Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»
(ФГАОУ ВО НИУ МФТИ)

ОТЧЕТ О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

по теме:

«Изучение возможностей искусственного интеллекта для автоматизированного выявления признаков нарушений законодательства РФ в изображениях и видеоматериалах»

(шифр «НИР АС ОКУЛУС»)

Концепция АС ОКУЛУС

Руководитель НИР,
Заведующий лабораторией бизнес-решений на основе ИИ

_____ Д.С. Величкин
«___»_____ 2021 г.

Долгопрудный

2021

Содержание

1.Основания разработки Концепции

Концепция развития автоматизированной системы «Окулус» (далее – АС ОКУЛУС) разработана в соответствии с проектом стратегического развития № 9 «Выявление нарушений в изображениях и видео», в рамках которого проводятся научно-исследовательские работы по выявлению нарушений в изображениях и видеоматериалах. Разработка Концепции необходима для определения требований к последующим этапам проектирования и внедрения АС ОКУЛУС, автоматизирующей процессы распознавания запрещенной информации в изображениях и видеоматериалах для выполнения требований Федерального законодательства:

- (Федерального закона от 27.07.2006 № 149-ФЗ «Об информации, информационных технологиях и о защите информации»,
- Федерального закона от 25 июля 2002 г. № 114-ФЗ «О противодействии экстремистской деятельности»,
- Федерального закона от 29.12.2010 № 436-ФЗ «О защите детей от информации, причиняющей вред их здоровью и (или) развитию».

2.Общие положения

Концепция развития АС ОКУЛУС определяет направления и стратегию развития программно-аппаратной архитектуры АС ОКУЛУС, обеспечивающей необходимые и достаточные технические условия для эффективной реализации стратегических планов по борьбе с правонарушениями в медийном пространстве Российской Федерации в части, касающейся выявления противоправного контента в сети Интернет.

Концепция АС ОКУЛУС определяет организацию системы в части определения её элементов и их взаимоотношений друг с другом и со средой.

При разработке Концепции АС ОКУЛУС учтены ее следующие основные технологические свойства:

1. Модульный принцип построения системы.
   Масштабируемость комплекса технических средств АС ОКУЛУС.
2. Масштабируемость программных изделий АС ОКУЛУС.
3. Возможности интеграции с существующими и планируемыми в дальнейшем к разработке сервисами Заказчика с использованием API.
4. Высокая гибкость настройки конечной системы.
5. Размещение серверных мощностей в ЦОД с соблюдением максимального уровня беспрерывности доступа; соблюдение актуальных требований к уровню информационной безопасности; расположение всей облачной инфраструктуры на территории РФ.

3.Описание предметной области и объекта автоматизации

В настоящее время выявление нарушений в изображениях и видеоматериалах осуществляется в ручном режиме. Создание автоматизированной системы на основе искусственного интеллекта распознавания образов и событий в видеоконтенте позволит автоматизировать и ускорить процесс инспектирования информационного пространства. В рамках данной Концепции предлагаются функциональные, программные и аппаратные решения, позволяющие автоматизировать процесс распознавания запрещенной информации в изображениях и видеоматериалах.

Объектом автоматизации являются процессы выявления признаков нарушений законодательства Российской Федерации в изображениях и видеоматериалах (текст, символика, «водяные знаки», сочетания предметов, композиция образов, статика и динамика движений) для выполнения требований Федерального закона от 27.07.2006 № 149-ФЗ «Об информации, информационных технологиях и о защите информации», Федерального закона от 25 июля 2002 г. № 114-ФЗ «О противодействии экстремистской деятельности», Федерального закона от 29.12.2010 № 436-ФЗ «О защите детей от информации, причиняющей вред их здоровью и (или) развитию».

4.Предпосылки создания АС

Предпосылками создания АС ОКУЛУС являются:

1. Стремительное нарастание угроз информационного противодействия в медийном пространстве Российской Федерации, которое направлено на формирование … , разложение … , дезинтеграцию общества… , создание предпосылок для преступной деятельности…
2. Резкое увеличение объемов информационных материалов, содержащих …, что не позволяет оперативно реагировать на …
3. Усложнение методов сокрытия и маскировки противоправного видеоконтента в информационных материалах …
4. Необходимость исполнения требований ФЗ в сферах …
5. Совершенствование и развитие методов искусственного интеллекта, что позволяет на современном этапе его развития применять его элементы для решения задач …

5.Цели и задачи создания АС

Целями создания АС ОКУЛУС являются

- повышение оперативности выявления противоправного контента в информационных материалах;
- повышение достоверности оценок несоответствия выявленного контента законодательству РФ;
- автоматизация распознавания признаков запрещенной информации в изображениях и видеоматериалах (текст, символика, «водяные знаки», сочетания предметов, композиция образов, статика и динамика движений);
- увеличение количества объектов наблюдения;
- повышение эффективности процессов выявления признаков нарушений законодательства РФ в изображениях и видеоматериалах;
- оптимизация расходов на выявление признаков нарушений законодательства РФ в изображениях и видеоматериалах.

6.Архитектура и описание АС ОКУЛУС

В рамках Концепции функциональная схема АС ОКУЛУС представлена в двух вариантах (на момент утверждения технического задания на создание АС ОКУЛУС функциональная схема должна быть определена однозначно и зафиксирована в ТЗ на создание и работу АС на всех этапах жизненного цикла).

Под моделью ИИ понимается реализованный в программном коде в соответствующем фреймворке машинного обучения алгоритм выявления признаков ЗИ (текста и видеоизображений в материалах, распространяемых в сети Интернет) вместе с данными, необходимыми для его выполнения (конфигурационные данные для запуска и выполнения).

6.1.Функциональная архитектура АС ОКУЛУС (вариант 1)

Функциональными подсистемами АС ОКУЛУС являются:

- Подсистема интеграции,
- Подсистема обработки данных,
- Подсистема выполнения задач,
- Подсистема ML-операций,
- Подсистема внутренней отчетности,
- Подсистема хранения данных,
- Подсистема ведения нормативно-справочной информации,
- Подсистема администрирования.

Функциональная архитектура АС ОКУЛУС (вариант 1) представлена на рисунке 6.1.

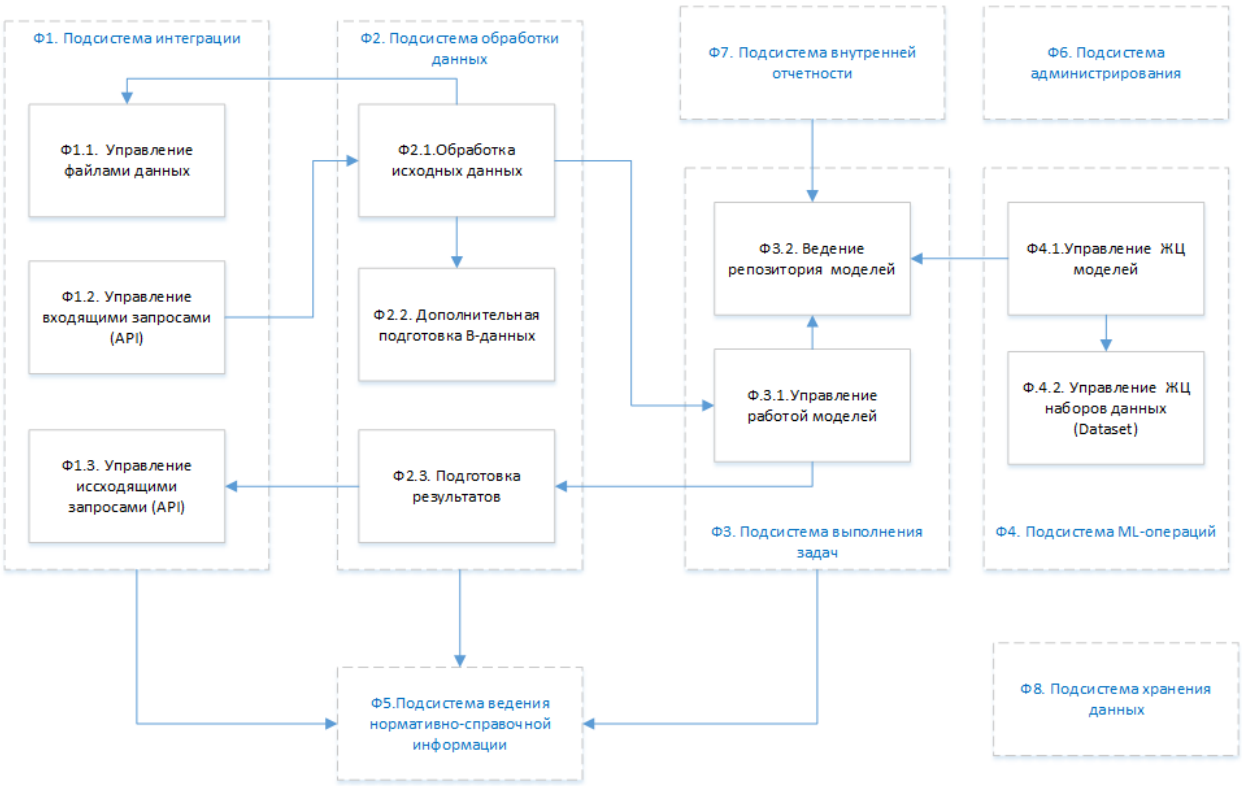

Рисунок 6.1 – Функциональная архитектура АС ОКУЛУС (вариант 1)

Входящие в АС ОКУЛУС функциональные подсистемы рассмотрены в подпунктах 6.1.1-6.1.8.

6.1.1.Подсистема интеграции

Подсистема интеграции предназначена для обеспечения информационного взаимодействия ИИ с внешними АС ОКУЛУС.

Подсистема интеграции включает в себя следующие функциональные модули:

6. управление файлами исходных данных,
7. управление входящими запросами,
8. управление исходящими запросами.

Описание функциональных модулей Подсистемы интеграции представлено в подпунктах 6.1.1.1-6.1.1.3.

6.1.1.1.Функциональный модуль «Управление файлами данных»

Функциональный модуль «Управление файлами данных» предназначен для получения и сохранения фото- и видео (ФВ)-файлов во внешнем корпоративном хранилище данных.

Модуль «Управление файлами данных» является сервисом АС ОКУЛУС, реализующим следующую функциональность:

1. доступ к внешнему хранилищу данных по Application programming interface (API) для копирования требуемого файла (набора файлов) и передачи копий файлов (набора файлов) Подсистеме обработки данных для дальнейшей обработки,
2. получение от Подсистемы обработки данных файла (набора файлов) и размещение этих данных во внешнем корпоративном хранилище данных – если это требуется (например, исходное фото с добавленной аннотацией).

Функциональный модуль «Управление файлами данных» реализуется с учётом следующих особенностей:

1. вызов функций модуля осуществляет Подсистема обработки данных,
9. приоритетным вариантом реализации модуля является использование протокола доступа к облачному хранилищу данных, подобного S3 API,
10. параметры подключения модуля к внешнему корпоративному хранилищу данных создаются и хранятся в Подсистеме администрирования АС ОКУЛУС.

6.1.1.2. Функциональный модуль «Управление входящими запросами»

Функциональный модуль «Управление входящими запросами» предоставляет интерфейс доступа к функциональности АС ОКУЛУС внешним системам.

Функциональный модуль «Управление входящими запросами» предназначен для получения и первичной обработки запросов со стороны внешних систем, а также инициирования дальнейших процессов обработки данных.

Модуль обрабатывает следующие виды запросов от внешних систем:

1. запросы на выявление ЗИ в указанном файле (наборе файлов),

2. запросы, содержащие корректировку результата ранее выполненного запроса выявление ЗИ. Такие запросы содержат ссылку (идентификатор) на исходный запрос и набор признаков ЗИ, по которым был неверно сделан вывод об отсутствии/наличии ЗИ.

