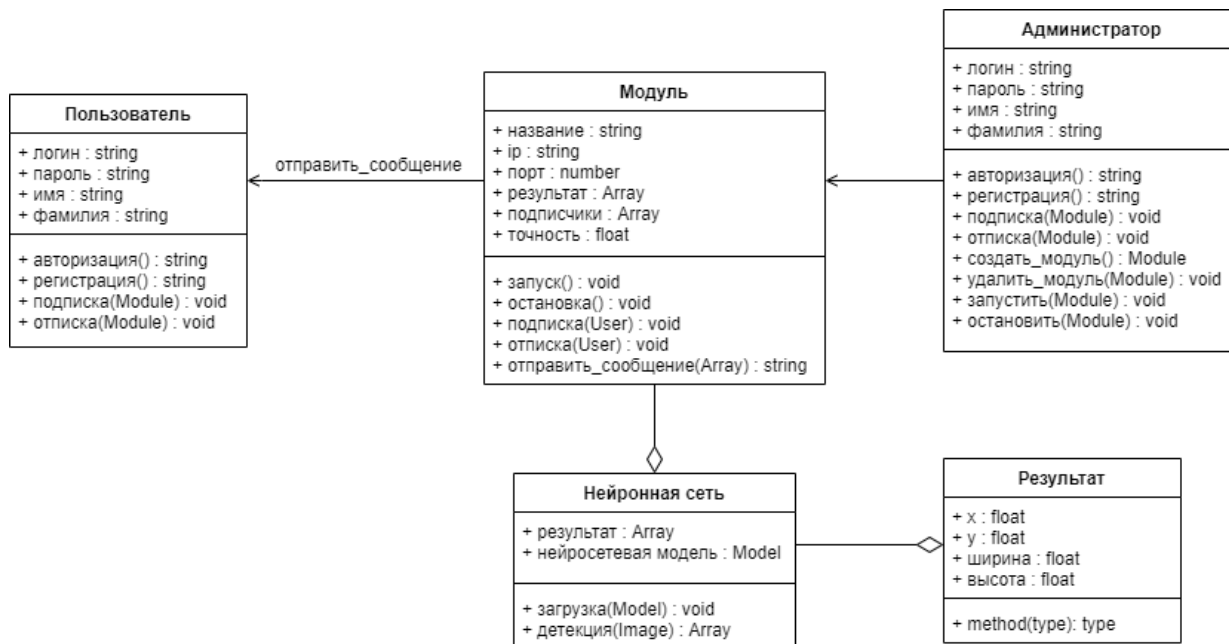


Рисунок 18 - Диаграмма классов

На рисунке 18 представлена диаграмма классов, или проще говоря, схема сущностей внутри системы, и их связи между собой. Каждая сущность описывается рядом параметров (или полей) класса и некоторыми функциями, которые эти сущности могут выполнять. Зависимости на схеме показаны стрелочками. Если взаимодействие происходит только за счёт определенного действия – оно подписано сверху.

3.7.9 ДИАГРАММА ПАКЕТОВ

Пакеты предоставляют механизм группировки взаимосвязанных элементов UML и ограничения видимости их имен. Например, все элементы, связанные с нейронными сетями, можно разместить в пакете с именем Neuroweb. Диаграммы пакетов обеспечивают отличную возможность визуального представления зависимостей между частями

вашей системы и часто используются для диагностики или определения порядка компиляции[34].

В пакетах могут группироваться практически любые элементы UML (в том числе и сами пакеты). Каждый пакет обладает именем, уточняющим область видимости каждого элемента пакета.

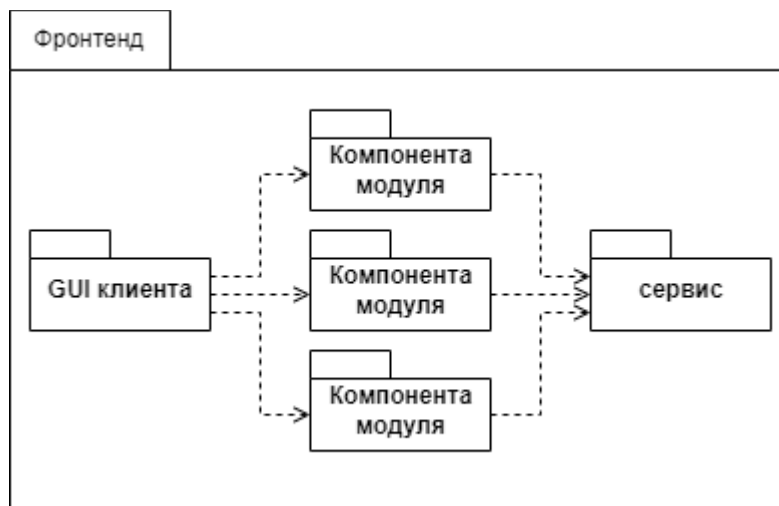


Рисунок 19 - Диаграмма пакетов фронтенда

На рисунке 19 представлена диаграмма пакетов фронтенд-сервиса разрабатываемой системы. Данная схема состоит из трех уровней: Graphic User Interface (GUI) или проще пользовательский интерфейс. Этот уровень отвечает за визуальное отображение фронтенда. Уровень компонентов модулей содержит состояние фронтенда. В основном это сторя, в которой хранятся данные непосредственно запущенного сеанса и кэши данных. Это нужно для того, чтобы не обращаться к серверу за данными постоянно, что значительно снизило бы производительность и отзывчивость системы. Третий уровень – это сервис, который участвует во взаимодействии фронтенда с бэкендом.

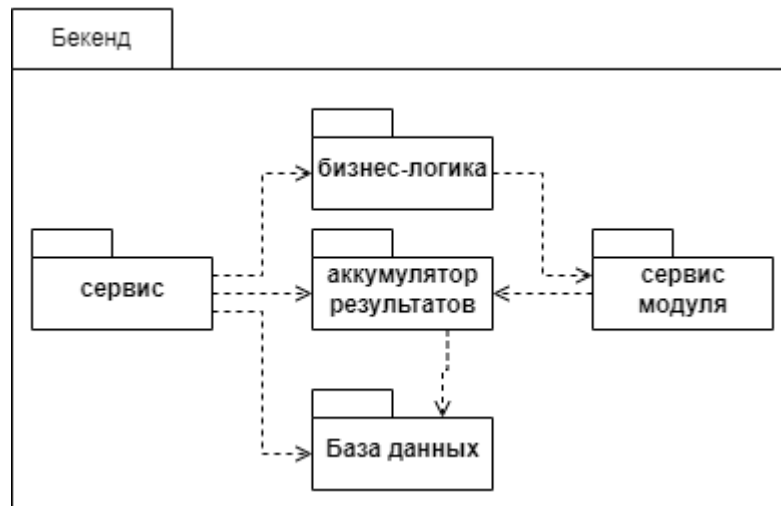


Рисунок 20 - Диаграмма пакетов бэкенда

Диаграмма пакетов бэкенда чуть посложнее предыдущей, но все так же состоит из трех уровней (рисунок 20). Сервис состоит из арі взаимодействий, который указывает, на какой запрос отвечает тот или иной функционал системы. Сервис модуля – это сервис, отвечающий за взаимодействие с нейросетевым модулем. Бизнес логика определяет логику работы приложения, функционал, лежащий под запросами. Аккумулятор результатов – это кэш данных, который хранится непосредственно в переменных кода, для ускорения процессов. Проще говоря, это область оперативных данных, которые меняются в режиме реального времени. База данных отвечает за долговременное хранение данных.

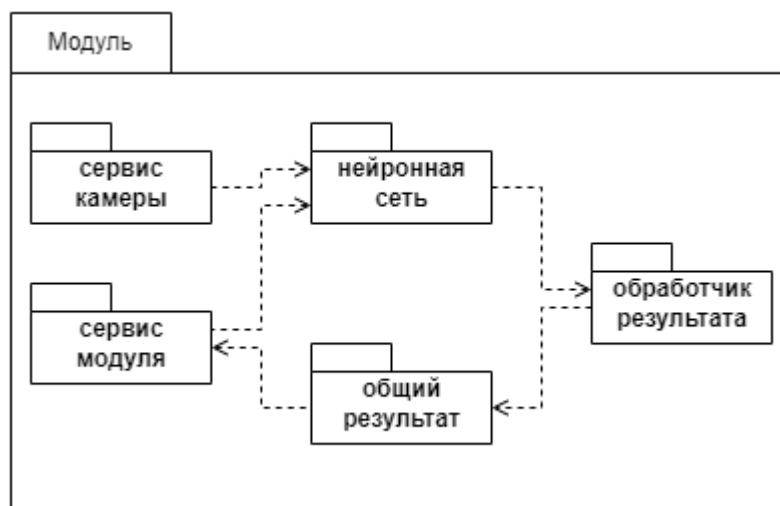


Рисунок 21 - Диаграмма пакетов нейросетевого модуля

На рисунке 21 представлена диаграмма пакетов нейросетевого модуля, которая состоит из сервиса камеры, сервиса модуля, нейронной сети и обработчика результатов. Сервис камеры отвечает за взаимодействие модуля с камерой или модулем захвата. Он принимает в систему закодированный кадр. Далее кадр принимает нейронная сеть, которая анализирует его и выдает результат в обработчик. Обработчик применяет все настройки и фильтры, установленные администратором.

В ходе проектирования системы в качестве вывода была разработана структура и проработан функционал всех модулей системы. Система должна содержать в себе несколько модулей, что должно позволить системе сохранить свое устойчивое состояние, так как выход из строя одного модуля не должен повлиять на работоспособность всей системы. Таким образом, в ходе проектирования, была разработана оптимальная схема для программного комплекса.

4 ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ

4.1 КДО-1 DOMINATION

Комплекс детектирования оружия "КДО-1" помогает оператору видеонаблюдения вовремя отреагировать на появление в зоне видимости камеры человека с оружием. Это готовое решение для внедрения в любую систему видеонаблюдения на одну входную группу[35].

В комплекс входит модуль видеоаналитики «Детектор оружия», работающий в полуавтоматическом режиме. Это значит, что модуль автоматически находит на изображении с камеры объекты, похожие на оружие, и уведомляет об этом оператора. Оператор обрабатывает полученное уведомление – подтверждает или отклоняет событие – и, если зафиксировано оружие, предпринимает соответствующие меры.



