

Минобрнауки России

Юго-Западный государственный университет

Кафедра программной инженерии

ОТЧЕТ

о преддипломной (производственной) практике

наименование вида и типа практики

на (в) ООО «Мцоб.Онлайн-сервисы»

наименование предприятия, организации, учреждения

Студента 4 курса, группы ПО-116

курса, группы

Конева Андрея Вячеславовича

фамилия, имя, отчество

Руководитель практики от
предприятия, организации,
учреждения

Оценка _____

директор

должность, звание, степень

Куркина А. В.

фамилия и. о.

подпись, дата

Руководитель практики от
университета

Оценка _____

к.т.н. доцент

должность, звание, степень

Чаплыгин А. А.

фамилия и. о.

подпись, дата

Члены комиссии

подпись, дата

фамилия и. о.

подпись, дата

фамилия и. о.

подпись, дата

фамилия и. о.

Курск 2025 г.

СОДЕРЖАНИЕ

1	Анализ предметной области	4
1.1	Актуальность проблемы загрязнения водоемов	4
1.2	Нефтяные разливы	4
1.2.1	Экологическая опасность нефтяных разливов	4
1.2.2	Причины и источники загрязнений	7
1.3	Методы мониторинга разливов нефти	8
1.4	Нейронные сети	10
1.4.1	История появления нейронных сетей	10
1.4.2	Современные типы нейронных сетей	14
2	Техническое задание	18
2.1	Основание для разработки	18
2.2	Цель и назначение разработки	18
2.3	Требования пользователя к программной системе	18
2.3.1	Требования к данным программной системы	18
2.3.2	Функциональные требования к программной системе	19
2.3.2.1	Сценарий использования «Обучение нейронной сети»	20
2.3.2.2	Сценарий использования «Тестирование нейронной сети»	21
2.3.2.3	Сценарий использования «Анализ изображения»	22
2.3.2.4	Сценарий использования «Сохранение результата анализа»	23
2.3.3	Требования пользователя к интерфейсу приложения	24
2.4	Нефункциональные требования к программной системе	24
2.4.1	Требования к надежности	24
2.4.2	Требования к аппаратному обеспечению	25
2.4.3	Требования к программному обеспечению	25
2.5	Требования к оформлению документации	26
3	Технический проект	27
3.1	Общая характеристика организации решения задачи	27
3.2	Обоснование выбора технологии проектирования	27
3.2.1	Язык программирования Python	27

3.2.2	Описание библиотеки PyTorch	28
3.2.3	Описание библиотеки OpenCV	28
3.2.4	Описание фреймворка PyQt6	29
3.2.5	Описание библиотеки NumPy	29
3.3	Архитектура программной системы	30
3.4	Описание нейронной сети	31
3.4.1	Метод распознавания пятен нефти	32
3.4.2	Архитектура U-Net	32
3.4.3	Структура нейронной сети	33
3.4.4	Обучение нейронной сети	35
3.5	Проектирование пользовательского интерфейса	35
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	40

1 Анализ предметной области

1.1 Актуальность проблемы загрязнения водоемов

Вода является одним из важнейших природных ресурсов, как для поддержания жизнедеятельности биологических систем, так и для обеспечения разнообразных промышленно-технологических процессов. Сегодня вода активно используется в различных ключевых отраслях. В металлургии вода применяется в процессах флотации - обогащения различных руд, а также для охлаждения доменных печей. Химическая промышленность активно задействует воду для очищения и охлаждения оборудования на различных нефтеперерабатывающих производствах, как компонент для создания продуктов нефтехимической отрасли, таких как пластмассы, реагенты и растворители, удобрения. Вода также активно применяется как один из материалов для создания фармацевтической продукции, такой как инъекционные растворы, препараты для наружного применения (спреи и капли). Энергетический сектор также использует воду для охлаждения энергетического оборудования на атомных и теплоэлектростанциях. Кроме того, вода так же обеспечивает работу гидроэлектростанций.

Загрязнение водоемов негативно сказывается на экологии, здоровье населения, препятствует развитию экономики, поэтому обеспечение экологической безопасности водоемов является одной из наиболее важных задач экологии. В настоящее время особую актуальность приобретает проблема нефтяных разливов.

1.2 Нефтяные разливы

1.2.1 Экологическая опасность нефтяных разливов

Разлив нефти – аварийный выброс нефти или нефтепродуктов в окружающую среду. Нефтяные разливы происходят с высокой частотой и являются серьезной проблемой нефтегазовой отрасли. Каждый разлив приводит к масштабным последствиям для окружающей среды и требует значительных

финансовых затрат для устранения самого разлива и минимизации экологического ущерба.

Ярким примером является инцидент на буровой платформе Deepwater Horizon, принадлежавшей британской нефтегазовой компании British Petroleum. 20 апреля 2010 года, в результате нарушения технологических норм цементировании скважины, произошел взрыв, который привел к гибели 11 человек и неконтролируемому выбросу нефти в Мексиканский залив на протяжении нескольких месяцев. В окружающую среду попало около 400 тысяч тонн нефти. Эта катастрофа, ставшая крупнейшим в истории США разливом нефти, привела к массовой гибели животных, таких как дельфины, морские черепахи, тем самым нанеся колоссальный ущерб морским экосистемам, по некоторым оценкам сохраняющийся и по сей день. Расходы на ликвидацию происшествия, а также затраты на штрафы и компенсации, составили около 65 миллиардов долларов.

16 марта 1978 года французский супертанкер Amoco Cadiz, попав в шторм, потерял управление и разбился о скалы у берегов французского региона Бретань. Это происшествие привело к выбросу 220 тысяч тонн сырой нефти в Атлантический океан, загрязнив более 300 километров французского побережья. Пролившаяся нефть образовала пятно протяженностью 19 километров и поникла в песок на глубину до 50 сантиметров, образовав асфальтоподобные корки, сохранявшиеся на протяжении нескольких лет. В защищенных от волн районах нефть сохранялась до 10 лет, тем самым замедляя восстановление экосистемы, понесшей масштабный ущерб – загрязнение вызвало гибель около 20 тысяч птиц, множества донных организмов, таких как моллюски и ракообразные. Рыбаки сообщали о сокращении уловов в два раза, а сама рыба нередко имела различные язвы и опухоли. Компания Амосо, владелец танкера, выплатила Франции компенсацию в размере 230 миллионов долларов.

Не менее разрушительными оказались последствия крушения танкера Prestige, затонувшего в ноябре 2002 года. Получив повреждения у побережья Испании, корпус судна раскололся, что привело к попаданию примерно 60

тысяч тонн нефти в Атлантический океан. Растянувшееся на более чем 2000 километров вдоль побережья Галисии нефтяное пятно поставило под угрозу существование 25 охраняемых видов животных, уничтожило популяции рыб, кораллов, моллюсков. Согласно государственной оценке, зачистка побережья Галисии обошлась в 2,5 миллиарда евро, а нанесенный ущерб составил 368 миллионов евро.

В июле 2010 года нефтепровод, принадлежащий американской корпорации Enbridge, разорвало в результате коррозии трубы, что послужило причиной попадания более 2,7 тонн битумной нефти в реку Каламазу, пролившейся на 40 километров по руслу реки и осевшей на дне. Осевшая нефть отравила донные отложения реки, что привело к массовой гибели рыб, популяции которых начали возвращаться только спустя 10 лет. Кроме того, токсины сделали речную воду непригодной для питья, а из-за ядовитых испарений нефти некоторые жители близлежащих районов жаловались на головные боли и тошноту. Поскольку убирать пролившуюся нефть пришлось не только с поверхности, но и со дна реки, очистные мероприятия были значительно затруднены, что повысило стоимость их проведения – она составила примерно 1,2 миллиарда долларов.

Утром 15 декабря 2024 года в Керченском проливе в результате шторма танкеры «Волгонефть-212» и «Волгонефть-239», перевозившие в общей сложности около 9000 тонн мазута, потерпели крушение. По различным оценкам в море попало от 2,5 до 5 тысяч тонн мазута, разлившегося по площади более 400 м². Загрязнение затронуло около 70 километров береговой линии и привело к гибели более 200 птиц и 70 дельфинов, среди которых были охраняемые виды. Реагируя на происшествие, власти ввели режим чрезвычайной ситуации на федеральном уровне и объявили о начале масштабных очистных работ с участием 300 человек и 60 единиц специальной техники, продолжающихся по сей день.

Вследствие принятия множества международных законов и соглашений, совершенствования технологических процессов можно отметить уменьшение частоты крупных происшествий, приводящих к масштабным разли-

вам нефти. Однако, некоторые исследования[1] отмечают, что информация о разливах нефти не всегда достоверна, утверждая, что многие менее крупные инциденты никогда не попадают в поле зрения государственных служб контроля и СМИ, что приводит к отсутствию данных об этих инцидентах в международных и государственных базах.

1.2.2 Причины и источники загрязнений

Основными источниками разливов нефти являются различные утечки на этапах добычи, хранения и транспортировки нефти и нефтепродуктов. Причинами возникновения утечек могут являться как стихийные бедствия, такие как штормы, землетрясения и эрозия, так и техногенные факторы, включая отказы оборудования, человеческие ошибки. В таблице 1.1 указаны основные источники попадания нефти в мировой океан[2].

Таблица 1.1 – Основные источники попадания нефти в мировой океан

Источник	млн. т/год
Морская транспортировка (кроме аварийных разливов)	1,83
Аварийные разливы	0,3
Речной сток, включая сточные воды городов	1,9
Сточные воды прибрежной зоны	0,8
Естественные нефтяные скважины	0,6
Добыча нефти в море	0,08
Всего	5,51

Наиболее часто разливы нефти появляются в результате аварии на танкере, перевозящем нефть или нефтепродукты. Среди причин возникновения таких аварий можно выделить столкновения, посадки на мель, штормовые повреждения, нарушение герметичности резервуаров. Эти происшествия нередко приводят к масштабным выбросам нефти в морскую или речную среду, при этом загрязняя не только воду, но и зачастую прибрежные зоны, в районе которых произошла авария.