Модуль «Управление входящими запросами» представляет собой сервис АС ОКУЛУС, реализующий следующую функциональность:

1. получение и разбор «тела» запроса (байтов данных, передаваемых в сообщении транзакции после заголовка),
11. первоначальная проверка корректности набора полученных данных и обработка исключительных ситуаций в случае их некорректности в соответствии с заданными правилами,
12. первичная классификация запросов,
13. упаковка разобранных данных во внутренние структуры данных в соответствии с видом запроса и передача их для дальнейшей обработки:
    - в Подсистему обработки данных в случае запроса на выявление ЗИ
    - в подсистему ML-операций – в случае запроса с корректировкой ранее полученного результата.

Функциональный модуль «Управление входящими запросами» реализуется с учётом следующих особенности: модуль «Управление входящими запросами» по завершении исполнения автоматически вызывает функциональность модуля «Обработка исходных данных» для передачи разобранных данных и последующего инициирования процесса выявлении признаков ЗИ («inference»).

6.1.1.3.Функциональный модуль «Управление исходящими запросами»

Функциональный модуль «Управление исходящими запросами» предназначен для передачи внешним потребителям уведомления о выполнении обработки входящего запроса с результатами классификации по каждому признаку ЗИ.

Модуль «Управление исходящими запросами» представляет собой сервис АС ОКУЛУС, реализующий следующую функциональность:

1. получение от вызывающей стороны (Подсистемы обработки данных) результатов классификации ФВ-данных,
14. формирование тела запроса,
15. вызов метода API соответствующего потребителя (внешней системы).

6.1.2.Подсистема обработки данных

Подсистема обработки данных предназначена для:

1. преобразования входных данных во внутренние структуры данных АС ОКУЛУС, необходимые для запуска и работы моделей классификации,
2. преобразования результатов работы моделей из внутренних структур данных в архитектура АС ОКУЛУСнеобходимой для формирования исходящего запроса части с уведомлением о выполнении задачи,
3. обеспечения целостности и связности входных и выходных данных.

Подсистема обработки данных включает в себя следующие функциональные модули:

1. модуль подготовки исходных данных,
2. модуль дополнительной подготовки видеоданных,
3. модуль подготовки результатов.

Описание функциональных модулей Подсистемы обработки данных представлено в подпунктах 6.1.2.1-6.1.2.3.

6.1.2.1.Функциональный модуль «Подготовка исходных данных»

Функциональный модуль «Подготовка исходных данных» предназначен для преобразования, категоризации и учёта данных, полученных в рамках выполнения функциональных модулей Подсистемы интеграции.

Модуль «Подготовка исходных данных» является внутренним сервисом АС ОКУЛУС, реализующим следующую функциональность:

1. преобразование структур, полученных из модуля «Управление файлами данных», в структуры данных, которыми оперируют модели (тензоры различных типов, объекты служебных классов, JSON-структуры при вызове смежных модулей),
16. формирование задачи на обработку данных – внутренней структуры данных АС ОКУЛУС, обеспечивающей взаимосвязь между входящим запросом, исходными данными, результатом вычислений и исходящим запросом. В рамках формирования задачи на обработку данных также реализуются следующие функции:
    - формирование уникального в рамках АС ОКУЛУС идентификатора задачи;
    - определение типа исходных данных (изображение или видеоматериал) и первичная классификация задачи;
    - определение набора выявляемых признаков ЗИ в зависимости от типа данных и информации, содержащейся во входном запросе.

Функциональный модуль «Подготовка исходных данных» реализуется с учётом следующих особенности: работа модуля инициируется вызовом из модуля «Управление входящими запросами» и завершается передачей структур данных в модуль «Управление задачами (управление выполнением задач)» в случае, если файл исходных данных, подлежащий классификации, содержит фотоизображение.

6.1.2.2.Функциональный модуль «Дополнительная обработка видеоданных»

Функциональный модуль «Дополнительная обработка B-данных» предназначен для дополнительной обработки видеоданных, необходимой для работы моделей.

Модуль «Дополнительная обработка B-данных» является внутренним сервисом АС ОКУЛУС, реализующим следующую функциональность:

1. кадрирование исходного видеофайла – формирование набора файлов, каждый из которых соответствует определенному кадру

видеоинформации. Параметры кадрирования задаются в настройках Подсистемы администрирования,

17. транскодирование – преобразование исходного потокового видеофайла из одного метода кодирования в другой в соответствии с действующими стандартами на кодеки. Формат файла (контейнер) при транскодированию изменению не подвергается. Параметры транскодирования задаются в настройках администрирования,
18. трансмультиплексирование – преобразование исходного видеофайла из одного формата (контейнера) в другой в соответствии с действующими стандартами на контейнеры. Параметры трансмультиплексирования задаются в настройках администрирования,
19. трансрейтинг – изменение битовой скорости потока данных, представленных в исходном видеофайле, либо создание другого представления,
20. трансайзинг – изменение разрешающей способности видеофайла.

Функциональный модуль «Дополнительная обработка B-данных» реализуется с учётом следующих особенностей:

21. приоритетным способом реализации модуля является использование библиотеки FFmpeg или подобных библиотек, поддерживающих программные интерфейсы доступа к функциональности,
22. модуль должен обеспечивать выполнение функций как в автоматизированном режиме при выполнении основного рабочего процесса выявления ЗИ («inference»), так и в «ручном» режиме через пользовательский интерфейс. В «ручном» режиме вызов модуля осуществляется в Подсистеме ML-операций для выполнения задач управления жизненным циклом наборов данных (datasets). В автоматизированном режиме функциональность модуля вызывается из модуля «Подготовка исходных данных».

6.1.2.3.Функциональный модуль «Подготовка результатов»

Функциональный модуль «Подготовка результатов» предназначен для форматирования классифицированных данных в целях дальнейшей обработки в Подсистеме выполнения задач.

Модуль «Подготовка результатов» является внутренним сервисом АС ОКУЛУС, реализующим следующую функциональность:

1. преобразование структур данных из формата, которым оперируют модели, в формат, пригодный для передачи результатов классификации внешней АС ОКУЛУС,
23. формирование структуры данных, представляющих результат вычислений,
24. наполнение задачи на обработку данных информацией о результатах выполнения задачи,
25. сохранение результатов выполнения данных в хранилище данных АС ОКУЛУС.

Функциональный модуль «Подготовка результатов» реализуется с учётом следующих особенностей:

1. модуль должен поддерживать обработку основных фреймворков машинного обучения,
26. выполнение функциональности модуля инициируется модулем «Управление работой моделей»,
27. работа модуля завершается вызовом модуля «Управление исходящими запросами» и передачей сформированной структуры данных.

6.1.3.Подсистема выполнения задач

Подсистема выполнения задач предназначена для непосредственного выполнения процесса выявления ЗИ моделями машинного обучения, реализованными в выбранном наборе библиотек.

Подсистема выполнения задач включает в себя следующие функциональные модули:

1. модуль управления работой моделей,
2. модуль ведения реестра.

Описание функциональных модулей Подсистемы выполнения задач представлено в подпунктах 6.1.3.1-6.1.3.2.

6.1.3.1. Функциональный модуль «Управление работой моделей»

Функциональный модуль «Управление работой моделей» предназначен для непосредственного выполнения автоматизированного процесса выявления признаков ЗИ в фото- и видеоданных.

Модуль «Управление работой моделей» является внутренним сервисом АС ОКУЛУС, реализующим следующую функциональность:

1. анализ поступившей на выполнение задачи, применение бизнес-правил для определения:
   - состава выявляемых видов ЗИ;
   - соответствующего набора моделей для использования по каждому виду ЗИ;
   - последовательность выполнения набора моделей.

28. загрузка и, при необходимости, десериализация (преобразование строки в объект) подходящих моделей из репозитория моделей, инициализация и запуск моделей на выполнение,
29. выполнение моделей в соответствии с заданной последовательностью выполнений (вычислительного графа),
30. организация процесса параллельной работы моделей в рамках одной задачи,
31. организация параллельных вычислений в рамках работы отдельной модели,
32. общая организация процесса параллельной обработки поступающих задач (управление пулом выполняющихся задач):
    - формирование пула выполняющихся задач;
    - организация совместного использования моделями вычислительных ресурсов, включая: приоритезацию

процессов, загрузку/выгрузку моделей из оперативной памяти, выделение и высвобождение вычислительных ресурсов под каждую задачу;

33. формирование операционной статистики и сохранение её в хранилище данных АС ОКУЛУС. К данным операционной статистики относятся:
    - количество задач в разбивке по типам и видам задач;
    - количество вызовов моделей в зависимости от типов задач и видов моделей;
    - результат классификации по каждой задаче и модели;
    - возврат результатов работы моделей в модуль «Обработка результатов» (Подсистема обработки данных).

Функциональный модуль «Управление работой моделей» реализуется с учётом следующих особенностей:

1. модуль обеспечивает выполнение моделей, разработанных в основных фреймворках машинного обучения (в качестве таковых могут быть рассмотрены SKLearn, TensorFlow, XGBoost, PyTorch),
2. модуль должен обеспечивать выполнение моделей на различных типах оборудования: как CPU, так и GPU.

6.1.3.2. Функциональный модуль «Ведение реестра»

Функциональный модуль «Ведение реестра» предназначен для хранения обученных моделей, а также всей необходимой для их запуска информации.

Модуль «Ведение реестра» является внутренним сервисом АС ОКУЛУС, реализующим следующую функциональность:

1. регистрация модели в Подсистеме выполнения задач,
34. выполнение базовых функций по хранению, удалению модели из реестра,
35. хранение, редактирование и удаление сопутствующей информации о модели (формализованное описания модели),
36. хранение данных, связанных с моделями:

- параметры модели;
- гиперпараметры модели;
- метрики качества моделей;
- параметры запуска моделей;
- статусы модели (с точки зрения использования в рабочем процессе).

Функциональный модуль «Ведение реестра» реализуется с учётом следующих особенностей:

1. должен быть обеспечен единый формат хранения моделей, параметров и гиперпараметров в сериализованном виде для основных фреймворков машинного обучения,
37. модуль должен предоставлять графический интерфейс для просмотра состава моделей и редактирования, связанной с ними информации,
38. ведение моделей должно быть версионным.

6.1.4.Подсистема ML-операций

Подсистема ML-операций предназначена для автоматизации деятельности пользователей АС ОКУЛУС по разработке, развертыванию и применению моделей машинного обучения. Подсистема ML-операций реализуется на принципах непрерывной интеграции (Continuous Integration, CI) и непрерывной поставке (Continuous Delivery, CD).

Подсистема ML-операций включает в себя следующие функциональные модули:

1. модуль управления ЖЦ наборов данных (datasets),
2. модуль управления ЖЦ моделей.

Описание функциональных модулей Подсистемы ML-операций представлено в подпунктах 6.1.4.1-6.1.4.2.

6.1.4.1. Функциональный модуль «Управление ЖЦ наборов данных»

Функциональный модуль «Управление ЖЦ наборов данных» предназначен для управления наборами данных, использующихся для обучения моделей, на всех стадиях их ЖЦ.

Модуль «Управление ЖЦ наборов данных» является сервисом АС ОКУЛУС, реализующим следующую функциональность:

1. создание новых датасетов (наборов данных) в ручном режиме:
   - создание и аннотация (добавление метаинформации) набора данных (каталога в неструктурированном хранилище данных);
   - добавление и аннотация файлов данных в набор;
   - сохранение созданного датасета в хранилище данных АС ОКУЛУС;
   - создание датасетов в системе путём импорта существующих датасетов из общедоступных источников (файловые ресурсы корпоративной сети) и размещение их во внутреннем хранилище АС ОКУЛУС.