Рисунок 22 - Скришнот КДО-1

Комплекс должен быть установлен на входе в помещение: в тамбуре, коридоре, проходной зоне. Для эффективной работы комплекса в комплекте поставляются две видеокамеры. Их монтируют внутри помещения с двух сторон от входа под углом 45° на расстоянии 2-4 метра до входа. Охраннику устанавливают компьютер и видеосервер для записи и хранения данных. Информация с видеокамер передается на компьютер по сети Ethernet. В случае обнаружения тревожного события система выводит на экран монитора уведомление и сопровождает его звуковым сигналом. При необходимости можно дополнительно настроить отправку уведомления на электронную почту или в мессенджер.

Установить комплекс может любой технический специалист, имеющий опыт в монтаже видеокамер. Для подключения камер к комплексу достаточно следовать приложенной инструкции.

4.2 НЕЙРОСЕТЬ ОТ СПЕЦЛАБ

Нейросеть от Спецлаб способна распознавать угрожающие людям предметы, в том числе различные типы, как холодного, так и огнестрельного оружия. Это не гаджет, а профессионально обученная нейронная сеть, которая способна работать в условиях толпы и плохого освещения[36].

Т.к. задача выявления таких опасных предметов актуальна не только для гражданских служб, в Спецлабе провели специальную адаптацию военной нейросети для небольших электронных приборов мобильного плана. Она способна моментально сигнализировать о появлении в зоне

видимости таких видов оружия как: пистолеты, винтовки, автоматические винтовки, автоматы, пулеметы, гранатометы, ножи и кинжалы.



Рисунок 23 - Демонстрация работы нейросетевой технологии от Спецлаб

Конечно, пока нейронная сеть не может отличать боевое оружие от игрушки, но мы имеем дело не с тупыми датчиками, система видеонаблюдения Спецлаб передает оператору наглядный видеоролик момента обнаружения криминального средства, и человек имеет возможность сам уточнить назначение цели.

4.3 GUNCHECK

Программа “GunCheck” позволяет мгновенно распознавать вооруженных людей, как в потоке людей в городских условиях, так и на открытой местности при помощи камер видеонаблюдения, которые

сегодня используются повсеместно. “GunCheck” – распознавание оружия на качественно новом уровне.

После определения угрозы, программа обратит внимание оператора на нужную камеру и выполнит запрограммированные действия[37].



Рисунок 24 - Скриншот GunCheck

На бюджетном GPU Mobile RTX 3070 с 8 GB видеопамяти, поддерживается до 4 камер с высоким FPS, с возможностью распознавания оружия на дистанциях до 30 метров. Если снизить расстояние до 15 метров, то обеспечивается работа уже 16 камер с аналогичным числом кадров в секунду. При использовании камер с кодеком H.265 можно поднять КПД ещё выше, за счёт пропуска обработки кадров, в которых нет изменения изображения.

Возможность работы в серверном режиме и подбор оборудования под ваши нужды и возможности, в том числе использование имеющегося.

4.4 АРСЕНАЛ+

«Антиколумбайн» разработала компания «Арсенал+». Система, созданная силами разработчиков компании, тестируется на нескольких социально-значимых объектах Тюмени, рассказали в компании[38].

Система работает на предобученной нейросети. Во многом от качества ее обучения зависит точность распознавания, говорят в «Арсенал+», но точность системы сравнима с человеческим глазом. Она также может определить материал, из которого сделан идентифицированный предмет.

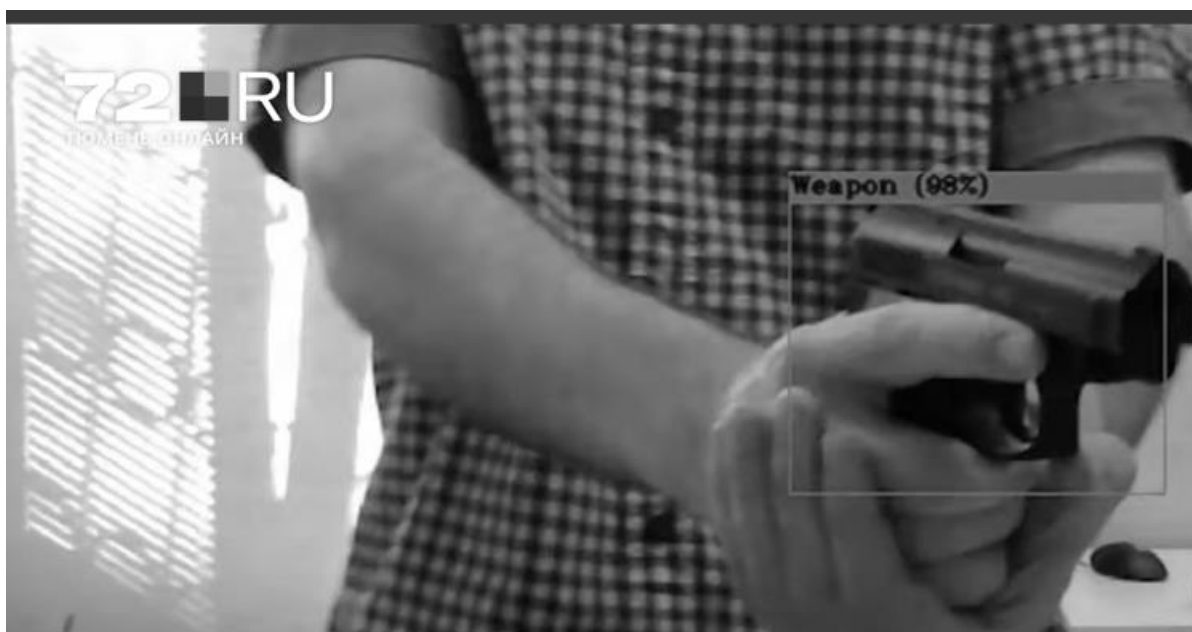


Рисунок 25 - Демонстрация работы программы «Арсенал+»

«На данный момент для работы требуется компьютер с видеокартой. Большая мощность не нужна. Одна из самых простых видеокарт может выдавать скорость около пяти кадров в секунду, этого достаточно, чтобы

определить оружие на картинке», – отметили в «Арсенал+», добавив, что камера должна быть цифровой.

Компания находится в стадии поиска партнеров для внедрения пилота. Для его участников софт и лицензии предоставят бесплатно. Для пилота системы на сервер видеонаблюдения компания установит тестируемое программное обеспечение, подключенное к одному или нескольким видеопотокам. Собранные данные послужат усилением выборки.

4.5 СРАВНИТЕЛЬНЫЙ АНАЛИЗ СУЩЕСТВУЮЩЕГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Сравнительный анализ систем приводится в Таблице 2. Целью данного анализа является выявление преимуществ и недостатков аналогичных систем.

В Таблице 2 приведены данные о наличии у рассмотренных аналогичных систем следующих возможностей:

- Масштабируемость систем;
- Мгновенное оповещение через СМС;
- Возможность удалённого управления системой;
- Возможность добавления новых модулей.

Были выделены основные параметры разрабатываемой нейросетевой системы, такие как: масштабируемость, система должна иметь возможность

разворачиваться на нескольких физических устройствах; отказоустойчивость – система должна продолжать работать даже при выходе из строя или технических работ на одном из модулей.

Таблица 2 - Сравнительная характеристика систем

	КДО-1	спецлаб	guncheck	антиколумбайн	Разрабатываемая СУО
Масштабирование	-	-	-	-	+
Возможность добавления новых модулей	-	-	-	-	+
Оповещение подписчиков	+	-	+	+	+
Возможность удаленного управления	-	+	-	-	+
Возможность частичного распознавания	-	-	-	-	+

В результате проведенного исследования аналогов разрабатываемой системы собраны следующие выводы:

- Некоторые рассмотренные системы не имеют возможности удаленного управления, подписки/отписки от рассылки;
- Во многих системах отсутствует возможность СМС-оповещения;
- Многие системы распознают только один вид угроз – огнестрельное оружие, однако угроза может изначально проявляться во многих видах, таких как забытые вещи и эмоциональное состояние;

- Так же системы реагируют только, когда огнестрельное оружие полностью покажется в кадре, однако нередко его проносят в сумках или под одеждой из-за чего оно может появляться в кадре частично.

Принимая во внимание факторы, описанные выше, было решено разработать собственную нейросетевую систему, которая будет соответствовать вышеперечисленным требованиям.

Проведённый анализ представленных решений позволяет сделать вывод о том, что необходимо разработать программный интеллектуальный комплекса контроля безопасности и модуль обнаружения оружия.

Функционала разрабатываемого программного комплекса для анализа и контроля безопасности будет достаточно для полного цикла работ по автоматизации анализа и контроля безопасности. Одной из основных функций является автоматический анализ видеопотока на предмет угроз. Также масштабируемость системы, которая проявляется в возможности добавления новых нейросетевых модулей, которые смогут анализировать на предмет угроз. Кроме того, классификация типа угроз позволяет оценить степень угрозы. Все описанные функции предоставляют возможность для полноценной работы по оценке и анализу видеопотока.

5 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ НЕЙРОСЕТЕВОЙ СИСТЕМЫ

5.1 ПЕРЕЧЕНЬ КОМПОНЕНТОВ СИСТЕМЫ, ИХ НАЗНАЧЕНИЕ И ОСНОВНЫЕ ХАРАКТЕРИСТИКИ

- Нейросетевой модуль – модуль, предназначенный для нейросетевого анализа видеопотока, выявлении угроз в кадре и оповещении подписчиков;
- Бекенд – обеспечивает прием результатов анализа и рассылку СМС оповещений подписавшимся клиентам, так же принимает запрос с панели администратора и обеспечивает связь с базой данных;
- Панель администратора – предназначена для администрирования системы в целом, добавления/редактирования/удаления нейросетевых модулей, созданию подписчиков и организации подписки;
- Подсистема оповещения – удаленный интернет-ресурс, который по пост-запросу позволяет отправить СМС подписчику;
- Подсистема захвата видеопотока.