Морская добыча нефти проводится с помощью масштабных комплексов, использующих комплекс инженерно-технических решений. Такие происшествия, как повреждения герметизации скважин, отказ оборудования, ошибки при цементировании или нарушения технологических регламентов приводят к выбросам добываемой нефти из подземных резервуаров, что в свою очередь приводит к разливам, плохо поддающимся локализации.

Трубопроводы, используемые для транспортировки нефти на большие расстояния, могут дать течь вследствие коррозии труб, механических повреждений, ошибок в обслуживании. В результате в воде может образоваться нефтяной разлив, который может долго оставаться незамеченным, если трубопровод проходит под водой. Кроме того, серьезную угрозу загрязнения представляет преднамеренное разрушение нефтяной инфраструктуры, особенно в зонах активных боевых действий. Атаки на эти объекты не только приводят к масштабным утечкам, но и затрудняют обнаружение и ликвидацию последствий разливов.

Промышленные предприятия, связанные с переработкой нефти, могут сбрасывать загрязненные нефтью или продуктами ее переработки воды в близлежащие водоемы. Наряду с этим отходы ремонтных мастерских, автозаправочных станций также могут попадать в водоемы, что приводит к еще большему загрязнению.

Нефтяные утечки происходят не только по вине человека, но и в ходе естественного просачивания из нефтеносных пластов земли через трещины в коре. Чаще всего это происходит в районах активной геологической активности. Несмотря на относительно незначительные объемы, эти утечки также оказывают негативное влияние на экосистемы.

1.3 Методы мониторинга разливов нефти

Своевременное обнаружение и ликвидация нефтяных разливов – ключевые задачи для сохранения экологической безопасности водоемов. В связи с этим широко распространены различные виды мониторинга состояния водоемов, среди которых выделяются визуальные методы наблюдения за по-

верхностью воды. В отличие от лабораторных исследований, визуальное исследование позволяет намного быстрее охватить большие территории и не требует физического контакта с объектом, что позволяет оперативно обнаружить новые разливы нефти, особенно в труднодоступных акваториях.

Одним из самых распространенных методов визуального мониторинга является спутниковая съемка. Современные спутники, оснащенные высокоточными сенсорами, позволяют получить качественные снимки поверхности воды и хорошо отображают аномалии, такие как нефтяные пятна. Спутниковая съемка охватывает сотни квадратных километров поверхности Земли в одном снимке, а использование групп спутников позволяет получать новые данные с высокой периодичностью. Кроме того, использование многих спутников бесплатно, а данные размещены в открытом доступе, что снижает стоимость процедур мониторинга. Однако, зависимость космической съемки от погодных условий и времени суток значительно осложняет наблюдение за поверхностью водоемов и является существенным недостатком этого метода.

В условиях плохой видимости, ограничивающих работу спутников, применяются радиолокационные спутниковые снимки. В отличие от оптических сенсоров, радары полагаются на электромагнитные сигналы и фиксируют отраженный поверхностью воды отклик, что и позволяет вести съемку в любую погоду и время суток. На радарных снимках нефтяные пятна обычно проявляются в качестве темных областей, что повышает их информативность в задачах обнаружения нефтяных пятен и пленок. Так, исследование радиолокационных снимков Керченского пролива с использованием спутников Sentinel-1A/B за период с 2017 по 2021 годы показало высокую точность этого метода и позволило исследовать частоту появления, масштаб нефтяных разливов и их источники на основании 2597 пятен, появившихся на снимках за этот период[3]. Несмотря на гибкость этого метода, полученные снимки нередко проявляют сторонние объекты, например различные биологические образования, создавая визуально похожие участки на изображениях. Наличие таких ложных данных требует участия эксперта, отличающего настоящие нефтяные пятна от схожих объектов, в процессе мониторинга. Кроме

того, радиолокационные изображения значительно дороже, что еще больше повышает стоимость мониторинга.

Наконец, высокую эффективность показал метод аэросъемки поверхностей воды. Вместе с развитием беспилотных летательных аппаратов, или БПЛА, наблюдение за водоемами с малой высоты стало высокоэффективным и экономически оправданным методом мониторинга. БПЛА часто оснащаются камерами высокого разрешения, что позволяет получать самые качественные снимки из перечисленных методов, а высокая степень гибкости в построении маршрутов и простота управления делают этот метод самым доступным. Недостатками аэросъемки с помощью БПЛА являются ограниченность радиуса и времени полета, что делает его малоэффективным для мониторинга больших поверхностей воды, таких как моря. Беспилотные аппараты часто используются в составе систем мониторинга вместе с спутниковыми системами для уточняющих съемок в определенных секторах, снимки которых получены из космоса.

Перечисленные выше дистанционные методы используются для получения визуальных данных, являющихся источником информации для обнаружения нефтяных разливов. Чтобы обнаружить загрязнение, необходимо проанализировать полученные данные. Учитывая высокое количество изображений, полученных для анализа, этот процесс занимает значительное время. Более того, если изображения получаются в режиме реального времени, например в формате видеосъемки, то для обнаружения разлива нефти необходимо постоянное присутствие наблюдателя. В настоящее время для задач анализа изображений все чаще используются нейронные сети.

1.4 Нейронные сети

1.4.1 История появления нейронных сетей

Идеи, которые легли в основу современных нейронных сетей, берут свое начало в философии и математике, а основные принципы исследовались задолго до появления вычислительных систем как таковых. Правила, руково-

дящие рациональной частью мышления были сформированы Аристотелем в далеких 384 – 322 годах до нашей эры. Впоследствии формальные правила рассуждений и процессы мышления стали предметом исследований не менее известных философов, среди которых были Томас Гоббс (1588-1679), Рене Декарт (1596-1650) и Рудольф Карнап (1891-1970)[4]. Известные математики, такие как Джордж Буль (1815-1864) и Готтлоб Фреге (1848-1925), внесли вклад в разработку логических систем, впоследствии ставших источниками исследований моделирования искусственного интеллекта.

Первая модель искусственного нейрона была предложена Уорреном МакКаллоком и Уолтером Питтсом в 1943 году. Модель представленного ими логического устройства показана на рисунке 1.1.

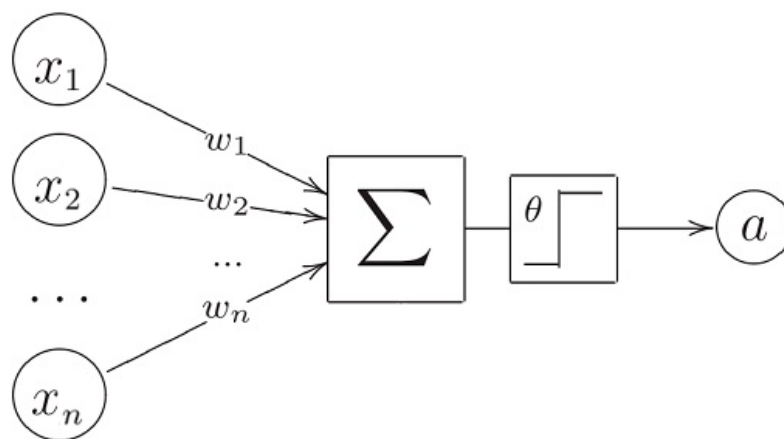


Рисунок 1.1 – Искусственный нейрон

Модель состоит из входов x_1, x_2, \dots, x_n , весов w_1, w_2, \dots, w_n , сумматора и функции активации. Попадая на входы, каждое значение x_i умножается на значение соответствующего веса w_i , обозначающего его важность, после чего поступает на сумматор, вычисляющий сумму всех поступивших на него сигналов:

$$z = w_1x_1 + w_2x_2 + \dots + w_nx_n$$

Полученная сумма затем сравнивается с пороговым значением θ . Если сумма больше, то нейрон возбуждается (выходное значение $a = 1$), в противном

случае остается в состоянии покоя (выходное значение $a = 0$). Данная модель может выполнять только бинарные операции, получая на вход значения 0 и 1, требует ручного задания весов, также являющихся бинарными, и порогового значения, что значительно ограничивает её возможности.

Несмотря на ограничения, предложенный МакКаллоком и Питтсом искусственный нейрон показал способность выполнять базовые логические операции, такие как И, ИЛИ и НЕ. Эта модель стала основой для дальнейшей теории нейронных вычислений, а все современные нейронные сети состоят из доработанных версий оригинальных нейронов.

Спустя несколько лет идеи, взяв за основу исследования МакКаллока и Питтса, американский ученый Фрэнк Розенблатт предложил схему перцептрона - устройства, моделирующего восприятие информации головным мозгом человека. Перцептрон состоит из входных датчиков, ассоциативных и реагирующих элементов[5]. Схема простейшего перцептрона показана на рисунке 1.2.