39. модификация существующих (зарегистрированных в АС ОКУЛУС) датасетов:
   - удаление выбранных ФВ-файлов;
   - добавление новых фото- и видеофайлов.

40. поддержание модификации датасета как в ручном, так и в пакетном режиме (слияние выбранного датасета с текущим),

41. управление версионностью датасетов, включая:
   - создание новой версии датасета в ручном режиме;
   - создание новой версии в автоматическом режиме (например, после модификации датасета);
   - архивирование и удаление выбранных версий.
   - выполнение сервисных функций:
   - изменение размеров изображений в соответствии с задаваемыми параметрами;

– автоматический поиск и устранение дубликатов изображений при модификации датасета (добавлении).

Функциональный модуль Управление ЖЦ наборов данных» реализуется с учётом следующих особенностей:

1. должна быть обеспечена поддержка (интеграция) с ПО разметки данных – формирование и ассоциация с каждым размеченным файлом данных разметки (метка класса, метки, координаты границ). Модуль должен обеспечивать целостность данных в формируемой таким образом связки данных.
42. Работа пользователей с модулем ведется через UI.

6.1.4.2.Функциональный модуль «Управление ЖЦ моделей»

Функциональный модуль «Управление ЖЦ моделей» предназначен для реализации конвейера машинного обучения (основных этапов ЖЦ модели машинного обучения).

Модуль «Управление ЖЦ моделей» является сервисом АС ОКУЛУС, реализующим следующую функциональность:

1. подготовка набора данных, анализ набора данных,
2. планирование экспериментов,
3. обучение (включая валидацию моделей),
4. оценка моделей и выбор наиболее оптимальной,
5. развертывание обученных моделей в производственной версии проекта (production)
6. мониторинг показателей работы модели в «производстве»,
7. функциональный поток выявления ЗИ в ФВ-данных.

Для поддержания этапов подготовки и анализа набора данных модуль предоставляет web-ориентированную среду разработки моделей, которая обеспечивает:

– совместную разработку исходного кода моделей в выбранном фреймворке (из числа поддерживаемых фреймворков);
– управление версионностью исходного кода;

– импорт исходного кода state-of-the-art моделей и размещение его в системе.

Для поддержания этапа планирования экспериментов модуль предоставляет следующие функциональные возможности:

1. формирование на основе пользовательского кода последовательности вычислений модели как отдельной сущности АС ОКУЛУС («pipeline»). Формирование включает в себя:
   – специфицирование отдельных этапов последовательности (входные/выходные параметры, данные и интерфейсы)
   – специфицирование последовательности выполнения этапов, условий перехода к тому или иному этапу.

43. формирование и хранение конфигурационных данных для запуска моделей во внутреннем хранилище АС ОКУЛУС.

Для поддержания этапа обучения модуль предоставляет следующую функциональность:

1. обеспечение ассоциативной связи с версией набора данных, на котором проводится обучение,
2. автоматизация процесса нахождения оптимальных гиперпараметров модели. Визуализация взаимосвязи метрик качества и вариантов значений гиперпараметров.
3. автоматизация процесса поиска оптимальной структуры модели (в случае нейронных сетей): количество слоев нейронной сети, количество нейронов в каждом слое,
4. автоматизация процесса обучения моделей:
   – многократное выполнение модели с различными комбинациями гиперпараметров;
   – накопление результатов по каждому запуску в хранилище информации системы. Осуществляется запись и хранение операционной информации (времени выполнения, использования ресурсов), результатов валидации модели,

метрик качества обученной модели по каждой комбинации гиперпараметров.

Для оценки моделей и выбора оптимальной модуль предоставляет следующую функциональность: визуализация результатов расчётов в виде таблиц и графиков, демонстрирующих взаимосвязь значений метрик и комбинаций гиперпараметров.

Для развертывания выбранной версии модели модуль предоставляет следующую функциональность:

1. физическое размещение и регистрация выбранной версии модели (сериализованного файла модели и конфигурационных файлов в репозитории моделей системы и хранилище данных,
2. активация/деактивация (присвоение соответствующего статуса) модели.

6.1.5.Подсистема внутренней отчетности

Подсистема внутренней отчётности предназначена для визуализации как в детализированном (на уровне отдельной логической транзакции), так и в агрегированным (за промежуток времени) виде:

1. операционных характеристик функционирования (время, потребляемые ресурсы) системы в целом и отдельных подсистем,
44. операционных характеристик выполнения процесса классификации фото- и видеоизображений в разрезе задач на обработку набора данных,
45. результатов классификации рабочих моделей,
46. запросов на корректировку результатов в сопоставлении с рабочей моделью, по результатам работы которой поступил запрос.

Подсистема обеспечивает визуализацию как структуры данных, так и динамики их изменения.

6.1.6.Подсистема хранения данных

Подсистема хранения данных предназначена для хранения неструктурированной (файловая система, озеро данных (метод хранения данных

системой или репозиторием в натуральном (RAW) формате, который предполагает одновременное хранение данных в различных схемах и форматах)) и структурированной информации (таблицы СУБД, хранилище данных (предметно-ориентированная информационная база данных)), ведущейся в системе.

Подсистема хранения данных обеспечивает поддержание связности хранимых данных между основными категориями данных, в том числе поддержка связок:

- «параметры работы»-«модель в продуктиве» - «гиперпараметры модели»,
- «модель в продуктиве» - «результаты классификации»,
- «модель в разработке» - «набор данных»,
- «гиперпараметры» - «модель в разработке» - «метрики качества»,
- «входящий запрос» - «модель в продуктиве - «исходящий запрос».

Хранилище неструктурированной информации используется для хранения:

1. данных подсистемы ML-операций:
   - исходного кода программных модулей;
   - сериализованных моделей и конфигурационных файлов;
   - размеченных и неразмеченных наборов данных для обучения моделей;
   - файлов, полученных от внешней системы в случае необходимости корректировки результата классификации.

47. информации Подсистемы обработки данных:
   - копии файла (файлов), подлежащих классификации;
   - промежуточные фото- и видеоданные данные, полученные при дополнительной обработке видеофайлов.

48. данных подсистемы выполнения задач: конфигурационных файлов и сериализованных моделей из реестра моделей.

Хранилище структурированной информации используется для хранения количественных данных:

1. данных Подсистемы интеграции:

- данные по входящим исходящим/соединениям для ведения статистики: метки времени входящего и исходящего запроса, вид запроса, идентификаторы;
- состав, форматы и объем обрабатываемых файлов;

49. операционных данных Подсистемы выполнения задач, в том числе:
    - время выполнения задачи по каждому этапу выполнения модели (этапу «pipeline»);
    - общее время выполнения задачи и общий результат;
    - время выполнения задачи и результат классификации по каждому признаку ЗИ;
2. данные Подсистемы ML-операций:
    - данные по метрикам качества обучаемых моделей по каждому запуску модели;
    - данные по гиперпараметрам обучаемых моделей;
    - операционные данные обучаемых моделей (время запуска, потребляемые ресурсы).

6.2.Функциональная схема (вариант 2)

Функциональными подсистемами АС ОКУЛУС являются:

- Подсистема интеграции,
- Подсистема обработки данных (описание совпадает с вариантом 1),
- Подсистема выполнения задач (описание совпадает с вариантом 1),
- Подсистема ML-операций (описание совпадает с вариантом 1),
- Подсистема внутренней отчетности (описание совпадает с вариантом 1),
- Подсистема хранения данных,
- Подсистема ведения нормативно-справочной информации (описание совпадает с вариантом 1),

– Подсистема администрирования (описание совпадает с вариантом 1),

– Подсистема администрирования сервисов обмена сообщениями.

Функциональная схема АС ОКУЛУС (вариант 2) представлена на рисунке 6.2.

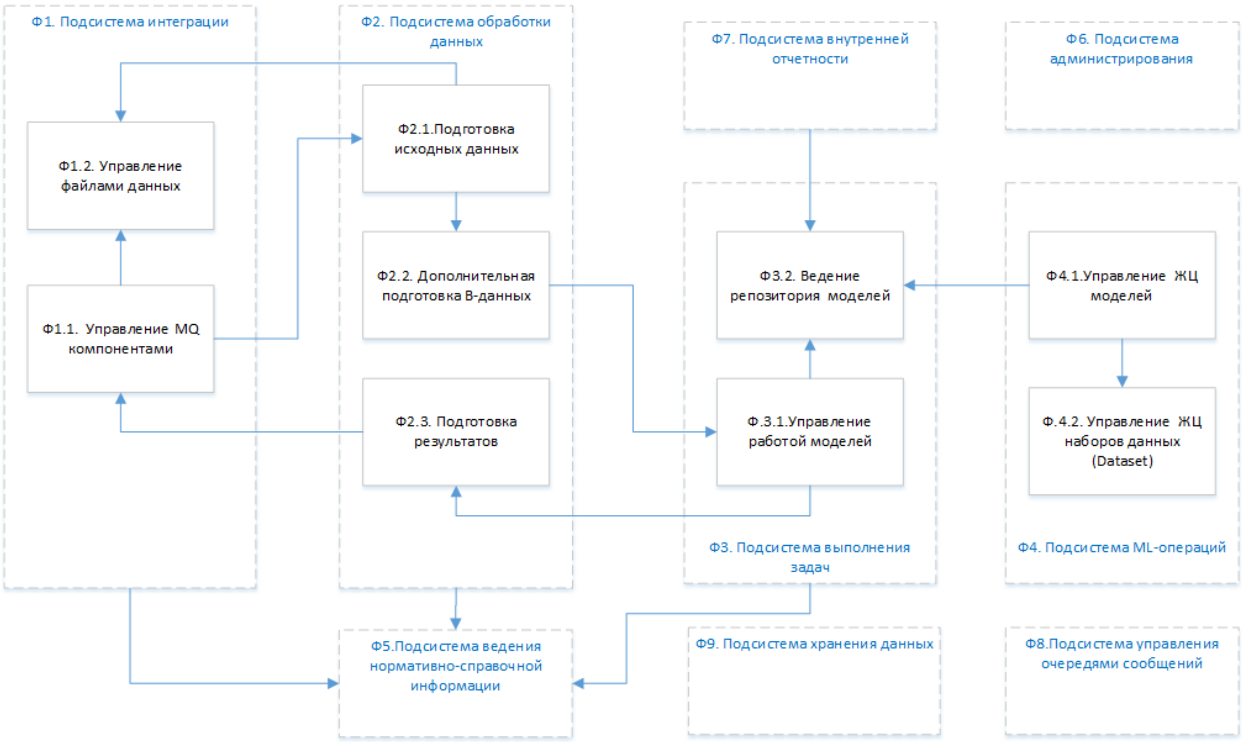

Рисунок 6.2 – Функциональная схема АС ОКУЛУС (вариант 2)

В сравнении с вариантом 1 основные изменения связаны с реализацией другого способа взаимодействия с внешними системами – через очереди сообщений.

В связи с применением очередей сообщений в Подсистему интеграции вводится компонент работы с очередями (MQ-компоненты), который реализует стандартный функционал работы с очередями:

– опрос входной очереди и извлечение из нее сообщений с задачей на выявление ЗИ,

– размещение в исходящей очереди сообщений с уведомлением о выполнении задачи выявления ЗИ.

Все остальные функции Подсистемы интеграции не изменяются.

Также вводится Подсистема управления очередями сообщений, которая реализует стандартный функционал администрирования очередей сообщений.

6.3.Описание вариантов использования

6.3.1.Состав и структура вариантов использования

Состав и структура вариантов использования приведены на рисунке 6.3.