5.2 ИСПОЛЬЗУЕМЫЕ API

API – Технология, благодаря которой разные программные обеспечения или информационные системы могут взаимодействовать друг с другом, или части одной системы.

5.3 АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

5.3.1 КЛАССЫ И ИНТЕРФЕЙСЫ

Во время разработки нейросетевой системы были реализованы следующие компоненты (программные классы ангуляр) и интерфейсы:

- `NeyroComponent` – компонента или класс, отвечающий за состояние списка нейросетевых модулей, или их интерфейс-представление на панели администратора;
- `NeyroModyleComponent` – компонента-класс, отвечающая за состояние отдельного интерфейс-представление отдельного модуля;
- `ListListenersComponent` – класс для работы со списком подписчиков;
- `ListenerComponent` – класс подписчика на определенный модуль;
- `PopurComponent` – класс элемента интерфейса, так называемого «попап»;
- `NeyroService` – класс-сервис, отвечающий за отправку запросов от `NeyroComponent` на бекенд и принятия результатов запросов;
- `NeyroModuleService` – класс-сервис, отвечающий за отправку запросов от `NeyroModuleComponent` на бекенд и принятия результатов запросов.

5.3.2 СЕРВЕРНАЯ ЧАСТЬ

Серверная часть нейросетевой системы, или бекенд был разработан на Node.js, с помощью которого были разработаны следующие функции:

- Маршрутизация и обработка входящих запросов;
- Работа с БД;
- Отправка запроса на СМС-оповещение;
- Прием результатов нейросетевого анализа.

5.3.2.1 API

Серверная часть нейросетевой системы предоставляет API для работы с системой в целом.

Во время разработки серверной части были реализованы следующие точки входа API (таблица 3):

Таблица 3 - Точки входа серверной части (API)

Путь (начиная от /api/v2)	Метод	Роль	Описание
/api/auth/login	POST	-	Авторизация в системе
/api/auth/register	POST	-	Регистрация в системе
/api/auth/getByToken	POST	-	Получение токена
/api/camera/get_imgres	Post	Администратор	Устанавливает соединение с камерой
/api/neyro/message_module	Post	Модуль	API по которому модуль сообщает результат анализа

/api/neyro/getResult	Post	Администратор	Получение результата на фронтенд
/api/neyro/submit	Post	Администратор	Подписка на модуль
/api/neyro/getallmod	Post	Администратор	Получение списка модулей
/api/neyro/getlocalip	Post	Администратор	Получение IP сервера
/api/neyro/getStatus	Post	Администратор	Получение онлайн-статуса модуля
/api/neyro/addModule	Post	Администратор	Добавление модуля
/api/neyro/editModule	Post	Администратор	Изменение модуля
/api/neyro/deleteModule	Post	Администратор	Удаление модуля
/api/listeners/getAllListeners	Post	Администратор	Получение списка подписчиков
/api/listeners/addListener	Post	Администратор	Добавление подписчика
/api/listeners/ get_all_names_module_by_phone	Post	Администратор	Получение списка модулей по номеру телефона
/api/listeners/podpiska	Post	Администратор	Подписка на модуль
/api/listeners/otpiska	Post	Администратор	Отписка от модуля

5.3.2.2 ROUTES

Серверная часть постоянно принимает запросы и обязана их каким-либо образом обрабатывать. Для приема запросов, необходимо продумать грамотную маршрутизацию или роутинг. Для этого серверная часть была разбита на компоненты, каждая из которых имеет как собственный роутер, так и собственный контроллер.

Внизу представлен список роутеров серверной части:

- auth_router;
- camera_router;
- listeners_router;
- neyro_router.

5.3.2.3 CONTROLLERS

Серверная часть после принятия запросов обязана их каким-либо образом обрабатывать. Для обработки запроса определенным образом, каждому маршруту должен соответствовать определённый контроллер.

Внизу представлен список контроллеров серверной части:

- auth_controller;
- camera_controller;
- listeners_controller;
- neyro_controller.

5.3.3 НЕЙРОСЕТЕВОЙ МОДУЛЬ

Нейросетевой модуль представляет собой самостоятельное мини-приложение, которое работает в автономном режиме. Данное приложение обладает собственным интерфейсом, представленный на рисунке 26.

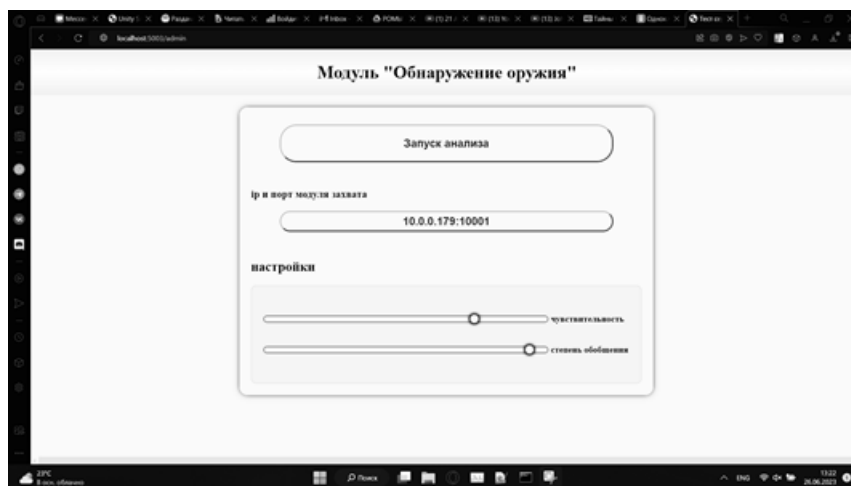


Рисунок 26 - Скриншот экрана настроек нейросетевого модуля

На рисунке видно, что мы можем указать ip-адрес модуля захвата изображения и отправить команду для начала нейросетевого анализа. Также мы можем установить базовые настройки нейронной сети, такие как точность анализа и степень обобщения. Если с точностью более-менее понятно, то вот степень обобщения требует разъяснения. Дело в том, что перед анализом кадр проходит этап фрагментации, и оружие может занимать несколько фрагментов, поэтому на одно огнестрельное оружие может распознаваться как несколько. Степень обобщения указывает степень разброса координат на изображении, которые должны распознаваться как единый результат. Модуль разработан на языке программирования Питон с применением такого фреймворка как Фласк. API данного модуля представлены в таблице 4.

Таблица 4 - Точки входа нейросетевого модуля

Путь (начиная от /api/v2)	Метод	Роль	Описание
127.0.0.1:5003/admin	GET	-	Демонстрация экрана настроек
127.0.0.1:5003/status	POST	-	Включение/выключение анализа
127.0.0.1:5003/getStatus	POST	-	Получение статуса анализа
127.0.0.1:5003/resultDetect	POST	-	Получение результата анализа
127.0.0.1:5003/subscribe	POST	-	Подписка на рассылку

5.3.3.1 НЕЙРОСЕТЕВАЯ МОДЕЛЬ

В ядре нейросетевого модуля лежит нейросетевой анализ видеопотока, который проводится нейронной сетью. Для разрабатываемой системы была разработана нейросетевая модель, которая представляет собой свёрточную нейронную сеть, разработанной при помощи фреймворка YoLo 5-й версии.

Таблица 5 - Подробное представление нейросетевой модели

	from	n	параметр	слой	аргументы
0	-1	1	3520	models.common.Conv	[3, 32, 6, 2, 2]
1	-1	1	18560	models.common.Conv	[32, 64, 3, 2]
2	-1	1	18816	models.common.C3	[64, 64, 1]
3	-1	1	73984	models.common.Conv	[64, 128, 3, 2]
4	-1	2	115712	models.common.C3	[128, 128, 2]
5	-1	1	295424	models.common.Conv	[128, 256, 3, 2]
6	-1	3	625152	models.common.C3	[256, 256, 3]
7	-1	1	1180672	models.common.Conv	[256, 512, 3, 2]
8	-1	1	1182720	models.common.C3	[512, 512, 1]
9	-1	1	656896	models.common.SPPF	[512, 512, 5]
10	-1	1	131584	models.common.Conv	[512, 256, 1, 1]
11	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']

12	[-1, 6]	1	0	models.common.Concat	[1]
13	-1	1	361984	models.common.C3	[512, 256, 1, False]
14	-1	1	33024	models.common.Conv	[256, 128, 1, 1]
15	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
16	[-1, 4]	1	0	models.common.Concat	[1]
17	-1	1	90880	models.common.C3	[256, 128, 1, False]
18	-1	1	147712	models.common.Conv	[128, 128, 3, 2]
19	[-1, 14]	1	0	models.common.Concat	[1]
20	-1	1	296448	models.common.C3	[256, 256, 1, False]
21	-1	1	590336	models.common.Conv	[256, 256, 3, 2]
22	[-1, 10]	1	0	models.common.Concat	[1]
23	-1	1	1182720	models.common.C3	[512, 512, 1, False]
24	[17, 20, 23]	1	75516	models.yolo.Detect	[23]

Модель свёрточной нейронной сети содержит 24 слоя. Первым слоем идет сенсорный слой. Этот слой необходим для преобразования изображения в массив чисел, с которым будет работать нейросетевая модель. В качестве активационной функции была выбрана одна из самых простых в понимании и реализации функция – функция линейного порога.

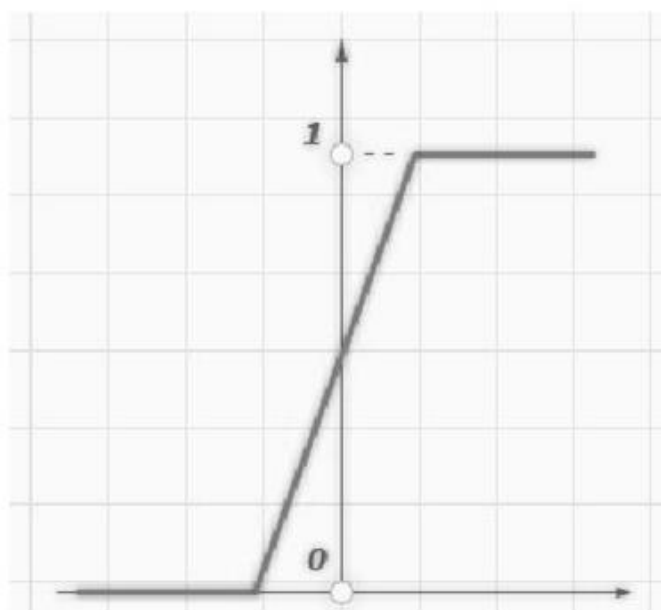


Рисунок 27 - Графическое представление функции линейного порога

На рисунке 27 представлена схема или графическое представление функции линейного порога. А вот формула функции представлена ниже, формула 2.

$$f(x) = \begin{cases} 0, x < 0; \\ \frac{x}{w_0}, w_0 > x \geq 0; \\ 1, x \geq w_0 \end{cases} \quad (2)$$

Данная функция повращает входные данные без «искажений», при условии, если эти данные лежат в диапазоне от 0 или 1... иначе функция будет просто возвращать значение верхней или нижней границы.