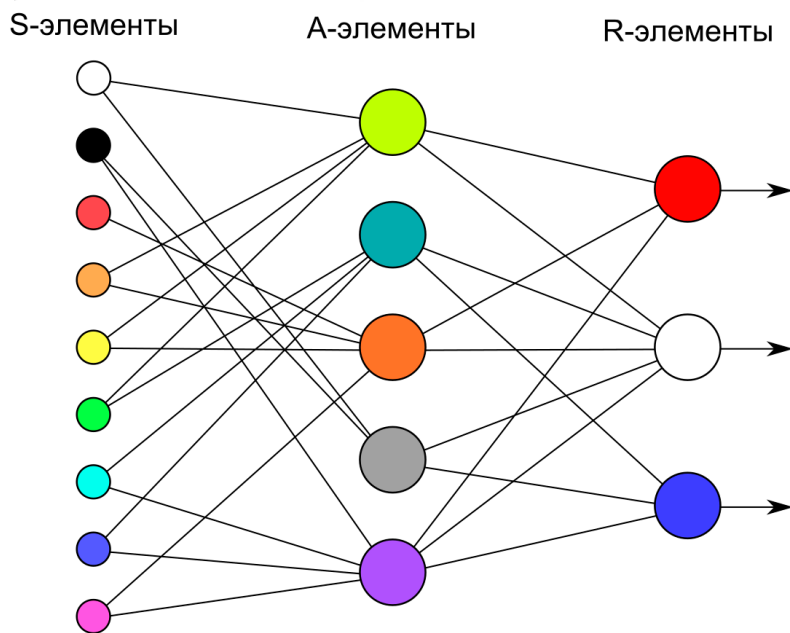


Рисунок 1.2 – Схема однослойного перцептрона

Однослойный перцептрон подходит для решения задачи классификации данных. Для обучения задаются не только входные значения, но и известные ожидаемые результаты. Как и в случае работы нейрона МакКаллока - Питтса, входные значения (S-слой) передаются на ассоциативные элементы (А-слой), где вычисляется взвешенная сумма поступивших сигналов, являющихся произведениями входных значений и весов связей и, наконец, если сумма превышает заданный порог θ , то результат подается на реагирующий элемент (R-слой). Если выходное значение не совпадает с ожидаемым, то веса связей S–А модифицируются по формуле:

$$w_{ij}(t + 1) = w_{ij}(t) + \eta \cdot \delta \cdot X_j,$$

где t и t_1 – номера текущей и следующей итераций процесса, η – коэффициент обучения, находящийся в диапазоне $0 < \eta < 1$, δ – разность между ожидаемым и полученным значением, i и j – номера входа и нейрона в слое соответственно, после чего перцептрон заново обрабатывает входные значения до тех пор, пока все выходные значения не совпадут с ожидаемыми.

Однако, несмотря на успехи в исследованиях, дальнейшее изучение нейронных сетей столкнулось с скептицизмом. Основной причиной стала невозможность однослойных сетей решать многие простые задачи, среди которых операция «исключающего ИЛИ». Формальное доказательство невозможности решения нелинейных задач однослойным перцептроном было предоставлено Марвином Минским и Сеймуром Папертом в книге «Перцептроны», выпущенной в 1969 году.

После выхода книги, исследовательский интерес к области нейронных систем значительно упал, но не исчез полностью. В 1970-х активно проводились исследования многослойных перцептронов, был разработан алгоритм обратного распространения ошибки для их обучения, позволяющий преодолеть многие ограничения, обозначенные в работе «Перцептроны», но впоследствии оказавшийся не универсальным. Также разрабатывались альтернативные модели нейронных сетей, такие как когнитрон, предложенная Кунихико Фукусимой в 1975 году, ставшая первой моделью, реализующей обу-

чение без учителя. Основной идеей когнитрона является чередование возбуждающих и подавляющих слоев, что позволяет извлекать отдельные признаки и обобщать изображения. Обучение без учителя основано на конкуренции между нейронами – чем сильнее конкретный вход, тем больший вес ему присваивается, что приводит к заглушению соседних нейронов в конкретном слое. Через пять лет Фукусима представил доработанную и более устойчивую версию когнитрона – неокогнитрон, состоящий из чередующихся слоев S-элементов, отвечающих за выделение признаков, и C-элементов, подавляющих искажения. Когнитрон и неокогнитрон эффективны для решения задач распознавания образов и стал основой для современных сверточных нейронных сетей.

Ключевым моментом развития нейронных сетей стала формулировка метода обратного распространения ошибки. Впервые этот метод был описан в 1974 году Александром Галушкиным и Полом Вербосом, одновременно и независимо сформулировавшим его, и формализован в 1986 году Дэвидом Румельхартом, Джеффри Хинтоном и Рональдом Уильямсом. Основной идеей стало распространение ошибки в обратном прямому распространению сигналов направлении. Метод вычисляет градиент ошибки по каждому весу в сети, начиная с выходного слоя и двигаясь к входному, что позволяет корректировать веса для минимизации ошибки предсказания. Появление обратного распространения позволило обучать многослойные нейронные сети на сложных зависимостях и выполнять более точную классификацию.

1.4.2 Современные типы нейронных сетей

В настоящее время нейронные сети являются эффективными инструментами машинного обучения, успешно применяемые в широком спектре задач. На сегодняшний день доступно большое количество различных архитектур нейронных сетей, обладающих уникальными свойствами и наиболее подходящих для конкретных задач.

Многослойный перцептрон, также известный как сеть прямого распространения, является базовой архитектурой полносвязной нейронной сети.

Многослойный перцептрон состоит из входного слоя, одного или нескольких скрытых слоев и выходного слоя, причем каждый элемент конкретного слоя связан со всеми элементами следующего за ним слоя.

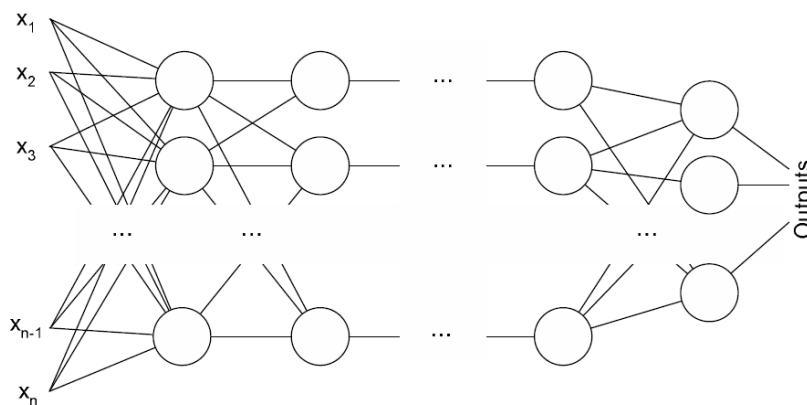


Рисунок 1.3 – Многослойный перцептрон

Во время обработки данные последовательно передаются от входа к выходу без использования сторонних циклов или обратной связи. Обучение полносвязных сетей выполняется при помощи алгоритма обратного распространения ошибки, используемого вместе с параллельным спуском. Эти нейронные сети позволяют аппроксимировать и приближать практически любые непрерывные функции. Недостатками данной модели является плохая масштабируемость на задачи с большими входными данными, а также большие временные затраты на обучение, что делает ее непригодной для работы с изображениями.

Рекуррентная нейронная сеть разработана для работы с последовательными данными, например текстом, аудиосигналами. Отличительной чертой этого типа является наличие внутренней памяти, позволяющей учитывать контекст предыдущих вычислений при обработке новых данных. Базовой архитектурой рекуррентной сети является сеть входных, скрытых и выходных узлов, каждый из которых соединен с остальными. На вход каждого шага вместе с входными данными подается результат обработки предыдущего шага, что позволяет моделировать временные зависимости. Из-за этого метода обработки данных могут возникать проблемы с исчезновением или взрывом градиента ошибки, что было решено при разработке усовершенствованной

сети с долгой краткосрочной памятью, LSTM. LSTM является сетью с ячейками памяти, позволяющими сохранять информацию на длительные промежутки времени. Обычно формируется с использованием «вентилей», использующихся для контроля информации на входах и выходах памяти блоков.

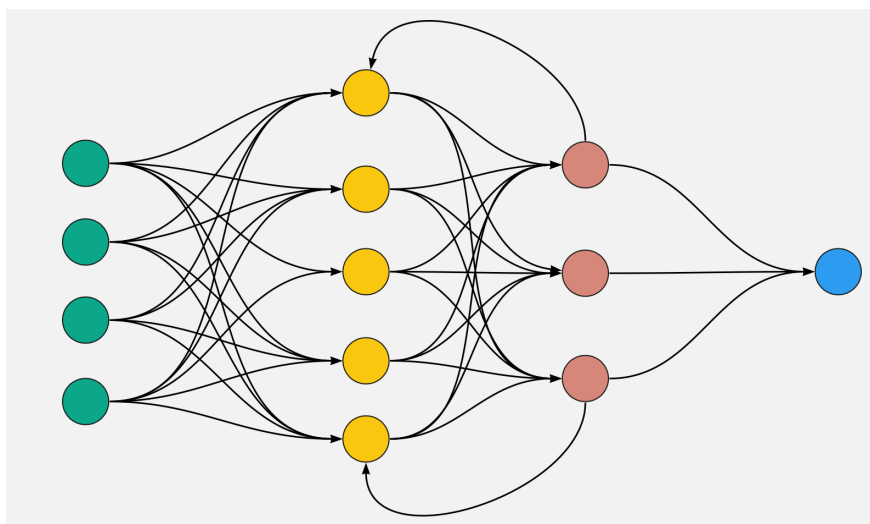


Рисунок 1.4 – Рекуррентная нейронная сеть

Сверточная нейронная сеть – однонаправленная многослойная сеть, направленная на распознавание образов.