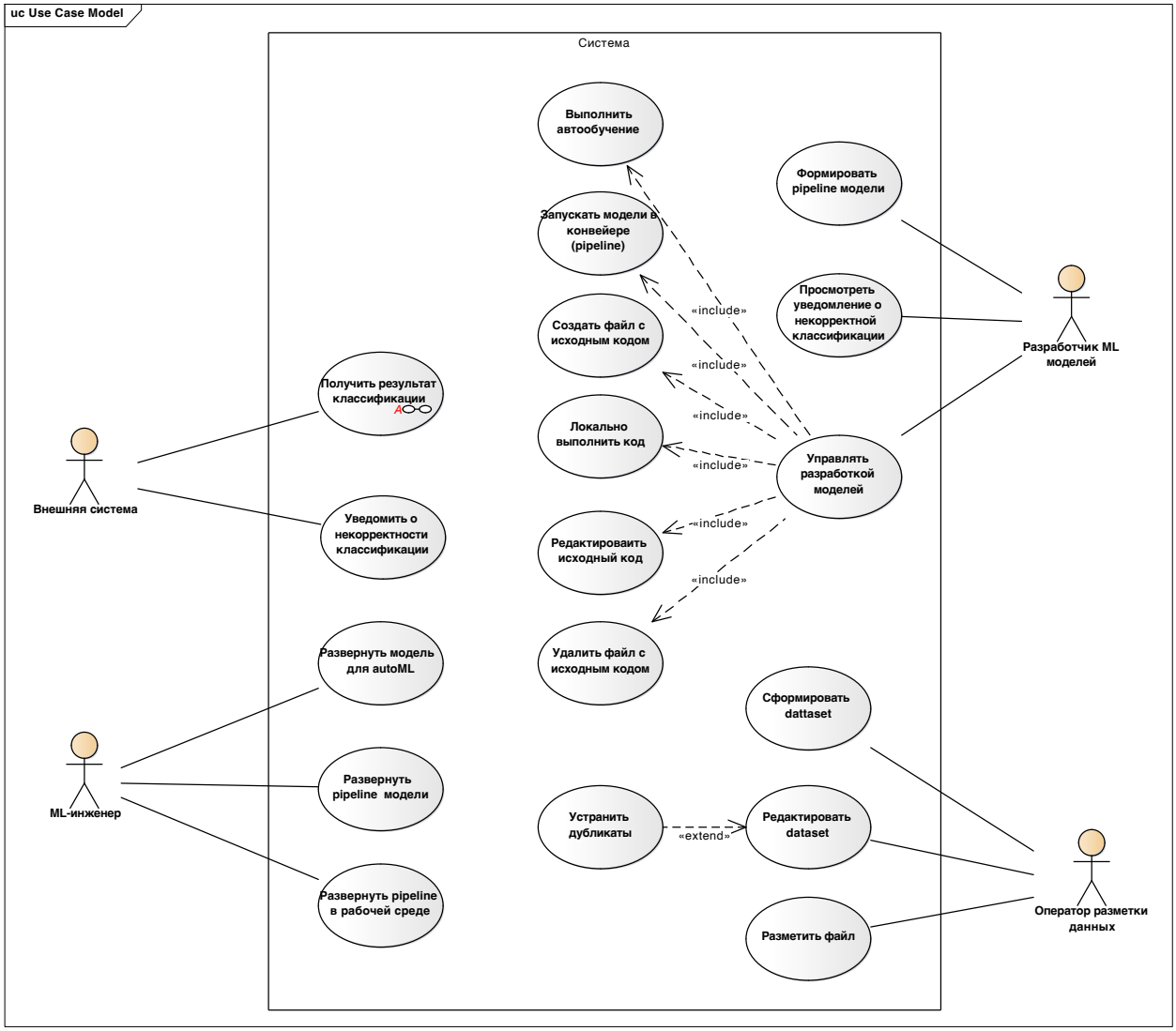

Рисунок 6.3 – Состав и структура вариантов использования

6.3.2.Состав и структура пользователей системы

В АС ОКУЛУС применяется следующий состав бизнес-ролей пользователей:

- внешняя система,
- ML-инженер,
- разработчик ML-моделей,
- оператор разметки данных.

6.3.3.Описание вариантов использования АС ОКУЛУС внешней системой

6.3.3.1.Вариант использования «Получить результат классификации»

В варианте использования «Получить результат классификации» реализован процесс автоматического вывода заключений о наличии признаков ЗИ в предоставляемых фото- и видеофайлах.

Типовой сценарий выполнения варианта использования АС ОКУЛУС внешней системой приведен на рисунке 6.4.

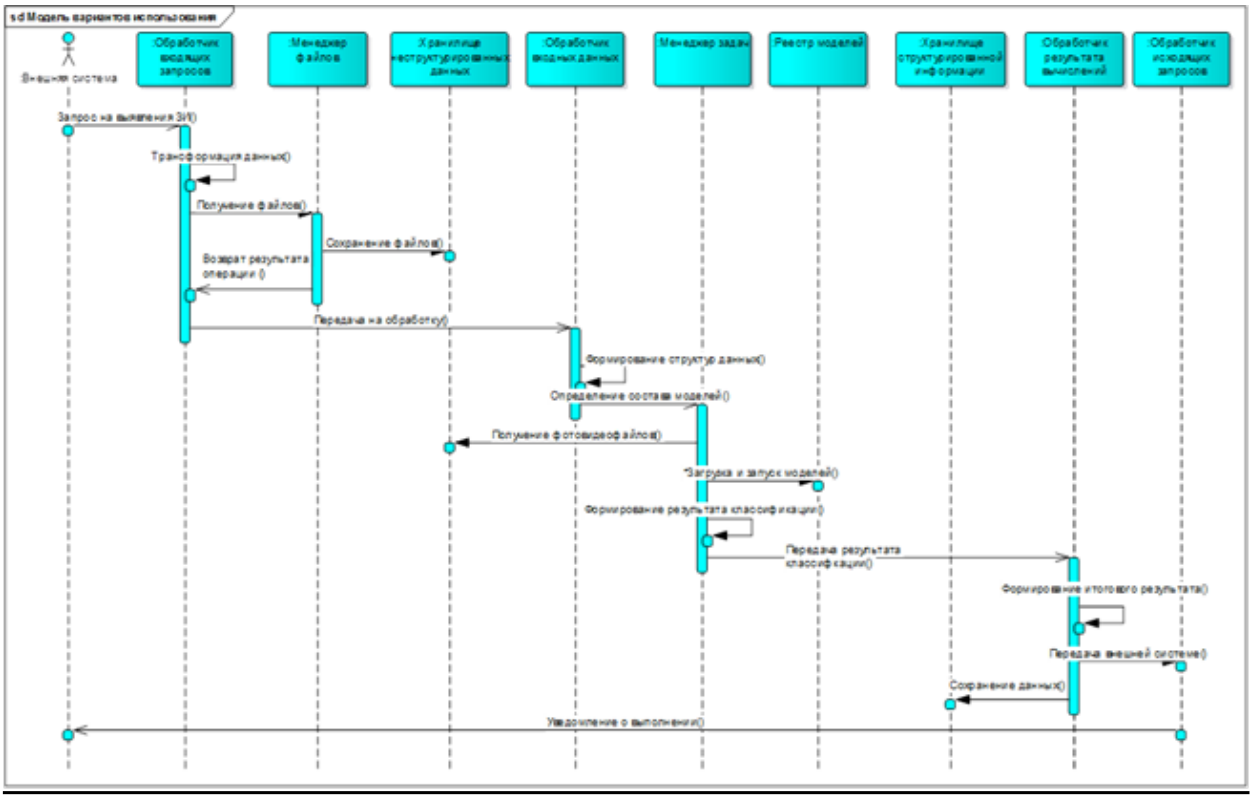

Рисунок 6.4 – Типовой сценарий выполнения варианта использования АС ОКУЛУС внешней системой

Сценарий включает в себя следующую последовательность действий:

1. запрос на выполнение поиска ЗИ. Внешняя система направляет запрос, используя предоставляемый системой API (предлагается реализация REST API). Запрос может содержать:
   - идентификатор внешней системы;
   - адрес или параметры доступа к месту хранения проверяемых файлов во внешнем хранилище (по каждому файлу);
   - метаинформация об обрабатываемых файлах: тип файлов, формат файлов, размер файлов.

– коды признаков ЗИ, по которым необходимо провести проверки (при отсутствии таковых выявление ЗИ будет проводиться по всем признакам, соответствующим типу файла).

2. трансформация данных. Компонент системы «Обработчик входящих запросов» выполняет десериализацию сообщения, разбор тела сообщения, формальную проверку корректности данных сообщения и преобразование разобранных данных во внутренние структуры данных.
3. получение файлов. Обработчик входящих запросов вызывает модуль «Управление файлами данных» для получения файлов из внешнего хранилища. При этом в качестве параметров вызова может передаваться следующая информация:
   – адреса (или параметры доступа) файлов во внешнем хранилище;
   – метаинформация о файлах;
   – коды проверяемых признаков ЗИ (при наличии в исходном сообщении).
4. сохранение файлов. Менеджер файлов копирует файлы из хранилища и размещает их во внутреннем хранилище. В зависимости от типа файла, его метаданных и настроек модуля, модуль перед сохранением может выполнить преобразования:
   – сжатие изображения;
   – преобразование к специальному графическому формату, используемому моделями.
5. возврат результата операции. Менеджер файлов направляет в модуль «Управление входящими запросами» сообщение об успешном завершении операции и возвращает ссылки на расположение во внутреннем хранилище неструктурированной информации скачанных файлов,

6. передача на обработку. Обработчик входящих запросов после возврата результата операции (получение файлов с внешнего хранилища):
    - формирует уникальный в рамках системы идентификатор и метку времени входящего запросов;
    - сохраняет в хранилище структурированной информации параметры входящего запроса, а также сформированный идентификатор и метку времени;
    - формирует результирующую структуру данных, которая может включать в себя:
        - идентификатор входящего запроса;
        - состав файлов и их адреса во внутреннем хранилище АС ОКУЛУС;
        - метаинформацию о файлах.
    - передает сформированную структуру обработчику входящих данных и завершает работу.

50. формирование структур данных. Обработчик входящих данных:
    - определяет состав признаков ЗИ, которые будут выявляться. Критерием сопоставления может являться:
        - тип файла (фото- или видеоизображение);
        - коды признаков ЗИ.
    - на основе состава признаков ЗИ формируется задача на выполнение — структура данных, идентифицирующая в АС ОКУЛУС процесс выявления ЗИ для соответствующего файла с фото- или видеоизображением.
    - формирует структуры данных для инициализации моделей (тензоры, объекты служебных классов).

51. определение состава моделей. Обработчик входящих данных:
    - определяет набор моделей для запуска. Соответствие между кодом признака и моделью задается при размещении модели в реестре моделей. Конкретный способ реализации

соответствия определяется на этапах проектирования АС ОКУЛУС и может включать в себя использование специализированного справочника АС ОКУЛУС или непосредственно использования реестра моделей,

- добавляет найденный набор моделей в задачу на выполнение,
- вызывает функциональность модуля «Управление работой моделей» для непосредственного выполнения классификации. В качестве параметров передаются: задача на выполнение, структуры данных для инициализации модуля.