Так же неросетевая модель содержит 9 сверточных (models.common.Conv или Convolutional Layer). Сверточная сеть предполагает на входе особую пространственную структуру. В частности, это предполагает, что входы, которые расположены близко друг к другу в пространстве, семантически связаны. Это предположение имеет смысл для изображений, так как пиксели, расположенные близко друг к другу, скорее всего семантически связаны. Эта концепция проиллюстрирована на рисунке 28.

В качестве вышеупомянутой активационной функции на сверточных слоях была выбрана сигмоида. Сигмоида (рисунок 28) представляет собой нелинейную функцию, имеющую форму буквы «S», которую часто используют для «сглаживания» значений некоторой величины. Она помогает нейронным сетям «обобщать» результат выборки.

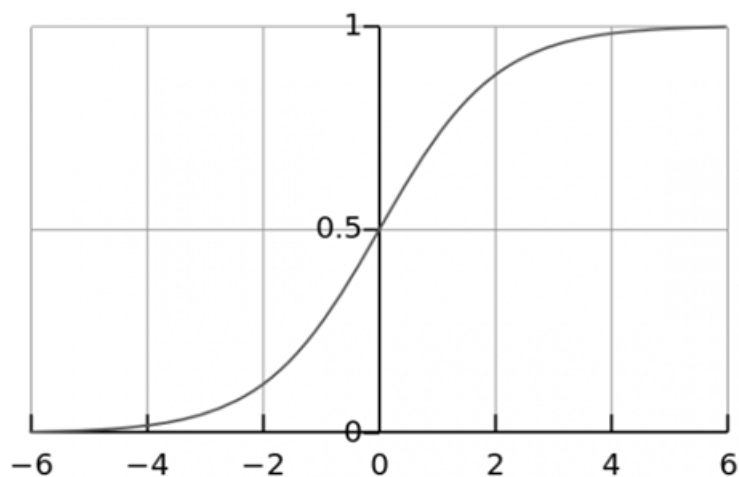


Рисунок 28 - График сигмоидальной активационной функции

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

Точно так же, как полносвязные слои преобразуют списки в списки, сверточные слои (рисунок 29) преобразуют изображения в образы. В результате сверточные слои можно использовать для выполнения сложных преобразование изображений, например применение художественных фильтров к изображениям в фотоприложениях.

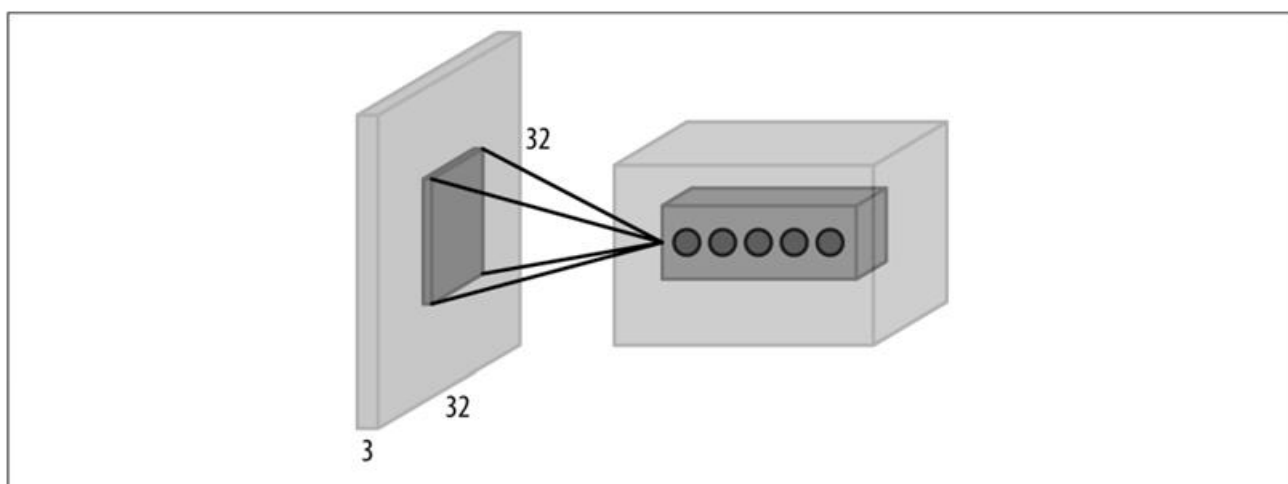


Рисунок 29 - Сверточный слой

Распознавание и отслеживание проходит по следующему алгоритму: все изображение разбивается с помощью сетки на ячейки размером $S \times S$. После этого каждая ячейка отвечает за предсказание нескольких вещей.

Во-первых, каждая ячейка отвечает за предсказание нескольких содержащих рамок и показателя уверенности (confidence) для каждой из них – другими словами, это вероятность того, что данная рамка содержит объект. Если в какой-то ячейке сетки объектов нет, то очень важно, чтобы confidence для этой ячейки был очень малым.

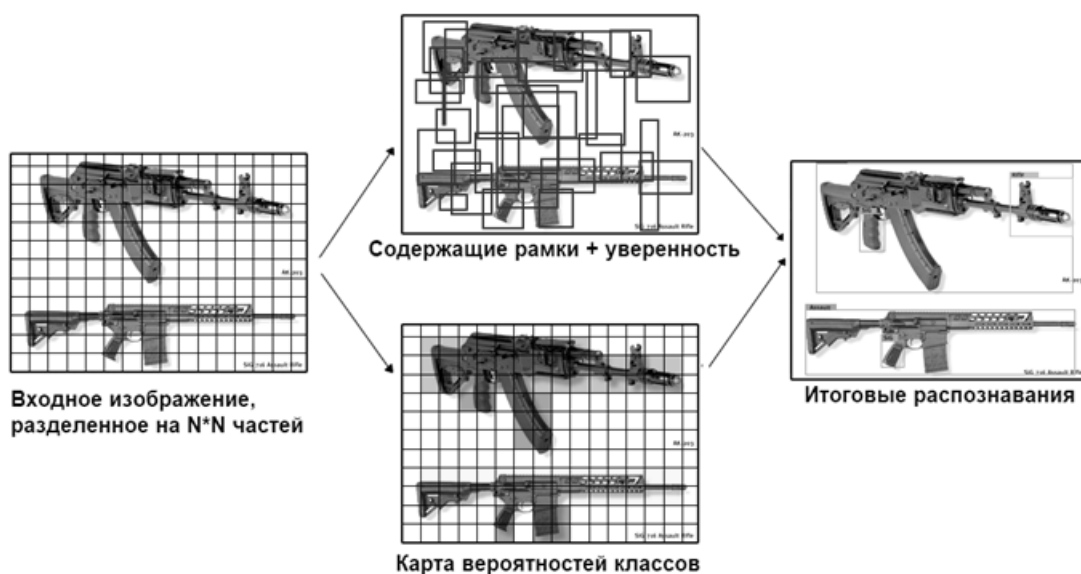


Рисунок 30 - Принцип разбиения изображения на фрагменты и поиск объектов на изображении

Во-вторых, каждая ячейка отвечает за предсказание вероятностей классов. Это не значит, что какая-то ячейка содержит какой-то объект, это всего лишь вероятность (рисунок 30). Таким образом, если ячейка сети предсказывает автомобиль, это не значит, что он там есть, но это значит, что если там есть какой-то объект, то это автомат[39].

На выходном слое количество нейронов совпадает с количеством классов (всего 23 штуки). Последней активационной функцией была ReLu.

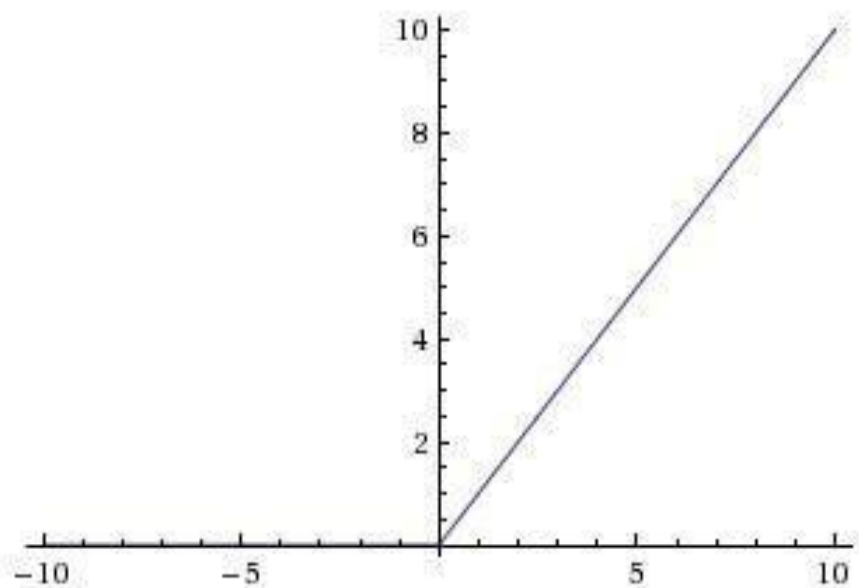


Рисунок 31 - Графическое представление активационной функции ReLu

$$f(x) = \max(0, x) \quad (4)$$

Это было сделано для того, чтоб результат анализа возвращался без зашумлений, но не выходил за нижнюю границу (0).