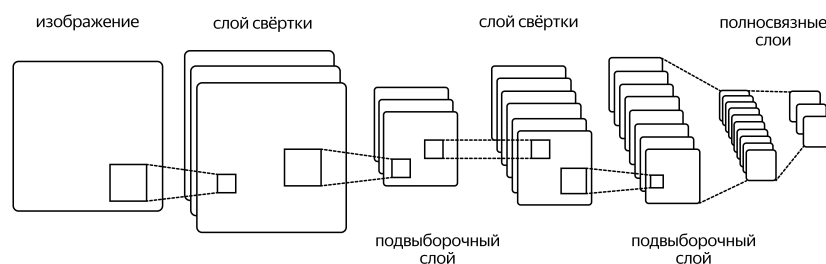


Рисунок 1.5 – Сверточная нейронная сеть

Основным отличием от полносвязных сетей является использование сверточных слоев, применяющих одинаковые фильтры, называемые ядрами свертки. Эти ядра являются матрицами весов и последовательно перемещаются по обрабатываемому слою, вычисляя локальную свертку – взвешенную сумму значений в пределах области матрицы, тем самым вычисляя локаль-

ные признаки. На выходе каждого фильтра формируется карта признаков, отражающая степень активации конкретного признака на различных участках слоя. Сверточные слои обычно чередуются с слоями подвыборки, сжимающими полученные карты признаков, выделяя и сохраняя наибольшее значение из конкретной небольшой области. После нескольких сверточных и подвыборочных слоев карты признаков передаются на полносвязные слои, обрабатывающие полученные данные. Сверточные нейронные сети чаще всего используются для обработки изображений, в частности задач сегментации.

2 Техническое задание

2.1 Основание для разработки

Основанием для разработки является задание на выпускную квалификационную работу бакалавра «Интеллектуальная система мониторинга и распознавания загрязнений водоемов».

2.2 Цель и назначение разработки

Программная система предназначена для автоматического распознавания пятен нефтяных разливов на изображениях и их визуального выделения.

Пользователи должны иметь возможность загружать собственные изображения для анализа системой. Также должен быть реализован функционал сохранения полученных результатов программного анализа .

Задачами данной разработки являются:

- создание интеллектуальной системы распознавания на основе технологий нейронных сетей;
- обучение созданной интеллектуальной системы для распознавания пятен нефтяных разливов;
- разработка функционала для тестирования точности анализа полученной системы;
- реализация настольного приложения с графическим интерфейсом для взаимодействия пользователей с системой.

2.3 Требования пользователя к программной системе

2.3.1 Требования к данным программной системы

Для обучения нейронной сети и анализа изображений на предмет наличия пятен разливов нефти программной системе требуется датасет, состоящий из изображений в формате JPEG, сохраненные в отдельной папке. Кроме того, для сохранения параметров обученной модели нейронной сети исполь-

зуются файлы формата .pth, содержащие веса и смещения модели. Для анализа изображений и тестирования нейронной сети также требуются файлы в формате .pth, содержащие параметры, предварительно полученные после обучения нейронной сети.

2.3.2 Функциональные требования к программной системе

Разработанная программа должна реализовывать следующие функциональные возможности:

- обучение нейронной сети: после выбора директории с изображениями и директории сохранения полученных параметров, программа запускает процесс обучения нейронной сети и, по его завершении, сохраняет полученный набор параметров в указанную пользователем директорию;
- тестирование нейронной сети: после загрузки тестового изображения и предварительно сохраненного файла параметров нейронной сети, программа должна обрабатывать полученное изображение при помощи модели сети, выводить результат анализа в виде бинарного изображения, идеальный ожидаемый результат, полученный с помощью простого алгоритма обработки и метрики, оценивающие эффективность работы нейронной сети;
- распознавание нефтяных пятен: программа должна распознавать пятна разливов нефти на фотографиях при помощи предоставленных пользователем параметров на загруженном изображении;
- сохранение результатов: пользователь должен иметь возможность сохранения проанализированных изображений на жесткий диск.

На рисунке 2.1 предоставлены функциональные требования к системе, представленные в виде диаграммы прецедентов.

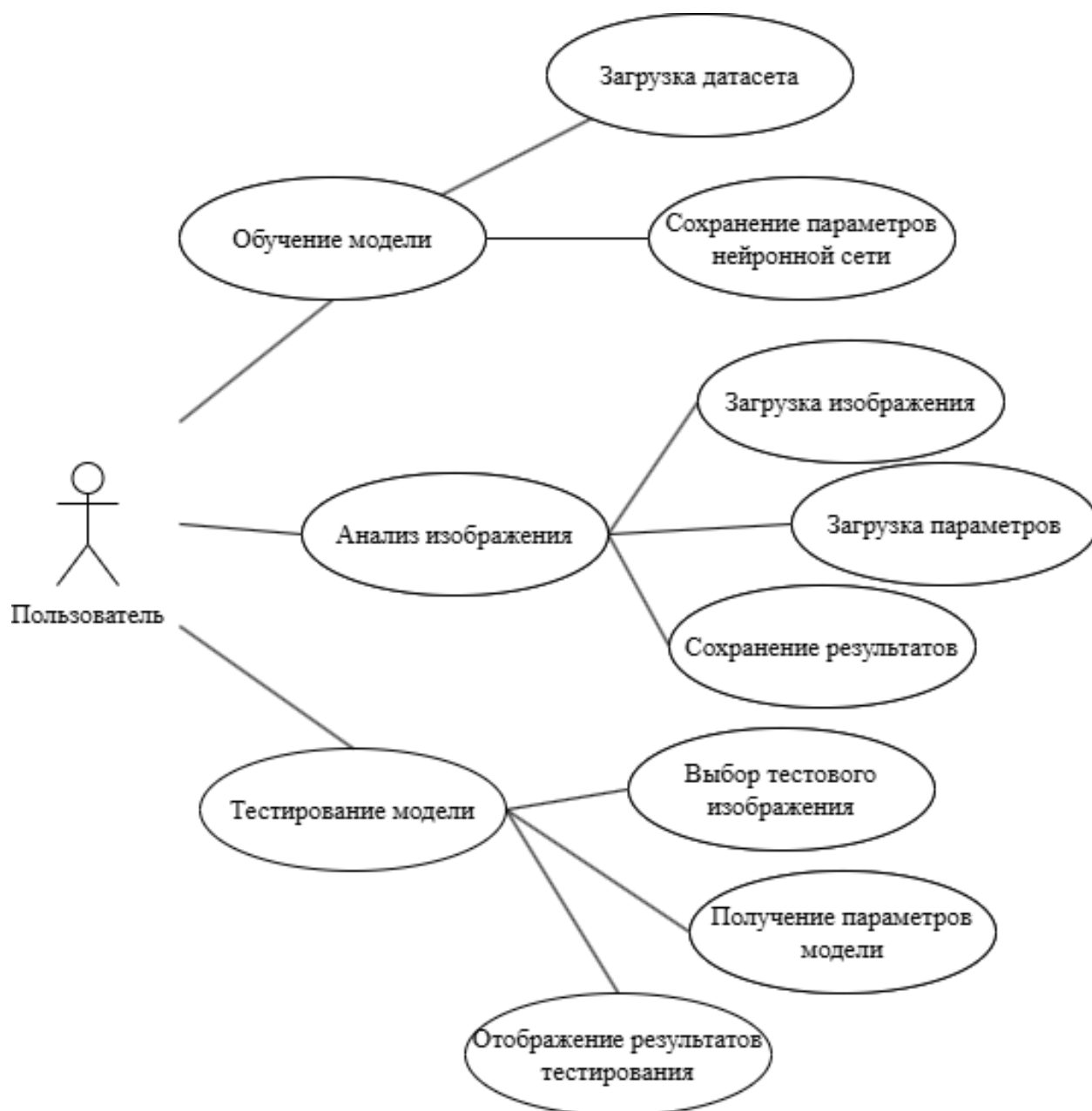


Рисунок 2.1 – Диаграмма прецедентов

2.3.2.1 Сценарий использования «Обучение нейронной сети»

Заинтересованные лица и их требования: пользователь желает обучить модель нейронной сети для распознавания нефтяных пятен на поверхности водоемов.

Предусловие: программа запущена, выбран режим «Обучение».

Постусловие: программа сохраняет файл, содержащий параметры весов нейронной сети.

Основной успешный сценарий:

1. Пользователь нажимает на кнопку «Выбрать папку» .
2. Программа открывает диалоговое окно выбора папки.
3. Пользователь выбирает папку, содержащую изображения для обучения нейронной сети.
4. Программа отображает путь до выбранной папки в специальном поле.
5. Пользователь нажимает на кнопку «Выбрать путь сохранения».
6. Программа открывает диалоговое окно выбора папки сохранения.
7. Пользователь выбирает директорию, в которой будет сохранен файл, содержащий параметры нейронной сети.
8. Программа отображает путь до выбранной директории сохранения в специальном поле.
9. Пользователь нажимает на кнопку «Обучить нейросеть».
10. Программа начинает процесс обучения нейронной сети на полученных изображениях.
11. Программа отображает прогресс процесса обучения пользователю.
12. Программа сохраняет полученный файл, содержащий параметры нейронной сети в папку, указанную пользователем, после завершения процесса обучения.

2.3.2.2 Сценарий использования «Тестирование нейронной сети»

Заинтересованные лица и их требования: пользователь желает протестировать работу предварительно обученную нейронную сеть и узнать качество сегментации.

Предусловие: программа запущена, выбран режим «Тестирование», имеется файл с настройками нейронной сети.

Постусловие: программа отображает метрики качества сегментации, а также сравнение ожидаемых и полученных результатов.

Основной успешный сценарий:

1. Пользователь нажимает на кнопку «Выбрать изображение».

2. Программа открывает диалоговое окно выбора изображения, используемого для тестирования нейронной сети.
3. Пользователь выбирает изображение.
4. Программа отображает путь до выбранного изображения в специальном поле.
5. Пользователь нажимает на кнопку «Выбрать настройки».
6. Программа открывает диалоговое окно выбора файла настроек нейронной сети.
7. Пользователь выбирает файл настроек нейронной сети.
8. Программа отображает путь до выбранного файла настроек в специальном поле.
9. Пользователь нажимает на кнопку «Запустить тестирование».
10. Программа выполняет предварительную обработку изображения.
11. Программа создает ожидаемую маску на основании загруженного изображения при помощи порогового алгоритма.
12. Программа обрабатывает обработанное изображение, используя нейронную сеть с загруженными параметрами.
13. Программа рассчитывает метрики качества сегментации изображения.
14. Программа отображает ожидаемую и полученную нейронной сетью маски классов, а также метрики качества сегментации.