52. получение фото- и видеоданных. Менеджер задач по адресам внутреннего хранилища неструктурированных данных, указанным в задаче на выполнение, копирует в оперативную память файлы видеоданных,
53. загрузка и запуск моделей. Менеджер задач:
    - выполняет загрузку моделей из реестра моделей;
    - запуск указанных в задаче на выполнение моделей. На вход модели подаются структуры данных для инициализации и загруженное изображение;
    - обеспечивает синхронизацию завершения задачи — задача является выполненной (присваивается соответствующий статус), когда завершены расчеты по всем моделям.
54. формирование результата классификации. После завершения работы моделей менеджер задач рассчитывает и извлекает из объектов, представляющих результаты вычислений:
    - результат классификации по каждому признаку;
    - время начала и время завершения вычислений каждой модели;
    - рассчитанные показатели точности;
    - оценку точности в результате по каждому признаку ЗИ;
    - причины, по которым был сделан вывод о наличии ЗИ (в виде наименовании выявленных объектов).

Менеджер задач данные ассоциирует приведенные выше категории данных (результаты) с задачей на выполнение.

55. формирование итогового результата. Обработчик результата, получив структуры данных, выполняет следующие действия:
    - рассчитывает общий приоритет выполненной задачи. Данный показатель предназначен для использования во внешней системе. Общий приоритет представляет собой скалярную оценку важности состава выявленной ЗИ для приоритезации дальнейшей обработки данных во внешней системе. Общий приоритет может быть рассчитан различными способами:
        - как сумма приоритетов по каждому признаку, где каждый конкретный признак принимает значение P. Значение показателя выставляется по каждому признаку экспертами предметной области из диапазона и сохраняется в специальном справочнике системы. Значение выбирается из диапазона 0 до N (0 — отсутствие ЗИ по данному признаку, N – максимально возможное значение),
        - как максимальное значение приоритета из всех значений (по всем признакам).
    - преобразовывает полученные данные в системные структуры данных, пригодные для формирования запроса с уведомлением о выполнении задачи.
56. передача внешней системе. Менеджер данных завершает работу, передавая в модуль «Управление работой моделей» следующие структуры данных: задача на выполнение с ассоциированными результатами расчета,
57. сохранение данных. АС ОКУЛУС сохраняет сформированную структуру данных в хранилище структурированной информации. Сохраненные данные могут быть использованы для отображения

Подсистемой внутренней отчетности, а также для обработки уведомления от внешней системы о некорректной классификации,

58. уведомление о выполнении задачи. Обработчик исходящих запросов формирует тело уведомления и направляет уведомление внешней системе. Уведомление может содержать:
    - идентификатор первоначального запроса внешней системы,
    - идентификатор обработанной задачи по запросу внешней системы (идентифицирует весь объем информации, связанный с обработкой запроса и сохраненный в АС ОКУЛУС),
    - общий результата выявления ЗИ (например, 0 – отсутствуют признаки ЗИ, 1 – выявлены признаки ЗИ),
    - перечень признаков, по которым выявлена ЗИ,
    - характеристика выявленной ЗИ (например, ключевые слова, запрещенный графический элемент).

6.3.3.2.Вариант использования «Уведомление о некорректной классификации»

Вариант использования «Уведомление о некорректной классификации» выполняется в следующем контексте. Модель может выдать ошибочный результат по одному или нескольким признакам: ошибочно выявить ЗИ либо пропустить ЗИ в фото- и видеоданных. Предполагается, что подобная ошибка может быть обнаружена во внешней системе. В этом случае внешняя система направляет в АС ОКУЛУС уведомление об ошибке.

Типовой сценарий выполнения варианта использования уведомления о некорректной классификации приведен на рисунке 6.5.

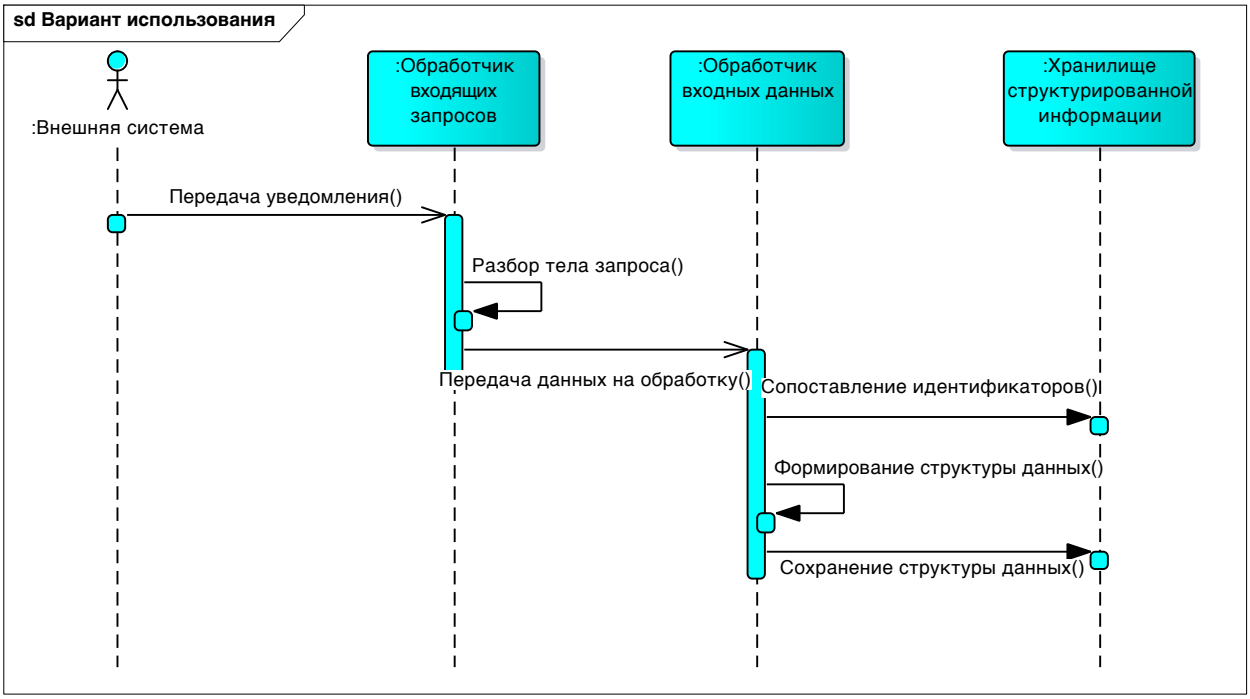

Рисунок 6.5 – Типовой сценарий выполнения варианта использования уведомления о некорректной классификации

Типовой сценарий варианта использования уведомления о некорректной классификации содержит следующие шаги:

1. передача уведомления. Внешняя система направляет запрос, используя предоставляемый АС ОКУЛУС API. Запрос может содержать следующие данные:
   - идентификатор задачи в АС ОКУЛУС (был передан ранее во внешнюю систему как результат);
   - тип сообщения;
   - идентификатор сообщения во внешней системе;
   - признаки ЗИ (код или наименование), неправильно классифицированные;
   - код ошибки (по каждому признаку), например:
     - 0 – ложноотрицательная ошибка, реальная ЗИ не определена,
     - 1 – ложноположительная ошибка.
2. разбор тела запроса. Обработчик входящих запросов разбирает тело запроса в системные структуры данных, выполняет формальную проверку данных.

3. передача данных на обработку. Обработчик входящих запросов передает на обработку разобранные данные,
4. сопоставление идентификаторов. Обработчик входных данных формирует с использованием идентификатора из запроса внешней системы запрос к структурированному хранилищу данных для поиска и извлечения данных по задаче, содержащей неправильно классифицированное изображение или видеоизображение;
5. формирование структуры данных, используя найденную в хранилище данных информацию;
6. сохранение структуры данных. Обработчик входящих данных сохраняет сформированный набор данных в хранилище. Набор данных становится доступен ML-разработчикам в специальном разделе UI для просмотра. ML-разработчики используют наборы данных для оценки и определения показателей деградации качества модели, а также для определения необходимости обновления соответствующего dataset.

6.3.3.3.Основные категории обрабатываемых бизнес-данных

Логическая схема основных категорий данных приведена на рисунке 6.6. Логическая схема данных может быть использована при организации хранения в структурированном хранилище данных.

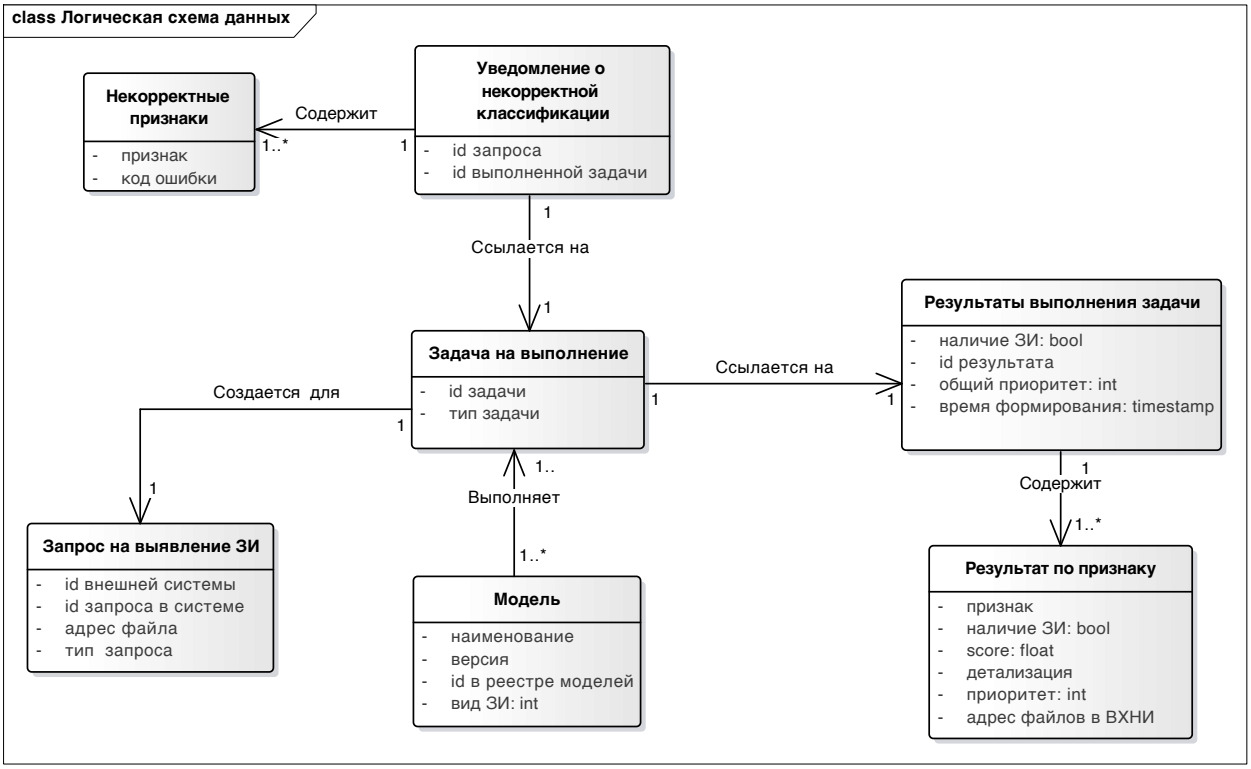

Рисунок 6.6 – Логическая схема основных категорий данных

Основными обрабатываемыми сущностями являются:

1. Уведомление о некорректной классификации – объект системы, который идентифицирует уведомление внешней Системы о том, что в рамках указанной задачи, результат классификации некорректен для одного или нескольких признаков.
2. Некорректные признаки – объект Системы, идентифицирующий набор некорректно классифицированных признаков.
3. Запрос на выявление ЗИ - объект Системы, который идентифицирует запрос от внешней Системы и связанные с ним данными. Появление запроса в системе инициирует его обработку и выявление признаков ЗИ в файлах, на который ссылается запрос.
4. Задача на выполнение - объект Системы, которая идентифицирует и связывает воедино весь состав информации, связанный с обработкой поступающих от внешней системы запросов.
5. Модель – объект системы, идентифицирует состав моделей, используемых для выполнения конкретной задачи выявлении ЗИ

6. Результаты выполнения – идентифицирует набор результатов классификации для задачи по отдельному признаку
7. Результат по признаку – идентифицирует результат классификации по каждому отдельному признаку

«Запрос на выявление ЗИ» может включать, в том числе, следующий набор атрибутов:

- идентификатор внешней системы – id или наименование системы из которой поступил запрос
- идентификатор запроса в Системе – id, однозначно идентифицирующий запрос в системе
- адреса файлов – адреса мест хранения файлов с фото- или видео изображениями на внешнем ресурсе
- тип запроса – может содержать код, уточняющий характеристику классифицируемого содержимого. Например, «фотоизображения», «видеоизображения».