5.3.3.2 ПОДГОТОВКА ДАТАСЕТА

Для обучения нейросетевой модели необходимо для начала собрать датасет – то есть набор изображений, на котором нужно обучать нейронную сеть. Для этого был задействован ресурс <https://www.kaggle.com> или просто Кегл. В данном ресурсе был найден

самый объемный датасет с набором разнообразного огнестрельного и холодного оружия.

Однако данный датасет был предназначен только для классификации, в то время как для выделения объекта в кадре необходимо изображение преобразовать в дуэт картинка-координаты определенного класса. Для этого с помощью программы `labelImg` на картинке были выделены нужные объекты и установлены соответствующие им классы.

Разобравшись с программным обеспечением, был составлен список соответствующих классов. Учитывая, что оружие может показаться частично в кадре — было принято решение расширить список классов. Грубо говоря, каждый класс содержал в себе еще несколько подклассов — это дуло с мушкой и рукоятка оружия, так как именно эти части оружия чаще всего показываются в кадре.

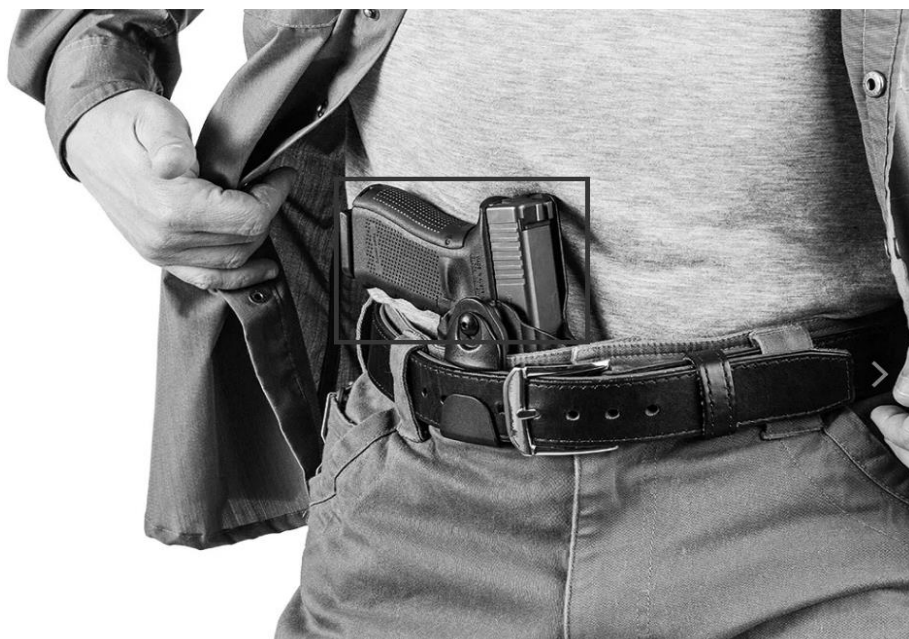


Рисунок 32 - Частичное появление оружия в кадре

На рисунке 32 показан пример того, как оружие может показаться частично в кадре. Легкое огнестрельное оружие часто переносится под одеждой, но рукоятка всегда остается открытой для того, чтобы оружие можно было легко и быстро достать. Система должна распознавать рукоятку и своевременно на это отреагировать.



Рисунок 33 - Второй пример частичного появления оружия в кадре

На рисунках 32-33 представлены примеры того, как оружие появляется в кадре частично. На первой картинке человек имел неосторожность полностью скрыть пистолет, из-за чего показалась рукоять, которая выделена была в отдельный класс. Во втором случае автомат Калашникова был не полностью спрятан в спортивной сумке, благодаря чему нейронная сеть может распознать узнаваемый образ части

автомата, а именно мушку с газовой трубкой. Подобных узнаваемых образов можно выделить и на других автоматах.

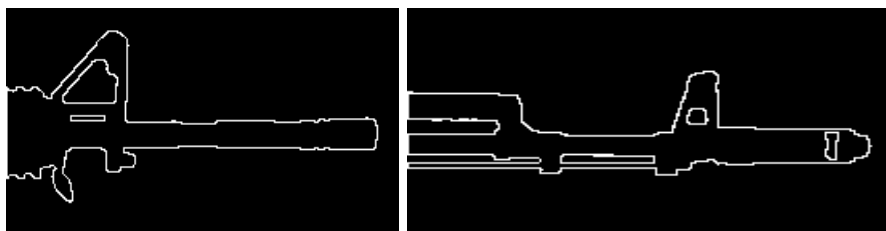


Рисунок 34 - Узнаваемые образы частей огнестрельного оружия

Мушка является одной из самых часто выпирающих частей автоматов, поэтому за узнаваемый образ был взят именно этот класс. Было выделено два дополнительных класса, каждый из которых был обучен узнавать в кадре подобную часть огнестрельного оружия. На рисунке 34 продемонстрирована мушка автоматов Калашникова (слева) и одной из самых распространённых американских штурмовых винтовок М4А1 (справа).

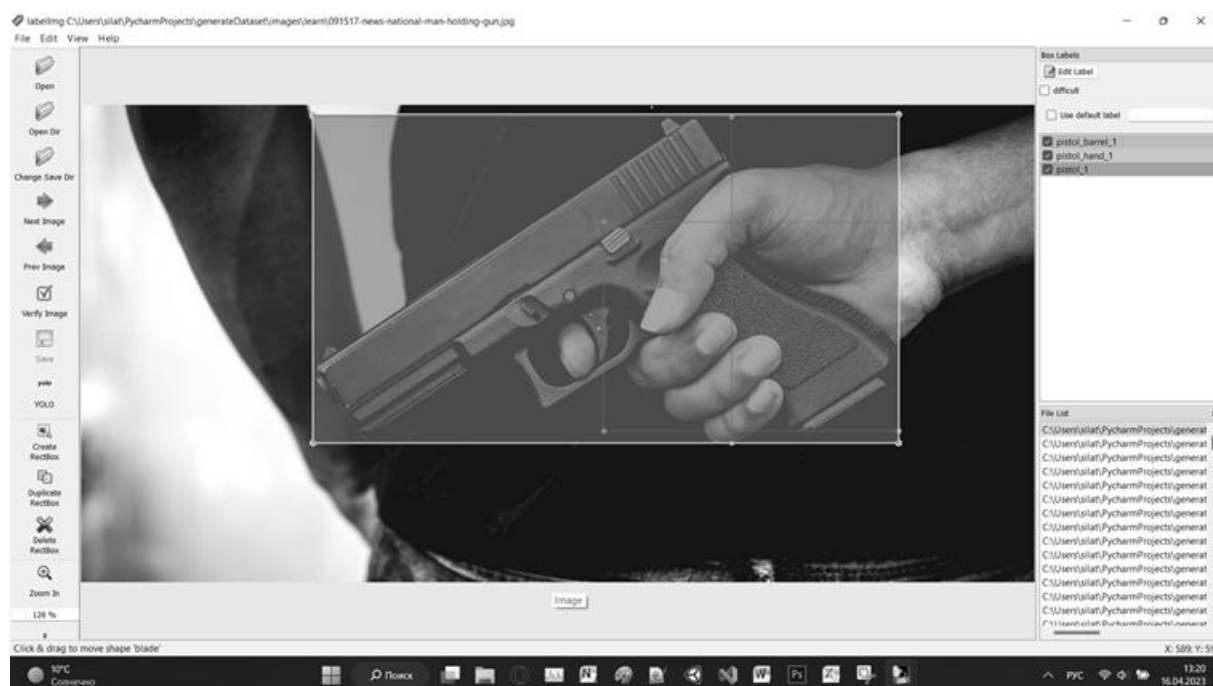


Рисунок 35 - Пример выделения объекта в программе labelImg

Программа labelImg была специально создана для создания датасета для нейронных сетей Yolo. Она помогает указать координаты и размеры объекта в кадре, которые позже сохраняются в виде координат. Координаты переводятся из стандартных целочисленных значений количества пикселей в процентное соотношение относительно ширины (для координаты x) и относительно высоты (для координаты y) картинки.

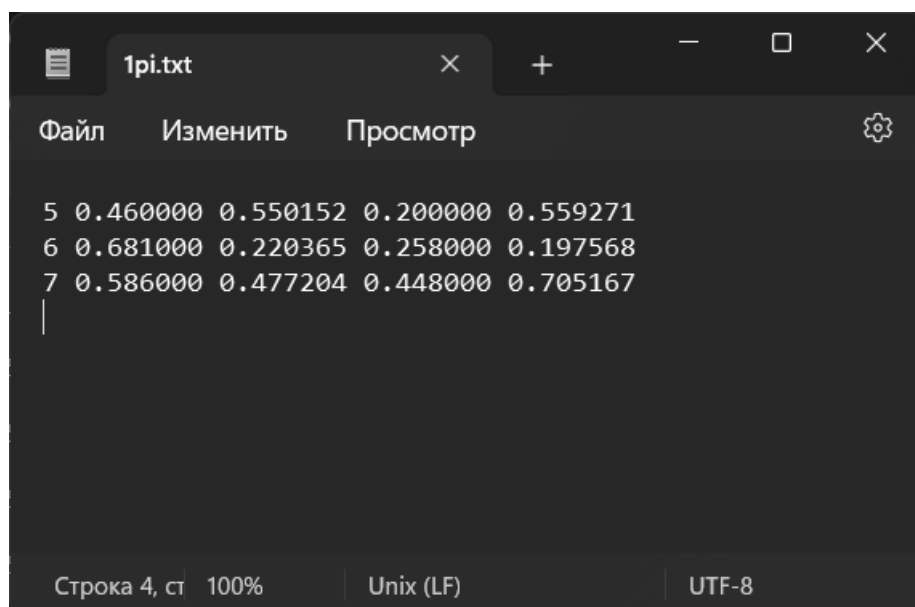


Рисунок 36 - Пример сохраненных координат каждого класса для отдельной картинки.

Для правильного установления координат на изображении и дальнейшей возможности работы с помощью нежной нейронной сети необходимо установить режим YoLo. В папке с изображениями сформируются текстовые файлы, с содержащимися в них координатах класса. На рисунке 36 видно один из таких файлов, который содержит индексы трех классов и их координаты на картинке. Стоит отметить, что нередко картинки преобразуются в квадратные изображения для работы нейронной сети, потому все координаты показаны строго в диапазоне от 0 до 1.