2.3.2.3 Сценарий использования «Анализ изображения»

Заинтересованные лица и их требования: пользователь желает проанализировать изображение на наличие нефтяного разлива.

Предусловие: программа запущена, выбран режим «Анализ изображения», имеется файл с настройками нейронной сети.

Постусловие: программа отображает результаты анализа изображения.

Основной успешный сценарий:

1. Пользователь нажимает на кнопку «Выбрать изображение».

2. Программа открывает диалоговое окно выбора анализируемого изображения.
3. Пользователь выбирает необходимое изображение.
4. Программа отображает путь до выбранного изображения в специальном поле и выводит его в главное окно.
5. Пользователь нажимает на кнопку «Выбрать модель».
6. Программа открывает диалоговое окно выбора файла с настройками модели.
7. Пользователь выбирает файл настроек нейронной сети.
8. Программа отображает путь до выбранного файла в специальном поле.
9. Пользователь нажимает на кнопку «Анализировать».
10. Программа выполняет предварительную обработку изображения.
11. Программа анализирует обработанное изображение на предмет наличия пятен нефтяных разливов с помощью нейронной сети.
12. Программа отображает результат анализа на исходной версии изображения, загруженной пользователем в главном окне программы.

2.3.2.4 Сценарий использования «Сохранение результата анализа»

Заинтересованные лица и их требования: пользователь желает сохранить результат анализа изображения нейронной сетью.

Предусловие: программа запущена, выбран режим «Анализ изображения», изображение предварительно загружено и проанализированно.

Постусловие: программа сохраняет результат обработки в виде изображения на жесткий диск.

Основной успешный сценарий:

1. Пользователь нажимает на кнопку «Сохранить результат».
2. Программа открывает диалоговое окно выбора папки для сохранения результата анализа.
3. Пользователь выбирает необходимую папку.

4. Программа сохраняет результат анализа в указанную папку в виде изображения.

5. Программа уведомляет пользователя об успешном сохранении изображения.

2.3.3 Требования пользователя к интерфейсу приложения

Приложение должно иметь следующие экраны:

1. Экран «Анализ изображения». Основной экран, реализующий функционал распознавания пятен нефти на изображениях поверхности водоемов, где реализована возможность загрузки изображения для поиска нефтяных пятен, просмотр результата работы нейронной сети и его сохранения на жесткий диск.

2. Экран «Обучение». Экран, позволяющий обучать нейронную сеть искать пятна разливов нефти на изображениях. Должен содержать возможность выбора папки, содержащей датасет для обучения, и директории сохранения файла, содержащего настройки параметров нейронной сети, полученные во время ее обучения.

3. Экран «Тестирование». Экран для тестирования обученной модели, позволяющий выбрать файл настроек нейронной сети, тестовое изображение, а также отображающий исходное изображение, ожидаемый результат и фактический результат обработки тестового изображения нейронной сетью.

2.4 Нефункциональные требования к программной системе

2.4.1 Требования к надежности

Программная система должна обеспечивать стабильную работу в различных условиях эксплуатации. В процессе работы приложения могут возникнуть следующие аварийные ситуации:

- отсутствие файлов изображений в папке, выбранной как директория хранения датапака;

- ошибки загрузки или сохранения файлов настроек моделей или изображений из-за проблем с файловой системой;
- ошибки в работе программы из-за загрузки пользователем файлов некорректных форматов.

Для предотвращения аварийных ситуаций программа должна корректно обрабатывать исключения при работе с файлами, предоставляя пользователям информативные сообщения об ошибках. В случае проблем с отсутствием прав доступа к директории сохранения файлов, полученных в результате работы программы, программа должна открывать диалоговое окно с выбором другой директории.

2.4.2 Требования к аппаратному обеспечению

Для корректной работы программного продукта требуется центральный процессор с количеством ядер от 6 и выше с частотой ядра от 2.4 ГГц. Размер необходимой оперативной памяти - 8 Гб и выше. Кроме того, для отрисовки графического интерфейса требуется видеокарта с объемом графической памяти 4 Гб и выше, монитор. Наконец, для управления программой необходимы клавиатура и мышь.

Для ускорения процесса обучения нейронной сети возможно использование вычислительных ресурсов графического адаптера NVIDIA с поддержкой технологии CUDA.

2.4.3 Требования к программному обеспечению

Для запуска и работы программы требуется компьютер под управлением операционной системы Windows 10 или Windows 11. При использовании графических адаптеров NVIDIA с поддержкой технологии CUDA для ускорения обучения нейронной сети необходима последняя версия драйверов соответствующего адаптера, а также программы CUDA Toolkit и cuDNN.

2.5 Требования к оформлению документации

Требования к стадиям разработки программ и программной документации для вычислительных машин, комплексов и систем независимо от их назначения и области применения, этапам и содержанию работ устанавливаются ГОСТ 19.102–77. и ГОСТ 34.601-90.

Программная документация должна включать в себя:

- анализ предметной области;
- техническое задания;
- технический проект;
- рабочий проект.

3 Технический проект

3.1 Общая характеристика организации решения задачи

Необходимо спроектировать и разработать интеллектуальную систему распознавания пятен разливов нефти на поверхности водоемов.

Интеллектуальная система состоит из модели нейронной сети и приложения, отвечающего за взаимодействие пользователей с системой. В приложении необходимо иметь возможность взаимодействовать с нейронной сетью при помощи графического интерфейса пользователя. Под взаимодействием понимается загрузка изображений, содержащих пятна нефтяных разливов, возможность распознавания пятен при помощи модели нейронной сети, а также её обучение и тестирование.

3.2 Обоснование выбора технологии проектирования

Технологии, языки программирования и архитектурные решения, использованные для создания интеллектуальной системы, отвечают современным практикам разработки и позволяют достичь высокой производительности и отказоустойчивости программы.

3.2.1 Язык программирования Python

Для реализации программной системы был использован язык Python. Python – язык программирования высокого уровня, обладающий высокой степенью гибкости и широко используемый как в научной отрасли, так и для коммерческой разработки. Особенно хорошо Python подходит для построения интеллектуальных систем и анализа изображений благодаря богатой системе внешних библиотек, позволяющих значительно упростить построение модели нейронной сети, подготовку данных, работу с изображениями, создание графического интерфейса пользователя. Интуитивный синтаксис языка позволяет снизить количество синтаксических ошибок, что в свою очередь значительно ускоряет разработку программных продуктов. Встроенная поддержка различных парадигм программирования позволяет удобно

реализовывать модульную архитектуру приложения с использованием классов, функций и последовательного кода. Кроссплатформенность языка облегчает адаптацию системы для работы на различных операционных системах, что при желании позволит легко создать нативные версии приложений для таких операционных систем, как Linux и macOS.

3.2.2 Описание библиотеки PyTorch

Для разработки модели нейронной сети была выбрана библиотека PyTorch. PyTorch является открытой библиотекой, содержащей широкий набор инструментов для машинного обучения. Данная библиотека активно применяется в разнообразных научных исследованиях, использующих модели нейронных сетей. PyTorch зарекомендовала себя благодаря гибкости, лаконичному синтаксису и обширным возможностям построения и обучения нейронных сетей.

Для определения структуры вычислений PyTorch использует динамический вычислительный граф, что означает построение графа операций непосредственно в процессе выполнения программы вместо предварительного определения его структуры. Данный подход значительно упрощает модификацию и отладку архитектуры модели нейронной сети, позволяет удобно реализовывать операции обучения, тестирования и анализа при помощи полученной модели нейронной сети.

3.2.3 Описание библиотеки OpenCV

OpenCV – одна из наиболее часто применяемых библиотек в области компьютерного зрения и обработки изображений. Она позволяет реализовать значительное количество операций для работы с изображениями и видео, таких как фильтрация, преобразования, геометрические операции. Высокая производительность позволяет эффективно читать, изменять размеры и форматы изображений, создавать их маски, визуализировать результаты работы интеллектуальной системы.

3.2.4 Описание фреймворка PyQt6

PyQt6 является набором расширений кроссплатформенного графического фреймворка Qt 6 версии для языка Python. PyQt6 позволяет создавать графические интерфейсы пользователя различной сложности – от простых оконных приложений до многооконных систем с развитой логикой взаимодействий.

Одним из ключевых преимуществ PyQt6 является большое число готовых элементов графического интерфейса, таких как кнопки, выпадающие списки, таблицы, позволяющих строить полноценный интуитивный пользовательский интерфейс. Графический интерфейс PyQt можно разрабатывать как вручную в коде, так и при помощи визуального редактора Qt Designer, возвращающего пользователю готовый Python-код для графического интерфейса. Кроссплатформенность PyQt6 позволяет запускать приложения, разработанные для операционной системы Windows на компьютерах с Linux и macOS не требуя значительных изменений, что позволяет без особых усилий создавать мультиплатформенные версии приложений.

3.2.5 Описание библиотеки NumPy

Библиотека NumPy предназначена для работы с многомерными массивами и матрицами, а также позволяет проводить сложные математические вычисления с высокой скоростью. NumPy предлагает большое количество функций для работы с массивами, таких как арифметические и логические операции, линейная алгебра.

NumPy позволяет преобразовывать изображения и их маски в числовые массивы для дальнейшей обработки, а также объединять и нормализовать эти маски. Компактный синтаксис и высокая производительность делают его подходящим для практически любых проектов, связанных с обработкой данных.