«Задача на выявление ЗИ» может включать, в том числе, следующий набор атрибутов:

- идентификатор задачи – идентифицирует задачу в системе
- идентификатор запроса – идентификатор запроса из внешней системы, для которого выполняется задача
- идентификатор результата - идентификатор набора результатов для соответствующей
- тип задачи – код (или наименование) одного из реализованных в Системе видов выявления ЗИ:
  - классификация изображений;
  - классификация частей изображений (при наложении окна с ЗИ на нейтральный фон);
  - детекция лиц (включая лица, закрытые маской),
  - идентификация лиц по ограниченному перечню персоналий,
  - классификация лиц по возрасту,
  - детекция графических символов,

- детекция URL и каналов мессенджеров в кадре,
- детекция и расшифровка QR-кодов в кадре,
- детекция текстового содержания и распознавание текста для поиска ключевых слов,
- классификация действий в видео.

«Модель» – может включать, в том числе, следующий набор атрибутов:

- наименование или идентификатор модели
- номер версии модели
- адрес хранения модели в реестре моделей
- вид ЗИ, которая выявляется моделью (приведен в отчете о выполнении НИР)

«Результаты выполнения задачи» может включать, в том числе, следующий набор атрибутов:

- идентификатор результата – идентификатор объекта
- наличие ЗИ – True если по задаче была выявлена ЗИ (без уточнения признаков, содержащих ЗИ), иначе False
- общий приоритет задачи - числовая характеристика важности/ значимости, выявленной ЗИ, которая высчитывается на основе приоритетов ЗИ, выявленных в отдельных признаках
- время формирования – время, затраченное на расчеты по задаче

Объект «Результаты по признаку» может включать, в том числе, следующий набор атрибутов:

- признак ЗИ - признак ЗИ, который выявляется в рамках задачи
- наличие ЗИ – 1, если присутствуют признаки ЗИ, 0 – в противном случае
- score – числовая характеристика доверия к результату классификации (атрибуту «наличие ЗИ»
- детализация – наименования конкретных выявленных графических объектов, относящихся к ЗИ (например, «Криминальные татуировки»). Может заполняться, если реализация модели позволяет получить такие значения

– приоритет - числовая характеристика важности/значимости, выявленной ЗИ в соответствующем признаке. Может быть задана для каждого признака экспертным методом и вестись в специализированном справочнике Системы
– адрес файлов в ВХНИ – адрес хранения файлов во внутреннем хранилище неструктурированной информации. Заполняется, если модель предполагает возврат графических объектов - сегментов изображения, на котором присутствует ЗИ

«Уведомление о некорректной классификации» может содержать следующий состав атрибутов:

– идентификатор запроса – идентифицирует конкретное уведомление из внешней системы
– идентификатор выполненной задачи – идентифицирует конкретную задачу, в рамках которой было выполнена некорректная классификация

Объект «Некорректные признаки» - может содержать следующий состав атрибутов:

– идентификатор уведомления – задает связь с соответствующим уведомлением
– признак ЗИ – признак ЗИ, который некорректно был определен Системой
– код ошибки – задает тип ошибки, которая была совершена. Например, 0 – Система не определила наличие ЗИ, 1 – Система определила отсутствие ЗИ при его фактическом наличии

Соответствие между признаками ЗИ, типами задач и моделями задается в специализированных справочниках Системы.

6.3.4.Описание вариантов использования Системы ролью «ML-разработчик»

6.3.4.1.Вариант использования «Управлять разработкой модели»

Разработка модели – это процессы:

– создания исходного кода, реализующего модель

– выполнение исходного кода (с целью обучения)
– выполнение исходного кода (с целью экспериментов на определенном наборе входных CV-данных)

Пользователь разрабатывает модели, формируя исходный код в поддерживаемых компонентом Kubeflow Notebook web-ориентированных средах разработки:

– JupyterLab
– RStudio
– Visual Studio Code

Каждая поддерживаемая Kubeflow Notebook среда обеспечивает стандартный набор функциональных возможностей по разработке (терминал, текстовый редактор, браузер файлов, окно стандартного вывода и т.д.).

Система обеспечивает ML-разработчику все возможности по управлению исходным кодом модели:

– создание файла исходного кода
– редактирование исходного кода
– локальное выполнение исходного кода
– выполнение кода в совместном воспроизводимом режиме (в компоненте Kubeflow Pipeline)
– удаление исходного кода

Основные возможности, предоставляемые Системой по управлению разработкой моделей:

– централизованно хранение и предоставление ML-разработчикам типовых docker-образов сред разработки, в которых будет разрабатываться исходный код
– создание, хранение и предоставление ML-разработчикам специализированных docker-образов сред разработки (отличается от типового наличием дополнительных специализированных библиотек и пакетов)
– UI по просмотру состава файлов, созданию/доступу/удалению файлов

– централизованное хранение исходного кода в кластерах Kubeflow
– Разграничение прав доступа к файлам с исходным кодом
– обеспечение версионности файлов с исходным кодом

6.3.4.2.Вариант использования «Создать файла с исходным кодом»

В web-ориентированной среде разработки файлы создаются на стороне сервера. С каждым файлом ассоциирован компонент «notebook-сервер», который отвечает за работу с файлом в web-интерфейсе компонента Kubeflow.

Сценарий предполагает выполнение следующих шагов:

1. Пользователь переходит в соответствующий раздел UI и выбирает элемент управления (gui control) «Создать новый notebook-сервер»
2. Система предоставляет страницу заполнения параметров создаваемого notebook-сервер
3. Пользователь заполняет параметры, включая:
4. наименование notebook-файла
5. выбирает тип (стандартный или специализированный) docker-образ среды разработки
6. выбирает конкретный docker-образ из состава отображаемых
7. Пользователь задает количество процессоров и оперативной памяти, потенциально необходимых для исполнения разрабатываемого кода
8. Пользователь задает требуемый объем хранения в персональном разделе хранилища неструктурированных данных
9. Пользователь сохраняет параметры создаваемого notebook-сервера
10. Система сохраняет параметры и создает файл исходных данных
11. Система отображает созданный notebook-сервер в списке доступных пользователю notebook-серверов исходного кода

6.3.4.3.Вариант использования «Редактировать исходный код»

Сценарий предполагает выполнение следующих шагов:

1. Система отображает в соответствующем разделе интерфейса перечень доступных разработчику notebook-серверов

2. Пользователь выбирает по ссылке необходимый notebook-сервер и переходит в web-интерфейс его редактирования
3. Пользователь вносит изменения в исходный код
4. Пользователь выдает команду сохранить изменения.
5. Система сохраняет изменения в файле и метаинформации о модификации файла: дата и время изменения

6.3.4.4.Вариант использования «Выполнить код»

Сценарий предполагает выполнение следующих шагов:

1. Разработчик открывает свой соответствующий jupyter notebook
2. Разработчик, используя возможности среды разработки, отправляет код или выбранный блок кода на выполнение.
3. Система выполняет код. При этом результаты выполнения недоступны никому, кроме разработчика.

6.3.4.5.Вариант использования «Удалить файл с исходным кодом»

Сценарий предполагает выполнение следующих шагов:

1. Пользователь переходит в соответствующий раздел UI и выбирает элемент управления (gui control) «Удалить notebook-сервер»
2. Пользователь выдает команду на удаление
3. Пользователь подтверждает удаление
4. Система выполняет «мягкое» (с возможностью восстановления) удаление файла (jupyter notebook) из Системы
5. Файл перестает отображаться в списке файлов, доступных пользователю

6.3.4.6.Вариант использования «Формировать pipeline модели»

Основной целью варианта использования является реорганизация кода, реализующего модель, в последовательность вычислений (workflow), с явным выделением входов и выходов каждого этапа. В качестве отдельного этапа может быть рассмотрена функция, реализующая каждую отдельную ML-задачу.

Реорганизованный код может быть распределен по одному или нескольким notebook ML- разработчика. Реорганизованный код используется ML-инженером в варианте использования «Сформировать pipeline модели»

6.3.4.7.Вариант использования «Запускать модели в конвейере (pipeline)»

Развернутая на стороне Kubeflow обеспечивает для ML-разработчиков ряд преимуществ:

- организация совместной разработки и обучения моделей – более опытный ML-разработчик может разработать основной workflow работы модели, другие – обучать модель, меняя конфигурацию гиперпараметров и входных данных.
- результаты вычислений и метрики качества ассоциируются с конкретным запуском и конкретной конфигурацией модели. Такая взаимосвязь обеспечивает нахождение оптимальных гиперпараметров, понимание зависимостей между входными данными и качеством моделей
- отсутствуют временные затраты на повторное развертывание модели, разработку визуализаций

Типовой сценарий в рамках варианта использования может включать в себя выполнение следующих шагов:

59. ML-разработчик заходит в специальный раздел UI
60. Система отображает доступные пользователю pipeline
61. Пользователь выбирает конкретный pipeline и создает новый (или использует существующий) т.н. «эксперимент» - сущность, в которой создаются различные конфигурации входных параметров, данных и гиперпараметров для запуска модели, реализованной в текущем pipeline.
62. Пользователь создает новый или использует существующий объект «Выполнение» (run)- сущность Системы, которая соответствует отдельному запуску pipeline с конкретной конфигурацией гиперпараметров и входных данных

63. Пользователь, используя элементы UI выдает команду «Выполнить»
64. Система выполняет заданные pipeline вычисления с текущей конфигурацией исходных данных и гиперпараметров.
65. Система агрегирует в рамках текущего запуска результаты вычислений и метрики качества
66. Система сохраняет их в хранилище структурированной информации и отображает их в UI в контексте конкретного запуска

6.3.4.8.Вариант использования «Выполнить автообучение»

Под автообучением в данном случае понимается автоматизация процесса поиска оптимального набора гиперпараметров и структуры нейронной сети. Использование автообучение позволяет существенно сократить время на поиск оптимальных гиперпараметров и структуры нейронной сети.

ML-разработчик может воспользоваться возможностями по автообучению, предоставляемыми компонентом Katib. Типовой сценарий варианта использования может содержать следующие шаги:

1. Пользователь переходит в соответствующий раздел UI Системы
2. Система отображает перечень доступных для автообучения моделей (порядок размещения моделей в компоненте описан в варианте использования «Развернуть модель для autoML»)
3. Пользователь выбирает необходимую модель, открывает форму редактирования параметров (описаны в варианте использования «Развернуть модель для autoML») и вносит значения необходимых параметров
4. Пользователь отправляет модель на обучение
5. Система обеспечивают выполнение модели
6. Система сохраняет в хранилище структурированной информации результаты вычислений, гиперпараметры и значения метрик качества в привязке к каждому (в рамках проведенного эксперимента) запуску модели.