После чего, для увеличения датасета, каждая картинка была подвергнута вращению на 90, 180 и 270 градусов, отзеркалены по горизонтали и вертикали. Над поворотом на 45, 135, 225 и 315 градусов был проработан алгоритм описывающего прямоугольника.

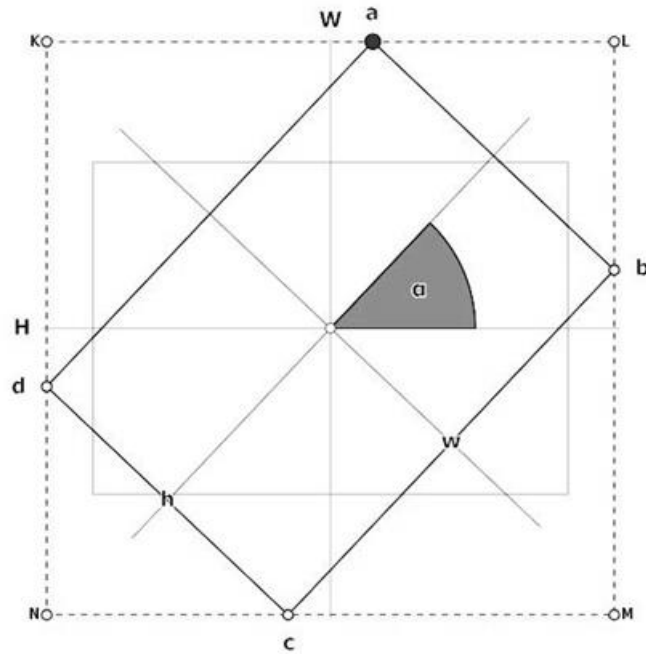


Рисунок 37 - Описывающий прямоугольник.

Дело в том, что нейронная сеть способна выделять объект в горизонтально расположенный прямоугольник (Rectangle), но изображения в 45 градусов она тоже обязана уметь распознавать. При включении в датасет изображения в 45 градусов был разработан алгоритм, который пересчитывал координаты выделяющего ректенгла. Формула этого алгоритма приведена ниже.

$$\begin{aligned} x &= ((old_x - c_x) * \cos(rad(45)) - (old_y - c_y) * \sin(rad(45))) + C_x \\ y &= ((old_x - c_x) * \sin(rad(45)) - (old_y - c_y) * \cos(rad(45))) + C_y \end{aligned} \quad (5)$$

Где c_x и c_y – это координаты очки, относительно которой велось вращение ректенгла, а old_x и old_y – это старые координаты точки.

Сопоставив все результаты, получился новый ректенгл, которые находился в нужном месте на изображении.



Рисунок 38 - Результат вращения на 45 градусов с новым выделением

На рисунке 38 представлен пример уже ранее обработанного изображения, после поворота его на 45 градусов, изменяя координаты границ ректенгла при помощи формулы 5. Это необходимо для того, чтобы увеличить размер датасета различными вариациями одной картинки.

5.4 ФУНКЦИОНАЛЬНЫЕ ПРОЦЕССЫ СИСТЕМЫ

Подсистема обнаружения фактов проноса оружия – это программное средство для мониторинга и обнаружения попытки проноса оружия на территорию университета. В случае с машинным обучением подсистема автоматизирует процесс мониторинга и обнаружения попыток проноса

оружия, а так же обеспечит своевременное оповещение. Ниже представлены функциональные возможности системы.

5.4.1 ГОСТЬ

Неавторизованный пользователь имеет следующие возможности:

- авторизация (рисунок 39);
- регистрация (рисунок 40).

Авторизация в системе осуществляется при помощи выдачи так называемого Bearer-token-a авторизации пользователю, который будет работать на протяжении определенного промежутка времени, после чего станет недействительным, и его надо будет обновить.

Список подписчиков

Авторизация Регистрация

abc

войти очистить

Created by A.B.C. © 2019
Антониев В.С.

Рисунок 39 - Страница авторизации

Регистрация происходит с помощью выдачи так называемого токена регистрации пользователю, который будет действовать в течение

определенного промежутка времени, после чего-либо станет недействительным, и его надо будет обновить.

Список подписчиков

Авторизация Регистрация

abc

...

введите имя

введите фамилию

Загрузить картинку

зарегистрироваться очистить

Рисунок 40 - Страница регистрации

Страница регистрации (рисунок 40) работает по тому же принципу, что и авторизации, за исключением поиска пользователя по логину. Если для авторизации пользователь с введенным логином должен быть найден, то для регистрации – наоборот, логин является уникальным параметром. После создания нового пользователя, авторизация происходит автоматически.

5.4.2 АДМИНИСТРАТОР

После авторизации в системе перед пользователем появится список нейросетевых модулей (рисунок 41).

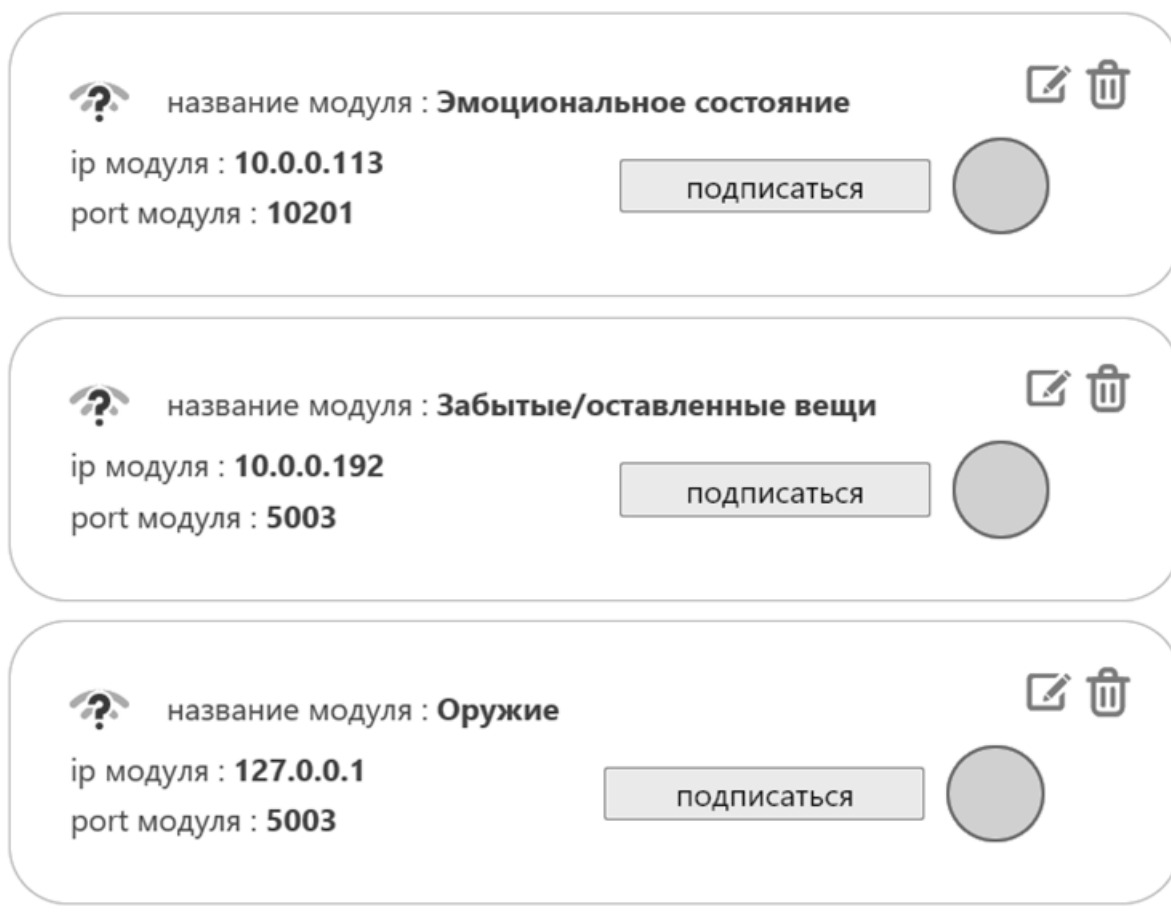


Рисунок 41 - Список нейросетевых модулей

В этом списке можно увидеть название каждого модуля, его IP-адрес, порт, результат анализа и его онлайн статус. Онлайн статусов может быть три (рисунок 42)



Рисунок 42 - Виды онлайн статусов

Слева-на-право: оффлайн – появляется, когда нейросетевой модуль не доступен в сети в связи с проблемами со связью или неисправностью с самим модулем; анализ отключен – модуль доступен в сети, включен и даже отвечает на некоторые запросы, гарантируя свою исправность, но нейросетевой анализ отключен; анализ включен – запущен нейросетевой анализ, что уже позволяет подсоединиться к данному модулю и принимать от него результат анализа.



Новый модуль

ключ модуля	<input type="text"/>
название модуля	<input type="text"/>
ip модуля	<input type="text"/>
порт модуля	<input type="text"/>
частота оповещений (в сек)	<input type="text"/>
сообщение	<input type="text"/>

Рисунок 43 - Страница входного тестирования

На рисунке 43 представлена форма добавления нового модуля в систему, в котором мы должны указать некоторые поля:

- Ключ или кодовое название – сокращенное название модуля, написанное латинскими буквами для обеспечения взаимодействия между пользователем и конкретным модулем (является уникальным);
- Название – расширенная версия ключа, которая необязательно должна быть уникальным. Поддерживает кириллицу и пробелы. Название должно раскрывать смысл действующего модуля;
- IP – адрес, который хранит в себе тот или иной модуль, грубо говоря, определенное физическое устройство, подключенное к сети, на котором функционирует модуль. Если бекенд и модуль расположены на одном компьютере, данное поле стоит заполнять стандартными 127.0.0.1 или localhost;
- Port – адрес запущенного модуля уже на физическом устройстве. **ВАЖНОЕ ПРИМЕЧАНИЕ:** порт должен быть открытым для обеспечения взаимодействия между бекендом и модулем, однако в локальных сетях первого уровня порты по умолчанию открыты;
- Частота оповещений (в сек) – время между отправкой сообщениями при выявлении данным модулем угрозы. Указывается в количествах секунд;
- Сообщение – текст сообщения, который отправится в виде SMS подписчику.