3.3 Архитектура программной системы

На рисунке 3.1 в виде UML-диаграммы представлена архитектура программной системы.

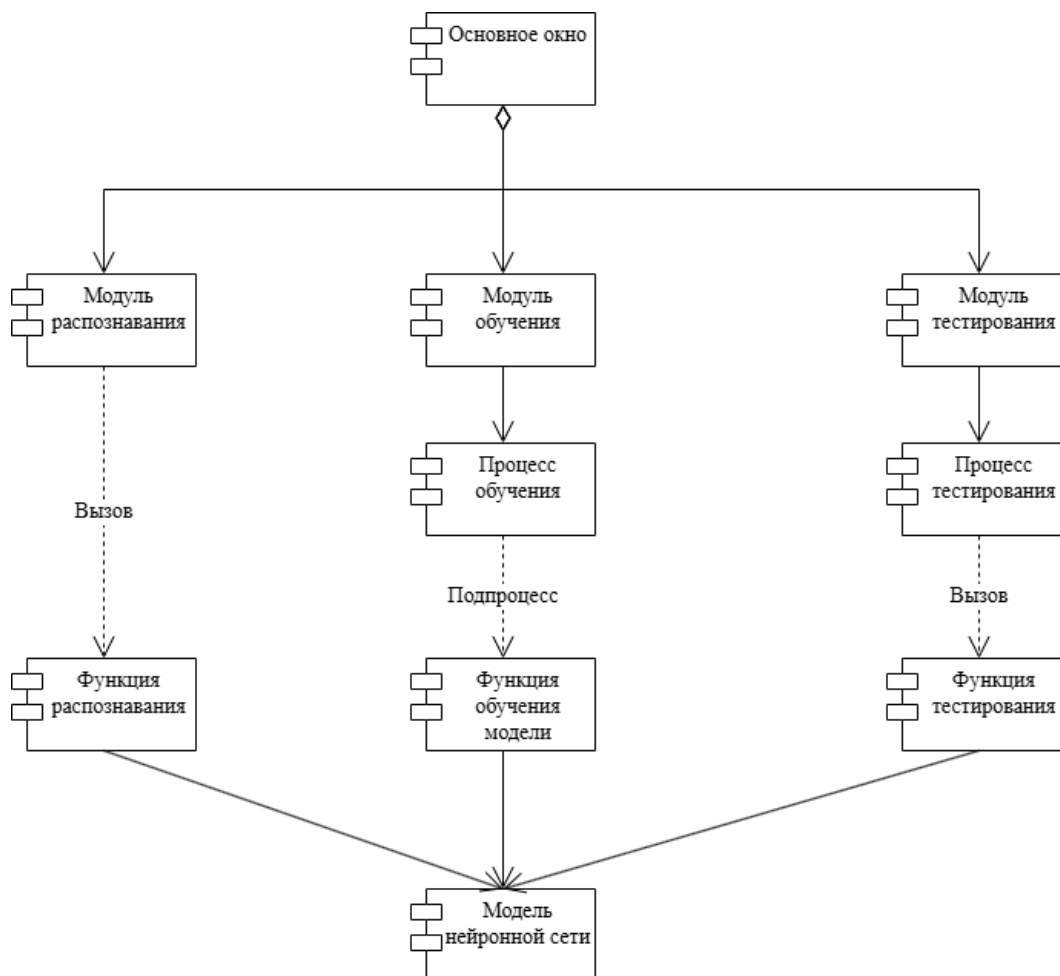


Рисунок 3.1 – Архитектура системы

Интеллектуальная система состоит из следующих компонентов:

1. Основное окно. Основное окно отвечает за вызов главного окна программной системы и инициализацию трех режимов работы программы и переключение между ними.
2. Модуль распознавания. Данный модуль вызывает окно для распознавания изображения, позволяющее выбрать изображение и параметры модели для анализа нейронной сетью, запускает процесс распознавания и отображает результат.

3. Функция распознавания. Использует модель нейронной сети и ее параметры для анализа полученного изображения, возвращая результат работы и сохраняя его в виде файла.

4. Модуль обучения. Вызывает окно, позволяющее выбрать директорию, содержащую изображения для обучения нейронной сети, и указать директорию для сохранения файла параметров обученной нейронной сети. Кроме того, данный модуль запускает процесс, отвечающий за обучение нейронной сети и отображение прогресса завершения обучения.

5. Функция обучения модели. Компонент загружает изображения для обучения, преобразует их в требуемый для обучения нейронной сети вид и обучает модель, сохраняя полученный в результате файл параметров.

6. Модуль тестирования. Вызывает окно, позволяющее выбрать параметры модели, тестовое изображение и запускает процесс, отвечающий за тестирование модели нейронной сети с сохраненными параметрами и возвращает результаты тестирования. Также отображает полученные метрики и результаты.

7. Функция тестирования. Выполняет оценку точности модели нейронной сети с параметрами, загруженными из предварительно сохраненного файла. Сравнивает результат анализа изображения нейронной сетью и ожидаемый результат, полученный при помощи пороговой фильтрации.

8. Модель нейронной сети. Хранит в себе структуру используемой в системе нейронной сети.

3.4 Описание нейронной сети

Для распознавания нефтяных пятен на изображениях поверхностей водоемов была спроектирована и разработана модель сверточной нейронной сети. Сверточные нейронные сети – однонаправленные модели, разработанные для распознавания образов на изображениях.

3.4.1 Метод распознавания пятен нефти

Для определения пятен нефти на изображениях сверточные нейронные сети могут применяться в соответствии с одним из следующих подходов:

- распознавание объектов на изображении;
- семантическая сегментация.

Распознавание объектов обучает нейронную сеть распознавать образы, выделяя их при помощи ограничивающих рамок прямоугольной области, окружающих распознанный объект. Нейронная сеть, обученная при помощи семантической сегментации, выполняет попиксельную разметку, присваивая каждому пикселю метку определенного класса. В отличие от распознавания объектов, семантическая сегментация позволяет снизить погрешность распознавания и повысить точность локализации, минимизируя возможность ложного распознавания сторонних объектов, окружающих нефтяные разливы, зачастую представляющие собой пятна неправильной формы.

Для создания интеллектуальной системы был выбран семантический подход сегментации, основанный на распознавании определенных уровней яркости пикселей. В большинстве случаев при переводе изображения, содержащего разлив нефти, в полутоновое, пятно будет изображено скоплениями темных пикселей. Обучив нейронную сеть распознавать пиксели, яркость которых ниже определенного значения, можно эффективно обнаруживать и выделять нефтяные разливы на изображении.

3.4.2 Архитектура U-Net

В данной работе была реализована нейронная сеть типа U-Net, адаптированная под особенности предметной области.

Архитектура U-Net была представлена в 2015 году группой исследователей из Университета Фрайбурга. Данная архитектура получила широкое распространение благодаря способности выделять объекты различной формы и масштаба с высокой точностью даже на маленьких обучающих выборках. Основной особенностью U-Net является симметричная структура: левая

часть сети, или энкодер, выполняет последовательное сжатие входного изображения, а правая, или декодер – восстановление пространственного разрешения до исходного размера.

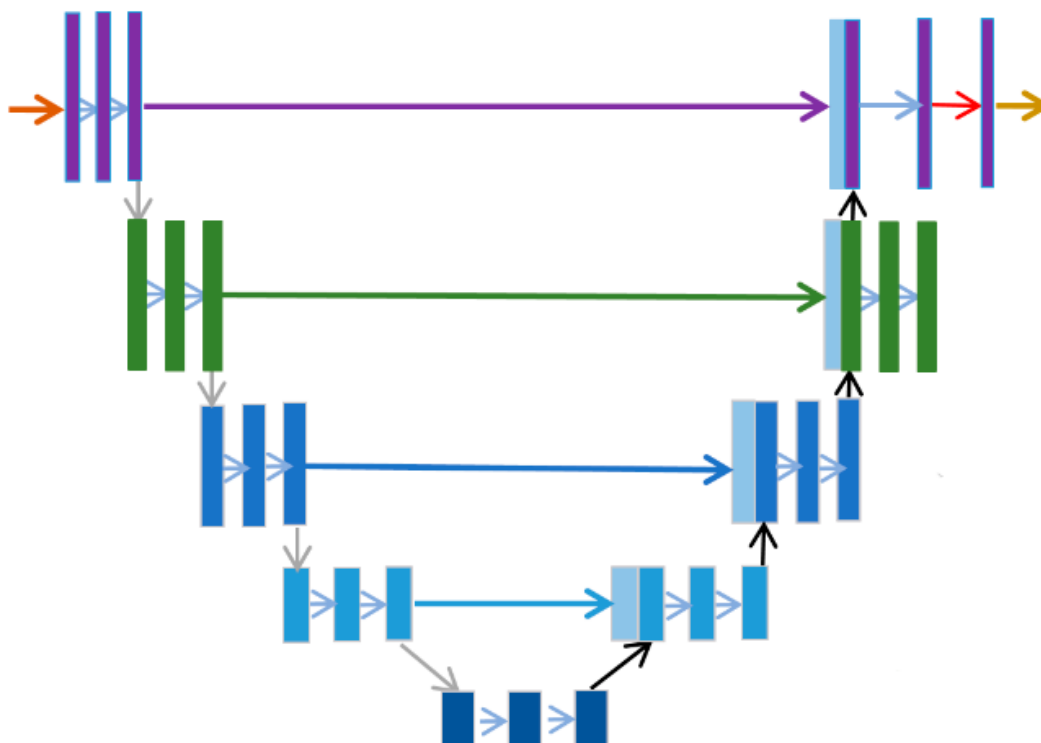


Рисунок 3.2 – Архитектура сети U-Net

Ключевым элементом U-Net являются прямые соединения между соответствующими уровнями энкодера и декодера, позволяющие использовать признаки, извлеченные на ранних этапах обработки, передавая их на этапы восстановления изображения, что предотвращает потерю пространственной информации. Данные особенности позволяют с легкостью адаптировать U-Net для использования в различных областях, например медицинская диагностика, спутниковый мониторинг, обработка изображений, полученных с беспилотных летательных аппаратов.