7. Система отображает наиболее оптимальный кортеж (при заданных условиях эксперимента) значение гиперпараметров
8. Система визуализирует все кортежи значений метрик и гиперпараметров (пример приведен на рисунке 6.7), выделяя наиболее оптимальный

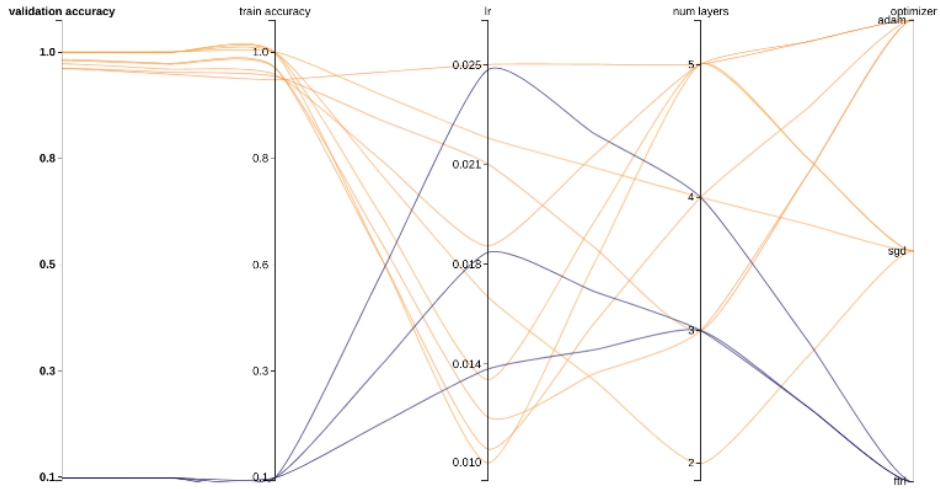

Рисунок 6.7 – Пример анализа кортежей значений метрик и гиперпараметров

6.3.4.9.Вариант использования «Просмотреть уведомление о некорректной классификации»

Вариант использования выполняется в следующем контексте. Система обработала уведомление о некорректности ранее проведенной классификации (см. описание варианта использования «Уведомление о некорректной классификации»). Накапливаемая информация по уведомлениям может быть использована для оценки степени деградации работающих моделей.

Сценарий может включать следующие шаги:

1. Пользователь переходит в раздел UI, связанный с просмотром уведомлений
2. Система отображает список уведомлений в структурированном виде (с возможностью фильтрации по дате получения, статусу)
3. Пользователь выбирает необходимое уведомление и переходит на страницу с детализированной информацией
4. Система отображает пользователю информацию по уведомлению. В состав ключевых атрибутов входят:

5. Наименование модели
6. Признаки
7. Результат классификации признака Системой (по каждому признаку) с выделением ошибочного результата
8. Использованный для обучения dataset
9. Текущий статус уведомления в Системе
10. Пользователь может (при необходимости) сменить статус конкретного уведомления. Возможными значениями статусов уведомления могут являться:
11. «Получено» - после получения и сохранения уведомления в системе
12. «Просмотрено» - изменяется пользователем

6.3.5.Описание вариантов использования Системы ролью «ML-инженер»

6.3.5.1.Вариант использования «Развернуть pipeline модели»

Основной целью варианта использования является:

- специфицирование этапов вычислений, каждый из которых является компонентом рабочего процесса (workflow). В общем случае, в качестве компонента может рассматриваться каждая функция, реализующая отдельную ML-задачу.
- развертывание кода в компоненте Kubeflow Pipelines. Развертывание и публикация кода модели позволит осуществить переход от модели, ведущейся изолированно в notebook конкретного разработчика, к модели как общедоступному ресурсу Системы.

ML-инженер, используя исходный код, может выполнить следующие действия:

- Специфицировать каждый этап workflow: наименование, входные параметры, возвращаемые значения, способ выполнения, состав результатов, который может быть визуализирован.
- Специфицировать связи между различными компонентами – организовать граф вычислений, поддерживаемый компонентом

– Выполнить сборку docker-контейнеров (на каждый компонент) и разместить их в реестре контейнеров Kubeflow

Спецификация на компонент представляет собой документ формата YAML, который размещается в компоненте Kubeflow Pipelines.

После размещения контейнеров и конфигурационных YAML-файлов, pipeline модели становится доступным для использования ML-разработчиками.

6.3.5.2.Вариант использования «Развернуть модель для autoML»

Для обеспечения возможностей проведения экспериментов по поиску оптимальных гиперпараметров и структуры нейронной сети, ML-инженер выполняет развертывание модели в компоненте Katib. Для этого выполняется следующая последовательность основных действий:

1. Формирование docker-контейнера для исходного кода модели и регистрация его в реестре Kubernetes
2. Формирование конфигурационного файла (yaml-формата), определяющего объем вычислений и вид алгоритма автообучения. Могут быть заданы следующие параметры:
   – пространство поиска – набор используемых гиперпараметров, вид распределения и диапазон изменений каждого параметра или список их возможных значений
   – цели автообучения – набор метрик, которые необходимо оптимизировать (целевые переменные)
   – алгоритм поиска оптимальных значений (из числа поддерживаемых Katib)
67. Регистрация конфигурационного файла в системе

Система отображает развернутую модель в специальном разделе UI, также становится возможной работа с моделью через интерфейс командной строки.

6.3.5.3.Вариант использования «Развернуть pipeline модели в рабочей среде»

Целью варианта использования является включение разработанной, обученной и развернутой в Kubeflow Pipeline модели в основной бизнес-процесс,

поддерживаемой Системой – определение ЗИ в поступающих фото– и видеофайлах (см. описание варианта использования «Получить результат классификации»).

С этой целью ML-инженер разрабатывает конфигурационный файл, содержащий параметры использования модели компонентом Kubeflow KFServing. После размещения файла в компоненте KFServing модель становится доступной для использования (работы в «продуктиве»).

6.3.6.Описание вариантов использования Системы ролью «Разметчик данных»

6.3.6.1.Вариант использования «Сформировать dataset»

Типовой сценарий формирования набора данных для обучения может включать в себя следующие шаги:

1. Пользователь, используя UI, формирует структуру каталогов в хранилище неструктурированной информации для хранения тестовой, валидационной и обучающей части набора данных
2. Пользователь добавляет выбранные каталоги в систему контроля версий, которая является компонентом Системы
3. Пользователь копирует в выбранные каталоги из внешних источников необходимый набор файлов, составляющих dataset
4. Пользователь фиксирует номер версии в системе контроля версий

6.3.6.2.Вариант использования «Редактировать dataset»

Редактирование набора данных включает в себя:

- добавление или удаление файла в существующем dataset
- изменение текущей версии набора dataset

Добавление и удаление файла в выбранный dataset выполняется пользователем в UI системы. В зависимости от настроек Системы добавление файла может сопровождаться следующими проверками и процедурами предобработки:

- контроль формата добавляемых файлов
- устранение дубликатов среди добавляемых файлов

Изменение текущей версии набора данных осуществляется на уровне каталога, добавленного в систему контроля версий.

6.3.6.3.Вариант использования «Разметить файл»

Разметка осуществляется с помощью специализированного инструмента аннотирования данных (например, LabelMe).

Пользователь, используя инструментарий:

- открывает требуемый каталог и файл
- выполняет разметку данных – выделяет на изображении области и точки
- сохраняет измененный файл вместе с метаинформацией в выбранном каталоге

6.3.6.4.Вариант использования «Устранить дубликаты»

Вариант использования выполняется автоматически при добавлении нового файла (набора файлов) в dataset. Возможна следующая последовательность действий:

1. Пользователь, используя UI выбирает с локального хранилища данных или иного источника требуемый набор файлов и выдает команду добавить набор в выбранный dataset
2. Система для каждого добавляемого файла выполняет алгоритм сравнения с существующими файлами набора данных. Сравнение осуществляется на основе сравнения семантической близости элементов изображения
3. Система, основываясь на метриках похожести, выдает список в UI всех возможных файлов-дубликатов из добавляемого набора данных
4. Пользователь может удалить определенные файлы из списка дубликатов
5. Пользователь выдает команду на подтверждение добавления
6. Система размещает добавляемые файлы в каталог выбранного dataset

7.Определение предпочитаемого стека технологий, используемых в АС

С точки зрения программной архитектуры Система представляет собой набор контейнеризированных приложений, выполняющихся в среде контейнеризации. Программная архитектура Система включает в себя компоненты следующих категорий:

- Базовый уровень — среда контейнеризации
- Прикладной уровень — компоненты, обеспечивающие:
  - Хранение данных
  - Обработка данных
  - Машинное обучение

Среда контейнеризации состоит из сервисов, обеспечивающих выполнение приложений в контейнерах. Состав и рекомендуемое ПО для их реализации приведены в таблице 7.1.

Таблица 7.1 – Состав и рекомендуемое ПО для реализации среды контейнеризации

Тип сервиса	Требования	Рекомендуемые технологии
Container engine	Должен обеспечивать среду выполнения контейнеризированными приложениями с поддержкой Persistent Volume Claims, возможностью организации баллансировки нагрузки и конфигурации DNS. Образы должны браться из внутреннего container registry сервиса. Публичные образы должны браться из Docker Hub.	- OpenStack Magnum - для создания инфраструктурной среды - Docker – как контейнеризатор - Kubernetes — как среда выполнения и управления контейнерами
Container registry	Должен обеспечивать хранение и предоставление Docker образов. Доступен только изнутри платформы.	Docker Registry
Service Mesh	Должен обеспечивать следующие возможности: - конфигурирование сетевых правил для организации взаимодействия между контейнеризированными приложениями без изменений кода. - балансировка сетевой нагрузки - аутентификация и шифрование трафика	Istio on Kubernetes

Программная архитектура прикладного уровня приведена на рисунке 7.1.

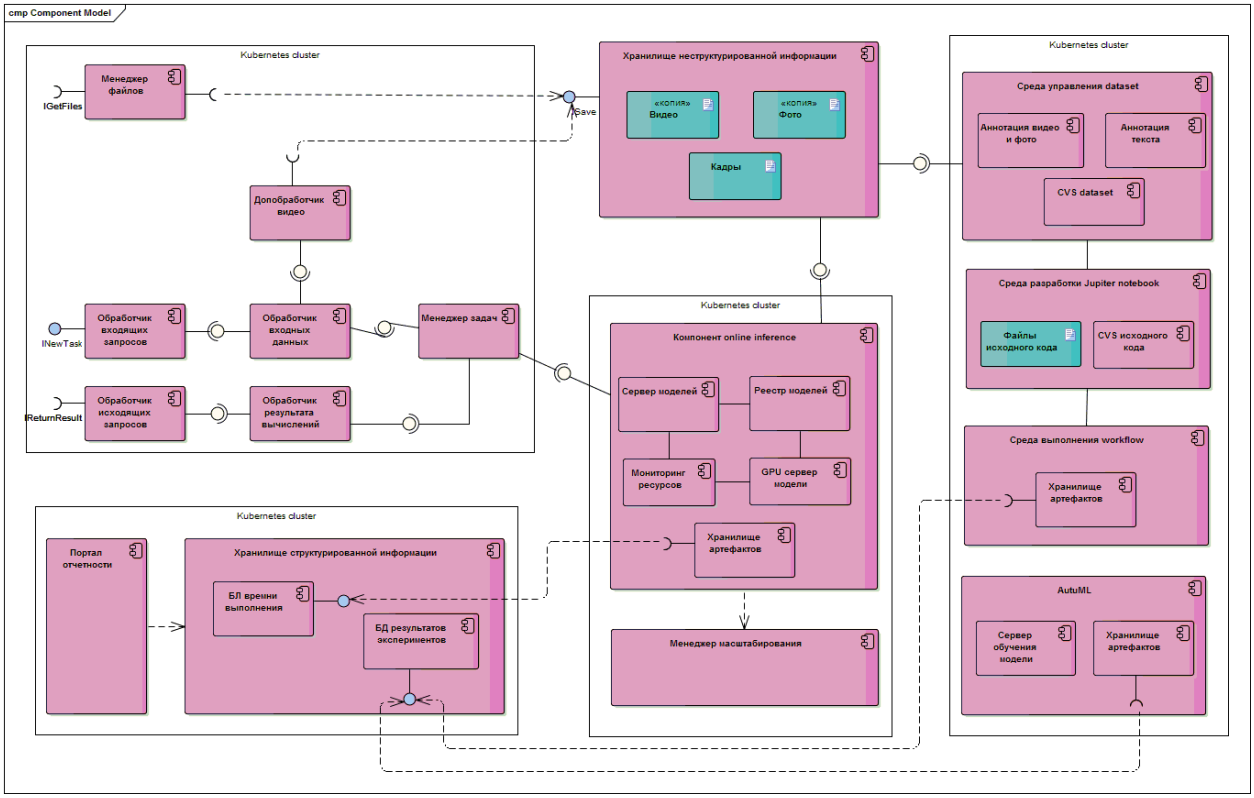

Рисунок 7.1 - Программная архитектура прикладного уровня

Состав компонентов уровня хранения данных и рекомендуемое ПО для их реализации приведены в таблице 7.2.