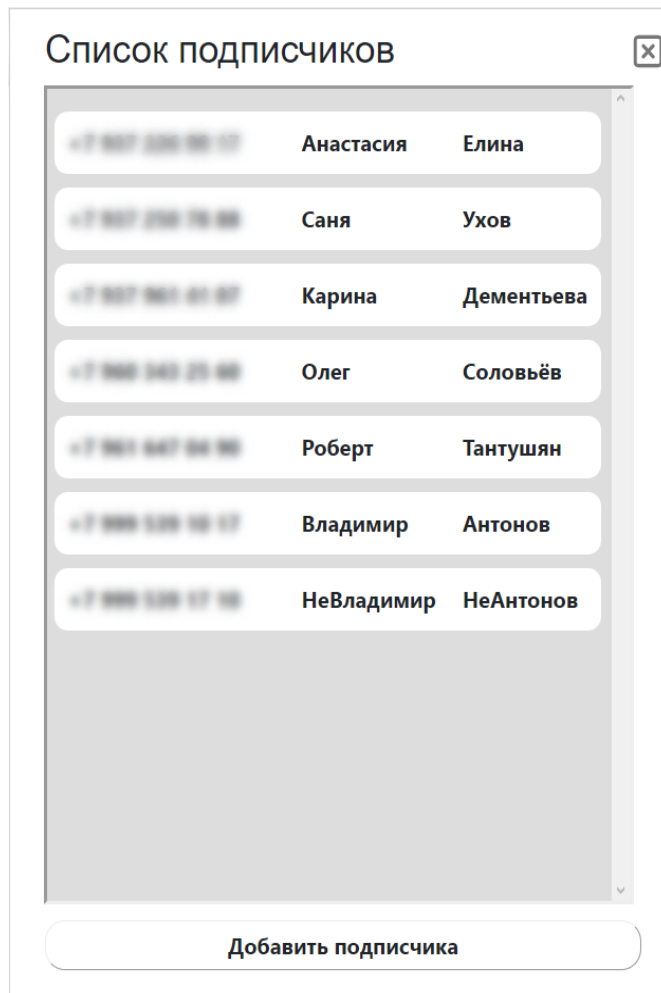


Рисунок 44 - Список подписчиков с номерами телефонов для отправки СМС-оповещения

На рисунке 44 представлено окно со списком подписчиков. Эти подписчики являются сущностями, на которые можно уже воздействовать при помощи набора команд **CRUD**: то есть их так же можно создавать и удалять. Для того, чтобы создать нового подписчика, нужно нажать на кнопку «Добавить подписчика» внизу списка.

The image shows a web form titled "Новый подписчик" (New subscriber). It contains three input fields: "номер телефона" (phone number), "имя" (name), and "фамилия" (surname). Below the fields are two buttons: "Создать подписчика" (Create subscriber) and "Отмена" (Cancel).

Новый подписчик	
номер телефона	<input type="text"/>
имя	<input type="text"/>
фамилия	<input type="text"/>
<div><input type="button" value="Создать подписчика"/> <input type="button" value="Отмена"/></div>	

Рисунок 45 - Форма создания нового подписчика

На рисунке 45 представлена форма создания нового подписчика. Для этого необходимо ввести имя и фамилию подписчика и его номер телефона. Стоит отметить, что номер должен быть в стандартном Российском формате (+7 xxx xxx xx xx). Каждый подписчик можно подписать на СМС рассылку при помощи мессенджера WhatsApp. После добавления подписчика мы возвращаемся к их списку. Для оформления подписки на СМС-рассылку необходимо кликнуть на подписчика. После этого открывается окно «Подписчик» (рисунок 46). Кликая на названия модулей, можно подписаться или отписаться. О статусе подписки свидетельствует цветовая индикация.

Подписчик

Имя :	Роберт	Оружие <i>подписаться</i>
фамилия :	Тантушян	
Номер телефона :	+7 961 647 04 90	
		Эмоциональное состояние <i>подписаться</i>
		Забывшие/оставленные вещи <i>подписаться</i>

Подписчик

Имя :	Владимир	Оружие <i>подписан</i>
фамилия :	Антонов	
Номер телефона :	+7 999 539 10 17	
		Эмоциональное состояние <i>подписаться</i>
		Забывшие/оставленные вещи <i>подписаться</i>

Рисунок 46 - Форма подписки

На рисунке видно, что в каждом подписчике присутствует список модулей и цветовая индексация на статус подписки. Красный модуль с надписью «подписаться» свидетельствует об отсутствии подписки. Это значит, что независимо от результата анализа модулей, пользователю НЕ придет оповещение. Зеленый индикатор говорит об обратном результате.

5.4.3 НЕЙРОСЕТЕВОЙ АНАЛИЗ

Для запуска анализа, необходимо открыть панель настроек нейросетевого модуля (рисунок 26) и нажать на кнопку «Запуск анализа»



Рисунок 47 - Результат анализа на нейросетевом модуле

Для наглядности принципа работы нейросетевого модуля, после запуска анализа открывается окно с отображением выделенного оружия (рисунок 47). На веб камеру было продемонстрирована картинка с Михаилом Тимофеевичем, который держал в руках свое изобретение, и нейронная сеть выделило огнестрельное оружие у него руках.

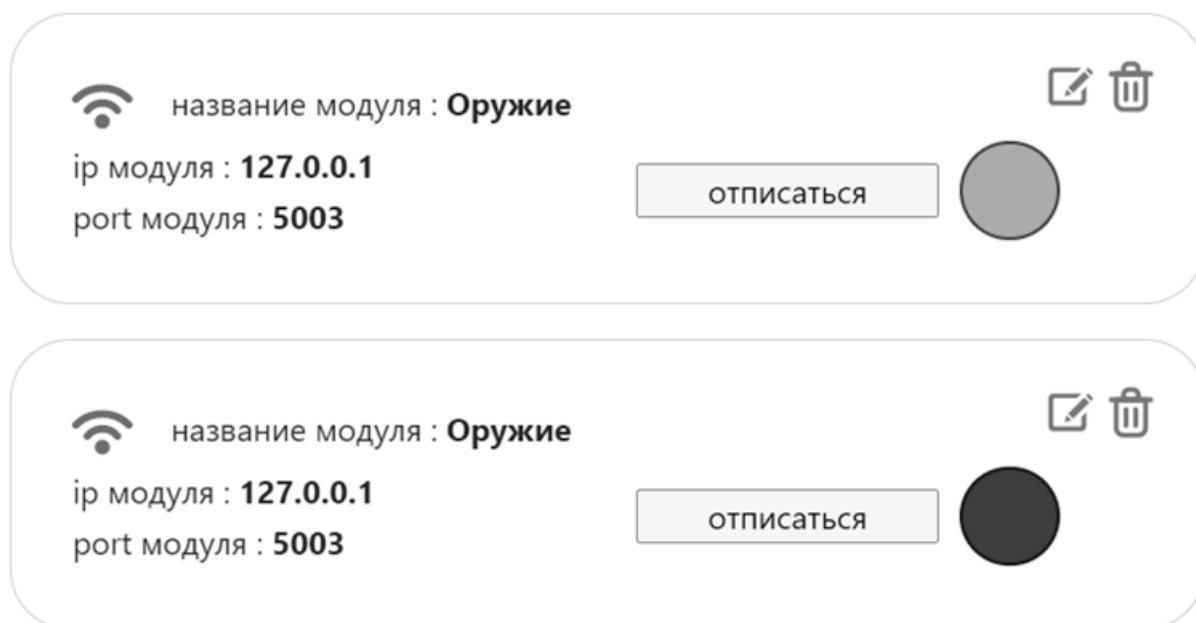


Рисунок 48 - Результат нейросетевого анализа на панели администратора

После положительного результата на панель администратора приходит оповещение о наличии угрозы, которые система отображает в виде красного индикатора. На рисунке 48 показан результат до (сверху) и после (снизу) обнаружения угрозы. Так же система может аккумулировать угрозы со всех модулей по принципу ИЛИ. Это значит, что из всех модулей, зарегистрированных и запущенных в системе, достаточно обнаружить угрозу одной, чтобы показать угрозу.

Если угроза была обнаружена определенным модулем, на который подписан пользователь – ему тут же приходит СМС-оповещение.

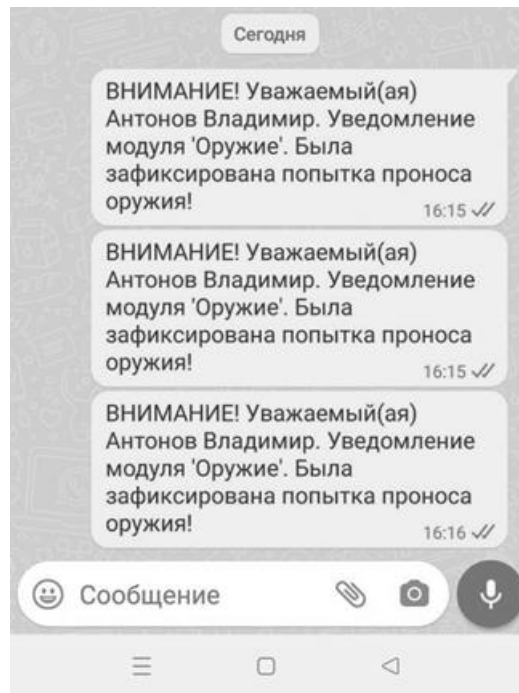


Рисунок 49 - СМС-оповещение в мессенджере WhatsApp

На рисунке 49 видно, как подписчику в приложении WhatsApp пришло СМС-оповещение. В сообщении указано название модуля, от которого пришло оповещение. Имя указывается при создании подписчика. Текст сообщения об угрозе устанавливается в настройках модуля на панели администратора при создании или обновлении модуля. Для этого надо нажать «Создать модуль» или на карандашик на самом модуле соответственно.