3.4.3 Структура нейронной сети

Реализация модели U-Net состоит из энкодера, «бутылочного горлышка» (bottleneck) и декодера. Энкодер состоит из двух уровней, содержащих по два сверточных слоя, преобразующих 1 канал в 64 и 64 канала в 128, создавая

64 и 128 карт признаков соответственно, каждая из которых отвечает за определенные особенности изображения, такие как горизонтальные и вертикальные края, их комбинации, текстуры. Каждый из сверточных слоев, используемых в уровнях, имеет размерность маски 3×3 , проходящей по изображению с шагом 1 (маска смещается на 1 пиксель) и извлекающей вышеуказанные признаки с сохранением размеров изображения. После извлечения энкодер уменьшает изображение в 2 раза, используя маску размерностью 2×2 , смещающуюся на 2 пикселя за шаг. Операция уменьшения повторяется после каждого извлечения признаков, в общем уменьшая размеры изображения в 4 раза.

Слой «бутылочного горлышка» продолжает процесс извлечения признаков, распознавая абстрактные признаки, такие как крупные структуры и формы, добавляя дополнительные 128 каналов до общей суммы 256, перемещая маску размерности 3×3 шагом в 1. Размер самого изображения при этом остается неизменным.

Слой декодера возвращает изображению исходные размеры. При помощи перемещения маски размерностью 2×2 с шагом 2 увеличивает размер полученного из предыдущего этапа изображения в 2 раза, растягивая каждый пиксель в блок размером 2×2 и корректируя значения при помощи весов, вместе с этим комбинируя данные признаков, уменьшая размерность каналов в 2 раза до 128. Затем слой уточняет признаки, полученные из предыдущего шага и признаки, полученные энкодером на 2 шаге свертки, и возвращает изображение с размерностью, полученной на предыдущем шаге и 256 картами признаков, содержащую признаки как высокого, так и низкого уровней. Наконец, сверточный блок использует маску 3×3 с шагом 1 и уточняет карты признаков высокого уровня из «бутылочного горлышка» и низкого уровня из энкодера, тем самым уменьшая количество каналов в два раза, до 128. Данный процесс повторяется два раза, по одному на каждый уровень декодера, восстанавливая исходные размеры изображения и объединяя карты признаков. Финальный слой объединяет оставшиеся 64 карты признаков в единую маску, значения которой обозначают вероятность принадлежности пикселя к

классу разлива, которая, после применения порога функцией активации, изменяется в бинарную маску (1, если пиксель является частью разлива и 0, если нет).

3.4.4 Обучение нейронной сети

Обучение модели нейронной сети происходит на наборе полутоновых изображений формата JPEG разрешения 624x320. Целевые маски для обучения создаются на основании яркостных признаков при помощи алгоритма бинаризации с порогом 0,5. Размер целевых масок совпадает с размером входного изображения. Датасет загружается из отдельной директории и полностью используется для обучения без разделения на обучающую, валидационную и тестовую выборки.

Обучение происходит на протяжении 25 эпох на группах из двух изображений. Дополнительные аугментации не применяются. Обучение проводится на центральном процессоре, при этом реализована возможность обучения на графическом процессоре при наличии поддерживаемого устройства. На каждой эпохе для каждой группы изображений входные данные и маски отправляются на вычислительное устройство, модель делает предсказание, после чего вычисляется потеря по формуле:

$$L = - [y \cdot \log(p) + (1 - y) \cdot \log(1 - p)] ,$$

где y – идеальное значение, p – предсказанная вероятность. Наконец, градиенты обнуляются, вычисляются заново, после чего веса модели обновляются.

3.5 Проектирование пользовательского интерфейса

На основании требований к пользовательскому интерфейсу, представленных в пункте 2.3.3, был разработан графический интерфейс программной системы.

На рисунке 3.3 представлен макет интерфейса окна «Анализ изображения». Макет содержит следующие элементы:

1. Кнопка переключения режимов окна.

2. Поле, содержащее путь до анализируемого изображения.
3. Кнопка выбора изображения для анализа.
4. Поле, содержащее путь до выбранного файла параметров нейронной сети.
5. Кнопка выбора файла параметров нейронной сети.
6. Поле для отображения загруженного изображения и результатов анализа.
7. Кнопка для запуска распознавания нефтяных пятен.
8. Кнопка для сохранения результатов анализа нейронной сети.

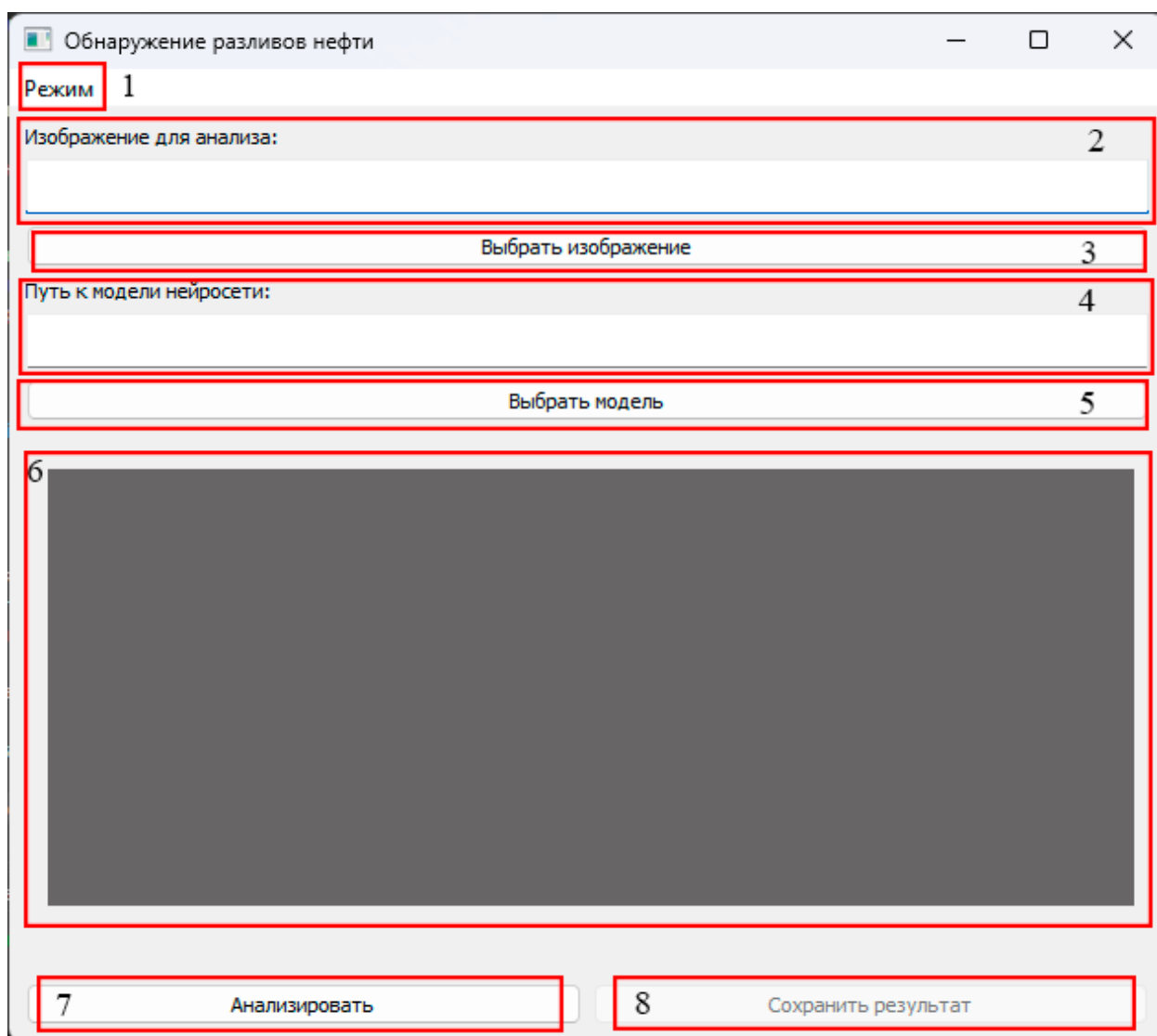


Рисунок 3.3 – Макет интерфейса окна «Анализ изображения»

На рисунке 3.4 представлен макет интерфейса окна «Обучение». Макет содержит следующие элементы:

1. Кнопка переключения режимов окна.
2. Поле, содержащее путь до выбранной папки с изображениями для обучения нейронной сети.
3. Кнопка выбора папки с изображениями для обучения нейронной сети.
4. Поле, содержащее путь до папки, в которую необходимо сохранить параметры модели.
5. Кнопка для выбора папки, в которую необходимо сохранить параметры модели.
6. Окно для вывода прогресса обучения нейронной сети.
7. Кнопка для запуска процесса обучения нейронной системы.