В ходе проведенной работы был разработан программный комплекс контроля безопасности, который включает в себя сервер модуля панели администратора, пользовательский интерфейс для взаимодействия с комплексом и модулями, нейросетевой модуль, включающий в себя нейросетевую модель для обработки видеопотока с камер видеонаблюдения и база данных, которая отвечает за долговременное хранение информации о системе.

Функционал разработанной системы полностью удовлетворяет поставленным задачам, а именно: анализ видеопотока на предмет угроз, масштабируемость система. Система позволяет добавлять новые нейросетевые модули, которые должны иметь стандартные API для взаимодействия с панелью администратора.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы был разработан программный интеллектуальный комплекс контроля безопасности посредством обнаружения фактов проноса оружия.

Для решения задач по созданию программного комплекса была изучена предметная область и проведен анализ существующих решений. Спроектированное программное обеспечение успешно реализовано и отвечает всем предъявляемым требованиям.

При создании программного интеллектуального комплекса контроля безопасности посредством обнаружения фактов проноса оружия использовались современные средства разработки, такие как: язык программирования Python, Node.js, Angular, который является наиболее популярным языком создания нейронных сетей, а также следующие его библиотеки: Tensorflow, Pandas Flask, Keras, Numpy, Yolo5 и Torch.

Выпускная квалификационная работа была создана на основе полученных знаний в результате анализа предметной области, анализа существующих решений, изучения методов распознавания изображений, а также разработке и отладке программного комплекса.

Подготовлены и оформлены результаты в виде пояснительной записки, сопроводительных документов, презентации, статьи на тему «Разработка программного интеллектуального комплекса контроля безопасности посредством обнаружения забытых вещей».

Результаты работы докладывались на конференциях «Методы разработки информационных систем и систем управления», «Проблемы

управления в социально-экономических и технических системах», «Проектное обучение». Работа заняла призовое 2 место на конференции «Методы разработки информационных систем и систем управления», «Проблемы управления в социально-экономических и технических системах». Основные результаты отражены в публикациях в статьях «Система видеораспознавания признаков угроз безопасности Safety Eye» в сборнике трудов XIX международной научно-практической конференции «Проблемы управления в социально-экономических и технических системах» (2023 г.), и «Система нейросетевого видеонаблюдения и контроля безопасности» в сборнике трудов XVIII международной научно-практической конференции «Проблемы управления в социально-экономических и технических системах» (2022 г.). Поданы документы на государственную регистрацию программы для ЭВМ.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лысцов Н.А., Мартышкин А.И. Нейронные Сети: Применение и Перспективы // Научное обозрение. Педагогические науки. – 2019. – № 3-2. – С. 35-38;
2. Научное обозрение. Нейронные сети: применение и перспективы [Электронный ресурс]. - Режим доступа: <https://science-pedagogy.ru/ru/article/view?id=1951> Дата обращения: 08.04.2023.
3. Научный аспект. Применение нейронных сетей в сфере безопасности и охранных систем [Электронный ресурс]. - Режим доступа: <https://na-journal.ru/2-2020-informacionnye-tekhnologii/2093-primenenie-nejronnyh-setej-v-sfere-bezopasnosti-i-ohrannyh-sistem> Дата обращения: 08.04.2023.
4. Эпоха нейросетевых технологий в видеонаблюдении началась! [Электронный ресурс]. - Режим доступа: <https://www.dssl.ru/publications/obzory/epokha-neyrosetevykh-tekhnologiy-v-videonablyudenii-nachalas> / Дата обращения: 08.04.2023.
5. У искусственного интеллекта есть когнитивные искажения (непредсказуемые): [Электронный ресурс]. - Режим доступа: <https://2academy.livejournal.com/119562.html> Дата обращения: 08.04.2023.
6. What Is SQLite: [Электронный ресурс]. - Режим доступа: <https://www.sqlite.org/index.html> Дата обращения: 08.04.2023.
7. Ethan Brown. Web Development with Node and Express Full use JavaScript stack. [Текст] СПб.: Питер, 2017

8. Зачем нам jQuery: [Электронный ресурс]. - Режим доступа: <https://itnan.ru/post.php?c=1&p=308134> html Дата обращения: 08.04.2023.
9. Poornima G. Naik, Girish R. Naik. MEAN Stack Web Development Explained to Novice Learners - Vol I (Covers Node.js, Express.js and MongoDB). [Текст] Shashwat publication, 2022
10. The Good and the Bad of Angular Development. [Электронный ресурс]. - Режим доступа: <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-angular-development/> Дата обращения: 08.04.2023.
11. Кристиан Хилл, Научное программирование на Python / пер. с англ. А. В. Снастина. [Текст] – М.: ДМК Пресс, 2021
12. Flask in Production: Minimal Web APIs: [Электронный ресурс]. - Режим доступа: <https://towardsdatascience.com/flask-in-production-minimal-web-apis-2e0859736df> Дата обращения: 08.04.2023.
13. Martín Abadi, Ashish Agarwal. Large-Scale Machine Learning on Heterogeneous Distributed Systems. [Текст] Preliminary White Paper, 2015
14. YOLOv5: The Leader in Realtime Object Detection: [Электронный ресурс]. - Режим доступа: <https://blog.roboflow.com/yolov5-is-here/> Дата обращения: 08.04.2023.
15. Ultramsg: [Электронный ресурс]. - Режим доступа: <https://ultramsg.com/ru/> Дата обращения: 08.04.2023.

16. Tao Lin, J. Gao, X. Fu, and Y. Lin, "A Novel Bug Report Extraction Approach", 15th International Conference on Algorithms and Architectures for Parallel Processing, 2015.
17. Mikhail Nikitinskiy D. Ju. Chalyy. Performance analysis of trickles and TCP transport protocols under high-load network conditions. [Текст] Dept. of Computer Science, 2013
18. JavaScript Object Notation: [Электронный ресурс]. - Режим доступа: <https://guide.couchdb.org/draft/json.html> Дата обращения: 08.04.2023.
19. Understanding HTTP – The internet’s communication protocol: [Электронный ресурс]. - Режим доступа: <https://medium.com/swlh/understanding-http-the-internets-communication-protocol-4f93212a21d1> Дата обращения: 09.04.2023.
20. Web Development Notes from a Data Scientist [Электронный ресурс]. - Режим доступа: <https://towardsdatascience.com/web-development-notes-from-a-data-scientist-95e4794fb4ab> Дата обращения: 09.04.2023.
21. Why you should use Node.js for data science [Электронный ресурс]. - Режим доступа: <https://opensource.com/article/20/6/data-science-nodejs> Дата обращения: 09.04.2023.
22. А. В. Сулыз, А. Н. Панфилов. Технология разработки одностраничного веб-приложения на платформе Angular. [Текст] УДК-004.422.81 ДГТУ – 2020
23. Detecting objects with YOLOv5 [Электронный ресурс]. - Режим доступа: <https://medium.com/mlearning-ai/detecting-objects-with-yolov5-opencv-python-and-c-c7cf13d1483c> Дата обращения: 09.04.2023.

24. Неустроев А. В. Использование в фреймворке Flask библиотеки SQLAlchemy. [Текст] ОО метод, г. Якутск, 2019
25. Нотация IDEF0 [Электронный ресурс]. – Режим доступа: <https://www.businessstudio.ru/wiki/docs/current/doku.php/ru/csdesign/bpmodeling/idef0>. Дата обращения: 09.04.2023
26. Методология sadt. Семейство стандартов idef. [Электронный ресурс]. – Режим доступа: <https://studfile.net/preview/6266132/page:5/> Дата обращения: 09.04.2023
27. Основы IDEF3 [Электронный ресурс]. – Режим доступа: <https://www.trinion.org/blog/opisanie-notacii-idef3>. Дата обращения: 09.04.2023
28. Что такое диаграмма DFD и как ее создать? [Электронный ресурс]. – Режим доступа: <https://www.lucidchart.com/pages/ru/диаграмма-dfd>. Дата обращения: 09.04.2023
29. Грейди Буч, Джеймс Рамбо, Айвар Джекобсон. Язык UML. Руководство пользователя = The Unified Modeling Language user guide. — 2-е изд. — М., СПб.: ДМК Пресс, Питер, 2004.
30. Бабич А. В. Введение в UML. ISBN 978-5-94774-878-9, 6. Лекция: Диаграммы прецедентов: крупным планом. М., СПб.: ДМК Пресс, Питер, 2007
31. Простое руководство по диаграммам активности UML (Activity diagram) [Электронный ресурс]. – Режим доступа: <https://creately.com/blog/ru/uncategorized-ru/учебник-по-диаграммеактивности/>. Дата обращения: 09.04.2023

32. Диаграмма последовательности UML [Электронный ресурс]. – Режим доступа: <https://pro-prof.com/archives/2769/>. Дата обращения: 10.04.2023
33. Краткие теоретические сведения о диаграммах классов [Электронный ресурс]. – Режим доступа: https://flexberry.github.io/ru/gpg_classdiagram.html. Дата обращения: 10.04.2023
34. Диаграммы пакетов [Электронный ресурс]. – Режим доступа: <https://cdn1.ozone.ru/multimedia/1005433571.pdf>. Дата обращения: 10.04.2023
35. Комплекс детектирования оружия "КДО-1" [Электронный ресурс]. – Режим доступа: <https://domination.one/solutions/kompleks-detektirovaniya-oruzhiya-kdo-1/> Дата обращения: 10.04.2023
36. Система событийного видеонаблюдения [Электронный ресурс]. – Режим доступа: <https://www.goal.ru/neuronet/weapon/> Дата обращения: 10.04.2023
37. Gunchack распознавание оружия в видеопотоке [Электронный ресурс]. – Режим доступа: <https://metakom.tech/portfolio/guncheck-raspoznavanie-oruzhiya-v-videopotoke/> Дата обращения: 10.04.2023
38. «Антиколумбайн» [Электронный ресурс]. – Режим доступа: <https://rb.ru/news/anticolumbine-system-russia/> Дата обращения: 10.04.2023
39. Распознавание объектов с помощью YOLO [Электронный ресурс]. – Режим доступа: <https://proglib.io/p/raspoznavanie-obektov-s-pomoshchyu-yolo-v3-na-tensorflow-2-0-2020-11-08> Дата обращения: 10.04.2023