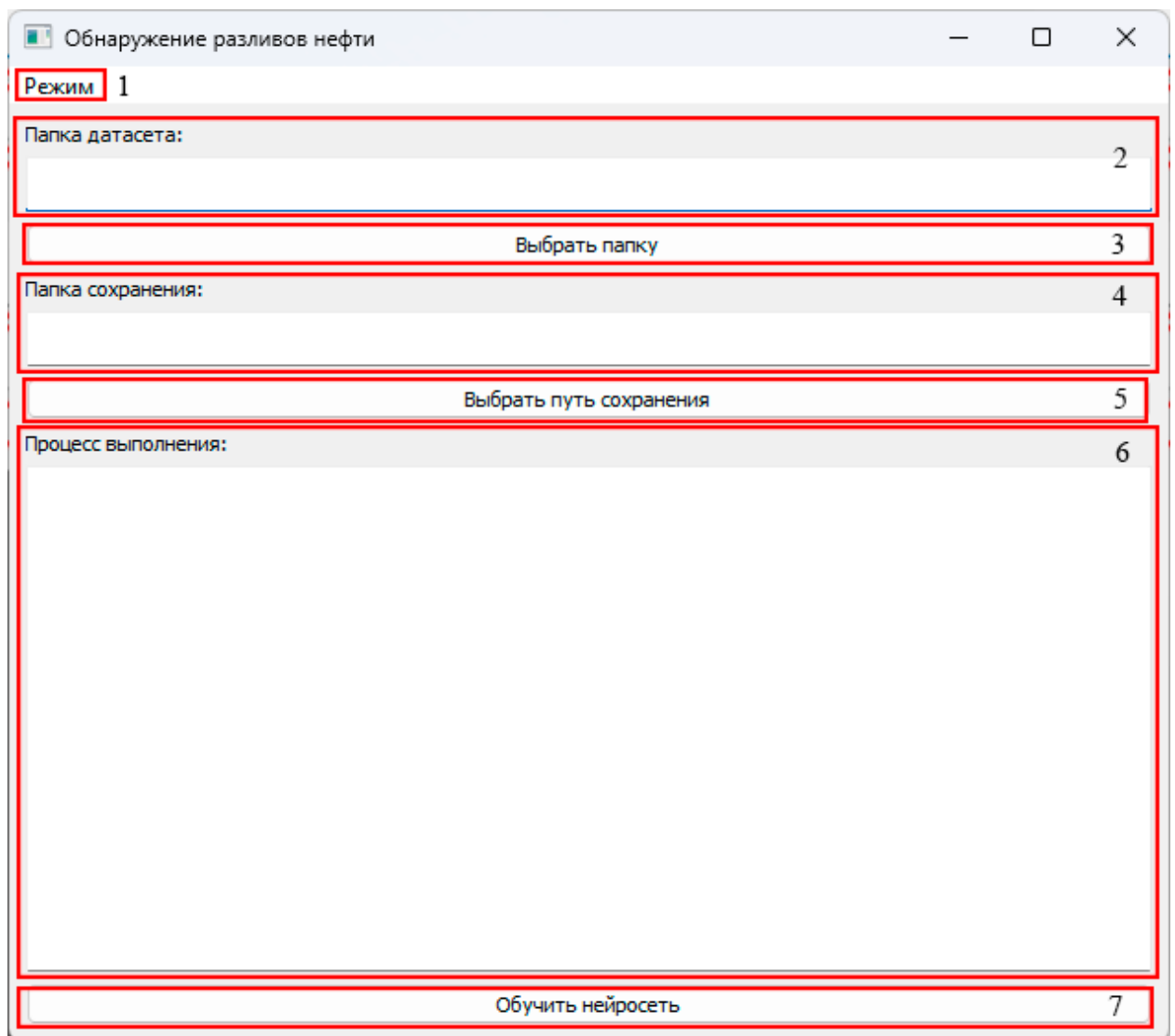


Рисунок 3.4 – Макет интерфейса окна «Обучение»

На рисунке 3.5 представлен макет интерфейса окна «Тестирование». Макет содержит следующие элементы:

1. Кнопка переключения режимов окна.
2. Поле, содержащее путь до тестового изображения.
3. Кнопка выбора изображения для тестирования нейронной сети.
4. Поле, содержащее путь до выбранного файла параметров нейронной сети.
5. Кнопка выбора файла параметров нейронной сети.
6. Поле вывода метрик тестирования нейронной сети.
7. Поле для отображения исходного изображения.
8. Поле для отображения ожидаемого результата обработки.
9. Поле для отображения фактического результата обработки изображения нейронной сети.
10. Кнопка для запуска тестирования.

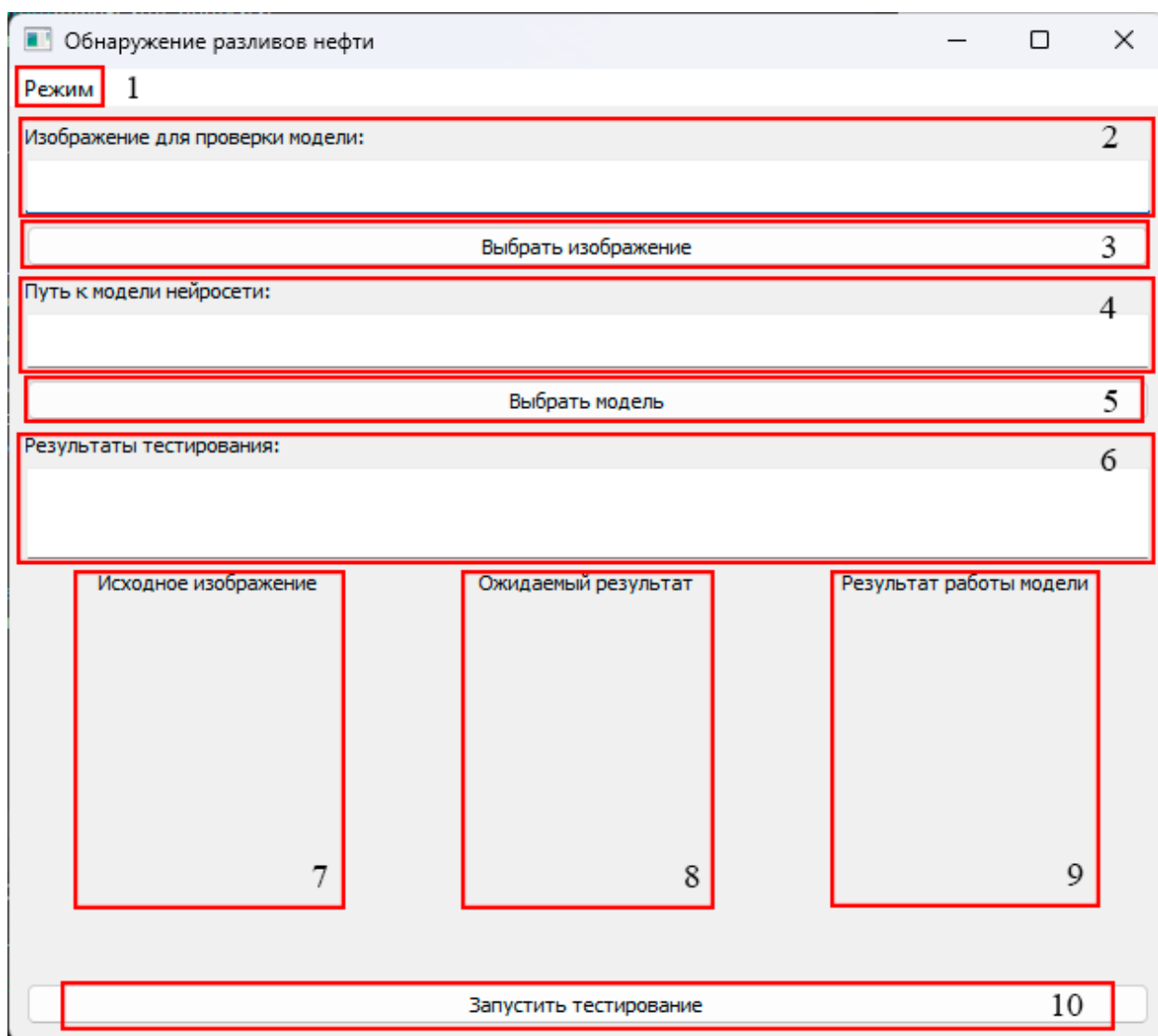


Рисунок 3.5 – Макет интерфейса окна «Тестирование»

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Алексеев Д. В. Сравнительный анализ баз данных по разливам нефти и нефтепродуктов с морских судов / Д. В. Алексеев, А. А. Лентарёв // Вестник Государственного университета морского и речного флота имени адмирала С. О. Макарова. — 2022. — Т. 14. — № 6. — С. 891–904. DOI: 10.21821/2309-5180-2022-14-6-891-904. — Текст: непосредственный.

2. Владимиров В. А. Разливы нефти: причины, масштабы, последствия // Стратегия гражданской защиты: проблемы и исследования. — 2014. — № 1. — URL: <https://cyberleninka.ru/article/n/razlivy-nefti-prichiny-masshtaby-posledstviya> (дата обращения: 10.05.2025). — Текст : непосредственный.

3. Клименко С. К., Иванов А. Ю., Терелева Н. В. Пленочные загрязнения Керченского пролива по данным пятилетнего радиолокационного мониторинга: современное состояние и основные источники // Исследование Земли из космоса. — 2022. — № 3. — С. 37–54. — DOI: 10.31857/S0205961422030071. — Текст : непосредственный.

4. Горбачевская Е. Н., Краснов С. С. История развития нейронных сетей // Вестник ВУиТ. — 2015. — № 1 (23). — URL: <https://cyberleninka.ru/article/n/istoriya-razvitiya-neyronnyh-setey> (дата обращения: 11.05.2025). — Текст : непосредственный.

5. Митина О. А., Ломовцев П. П. Перцептрон в задачах бинарной классификации // НАУ. — 2021. — № 66-1. — URL: <https://cyberleninka.ru/article/n/pertseptron-v-zadachah-binarnoy-klassifikatsii> (дата обращения: 11.05.2025). — Текст : непосредственный.