Таблица 7.2 - Состав компонентов уровня хранения данных и рекомендуемое ПО для их реализации

Тип сервиса	Требования	Рекомендуемые технологии
Хранилище неструктурированной информации	Должно обеспечивать ключевые возможности по хранению и доступа к данным: − адаптация к изменению нагрузки на хранилище − управление доступом и правилами хранения − обеспечение отказоустойчивости и доступности сервисов	Облачное объектное хранилище данных, реализующее S3-протокол доступа к данным и развернутое на инфраструктуре заказчика MinIO
БД времени выполнения	Должен представлять Online Transaction Processing Relational Database Management System (OLTP RDBMS) сервис, обеспечивающий консистентное хранение реляционных данных и их выборку, используя SQL.	PostgreSQL
БД результатов экспериментов	Должен представлять Online Transaction Processing Relational Database Management System (OLTP RDBMS) сервис, обеспечивающий консистентное хранение реляционных данных и их выборку, используя SQL.	PostgreSQL
Портал отчетности	Должен предоставлять визуализации структурированных данных в виде интерактивных графиков и диаграмм	Grafana

Состав компонентов уровня обработки данных и рекомендуемое ПО для их реализации приведены в таблице 7.3.

Таблица 7.3 - Состав компонентов уровня обработки данных и рекомендуемое ПО для их реализации

Тип сервиса	Требования	Рекомендуемые технологии
Обработчик входящих запросов	Должен реализовать алгоритм обработку входящих запросов на основе REST API	Python 3.7
Обработчик исходящих запросов	Должен реализовать алгоритм обработку исходящих запросов на основе REST API	Python 3.7
Менеджер файлов	Должен реализовать алгоритм сбора данных из внешнего объектного хранилища	Python 3.7, AWS SDK for Python (Boto3)
Обработчик входных данных	Должен реализовать требуемый алгоритм обработки данных	Python 3.7
Обработчик результатов вычислений	Должен реализовать требуемый алгоритм обработки данных	Python 3.7
Дополнительный обработчик видео	Должен реализовать алгоритм обработки изображений — преобразование формата и разбиение на кадры	Python 3.7, FFmpeg
Менеджер задач	Должен реализовать требуемый алгоритм обработки данных и управления запуском моделей	Python 3.7

Интеграция с внешними системами может быть также реализована с помощью очередей обмена сообщениями. В этом случае вместо обработчиков входящих и исходящих запросов реализуются входящая и исходящая очередь сообщений под управлением брокера сообщений. Предложения по технологиям приведены в таблице 7.4.

Таблица 7.4 - Предложения по технологиям для реализации сервисов системы

Тип сервиса	**Требования**	**Рекомендуемые технологии**
Брокер сообщений	Должен предоставлять возможность асинхронного обмена сообщениями между Системой и внешней системой Сервис также должен обеспечивать: − отсутствие единой точки отказа за счет распределения данных и программных приложений по узлам кластера. − возможность добавления новых источников данных и узлов с различными характеристиками в единый вычислительный кластер. − механизм реплицирования и синхронизацией реплик. − возможность уведомлений о получении сообщений для источников, а также различные стратегии отправки сообщений (строго однократная, как максимум однократная, периодическая, поочередная)	Kafka
Очередь входящих сообщений	Должен реализовать извлечение сообщений из входящей очереди	Python 3.7
Очередь исходящих сообщений	Должен реализовать размещение сообщений из входящей очереди	Python 3.7

Состав компонентов уровня машинного обучения и рекомендуемое ПО для их реализации приведены в таблице 7.5.

Таблица 7.5 - Состав компонентов уровня машинного обучения и рекомендуемое ПО для их реализации

Тип сервиса	Требования	Рекомендуемые технологии
Компонент on-line inference	Должен обеспечивать получение результата классификации обученной модели в рамках REST запроса, онлайн.	KFServing
Сервер моделей	Обеспечение работы модели в on-line inference	Out-of-the-box сервера моделей (TorchServer – для PyTorch)
Реестр моделей	Должен обеспечивать сериализацию и хранение моделей	KFServing (встроенные сервисы)
Мониторинг ресурсов	Должен отслеживать метрики и операционные показатели при on-line inference	Prometheus
Хранилище артефактов	Должен обеспечивать мониторинг метрик модели и данных.	Kubeflow Metadata,
Менеджер масштабирования	Должен обеспечивать автоматическое масштабирование управляемого приложения в зависимости от входящего трафика, а также управление прохождением трафика.	Knative Serving
Аннотация видео и фото	Ручная разметка фото и видео	CVAT
Аннотация текста	Ручная разметка текста	WebAnno / brat
CVS dataset	Должен обеспечивать управление версионностью наборов данных в привязке к обучаемым моделям	dvc
Среда разработки Jupiter Notebook	Среда для интерактивной разработки, аналитики и проведения экспериментов машинного обучения.	Kubeflow Notebook
CVS исходного кода	Должен обеспечивать управленит версионностью кода	Git
Среда выполнения workflow	Должен обеспечивать выполнение модели как компонентов графа вычислений и поддержку проведения экспериментов с моделями	Kubeflow Pipelines
Auto ML	Должен обеспечивать автоматических подбор архитектуры модели под имеющийся dataser, включая подбор гиперпараметров.	Kubeflow Katib
Сервер обучения модели	Должен обеспечивать обучение моделей с использованием ускорителей на больших объемах данных	Kubeflow Training Operator

8.Описание зоны эксплуатации АС

АС ОКУЛУС должна эксплуатироваться внутри помещений при следующих климатических условиях:

- температура от плюс 10 до плюс 35°С;
- относительная влажность до 60% при плюс 20°С, допускается кратковременное повышение влажности до 80% при плюс 25°С;
- давление от 700 до 1060 гПа.

Питание аппаратной части АС ОКУЛУС должно осуществляться от сети переменного тока напряжением 220 В и частотой 50 Гц.

В составе аппаратной части АС ОКУЛУС должны быть предусмотрены источники бесперебойного питания, для обеспечения функционирования АС ОКУЛУС в течении времени не менее 2 часов, в случае отключения электроэнергии на объекте размещения АС ОКУЛУС.

АС ОКУЛУС не требует проведения технического обслуживания.

Техническое обслуживание АС ОКУЛУС должно осуществляться представителями организации-разработчика (изготовителя).

9.Подходы (политика) сопровождения функционирования АС

Подходы сопровождения функционирования АС ОКУЛУС включают в себя следующие направления:

- оценка качества функционирования АС ОКУЛУС,
- совершенствование методов и алгоритмов, применяемых в АС ОКУЛУС,
- безопасная разработка ПО,
- ведение нормативной, проектной и эксплуатационной документации,
- поддержание достаточного уровня квалификации персонала,
- модернизация и внесение изменений АС ОКУЛУС,
- обеспечение ИБ в рамках всего ЖЦ АС ОКУЛУС.

Входящие в политику сопровождения функционирования АС ОКУЛУС направления рассмотрены в пунктах 9.1-9.7.

9.1.Оценка качества функционирования АС ОКУЛУС

Оценка качества функционирования АС ОКУЛУС должна проводиться на регулярной основе в соответствии с требованиями и порядком, установленными в нормативно-методической документации.

В целях проведения оценки формируется постоянно действующая Комиссия по оценке качества функционирования АС ОКУЛУС (далее – Комиссия). Созыв Комиссии осуществляется планово в соответствии с установленной документально периодичностью и внепланово в соответствии с требованиями, устанавливаемыми в организационно-распорядительных документах.

В рамках оценки качества функционирования АС ОКУЛУС должны рассматриваться выдаваемые результаты ИИ по обнаружению ЗИ в фото- и видеоизображениях, а также подаваться на тестирование случайные и подготавливаемые наборы данных.

9.2.Совершенствование методов и алгоритмов, применяемых в АС ОКУЛУС

Комиссия организует совершенствование обучающих подходов, наращивание объёмов и качества тестовых выборок данных для каждого вида моделей. Комиссия устанавливает количество образцов, необходимое для обучения классификатора, для каждого класса в отдельности.

В соответствии с заранее установленной периодичностью Комиссией проводится исследование научной базы в области ИИ, нейронных сетях и алгоритмов обнаружения, методов распознавания и анализа изображений и видеоизображений.

Периодически членами Комиссии осуществляется сравнительный анализ применяемых в моделях алгоритмов по заранее утверждённым критериям сравнения. Критерии сравнения должны быть определены документально для каждого класса данных в отдельности.

Комиссия анализирует применяемые в моделях методы:

- решения задач по распознаванию признаков ЗИ в изображениях и видеоматериалах,
- распознавания и классификации визуальных образов на изображениях и видеоматериалах на наличие признаков ЗИ,
- распознавания действий в видеоматериалах для выявления признаков ЗИ

Комиссия организует макетирование методов и алгоритмов выявления на изображениях и в видеоматериалах признаков ЗИ с целью доработки функциональности АС.

По результатам анализа методов и алгоритмов, применяемых в АС ОКУЛУС, Комиссия вносит предложения (рекомендации) по доработке АС ОКУЛУС.

9.3.Безопасная разработка ПО

В рамках создания и функционирования АС ОКУЛУС должна обеспечиваться безопасная разработка ПО в соответствии с выполнением

требований соответствующих нормативных актов РФ. Должны быть сформированы и документированы требования к разработке кода ПО, допустимым используемым технологиям и ПО. В случае заказной разработке необходимо предусмотреть обеспечение безопасности предоставляемого физического и/или удаленного доступа, а также организационной безопасности. Рекомендуется организовать обеспечение целостности ПО АС ОКУЛУС.

9.4.Ведение нормативной, проектной и эксплуатационной документации

На всех стадиях ЖЦ АС ОКУЛУС должны осуществляться:

- разработка и ведение проектной, эксплуатационной документации на АС ОКУЛУС,
- разработка локальных нормативных актов, регламентирующих порядок использования АС ОКУЛУС,
- документирование ответственности персонала в локальных и нормативных актах и должностных инструкциях работников, участвующих в создании, эксплуатировании, взаимодействии, администрирования и обслуживания АС ОКУЛУС.

9.5.Поддержание достаточного уровня квалификации персонала

В целях обеспечения безопасного функционирования и корректного использования АС ОКУЛУС должны обеспечиться следующие меры по поддержанию достаточного уровня квалификации персонала:

- проведение обучающих мероприятий по АС ОКУЛУС для персонала, участвующего в создании, эксплуатировании, взаимодействии, администрирования и обслуживания АС ОКУЛУС,
- повышение осведомленности, инструктаж, ознакомление с правилами и мерами обеспечения ИБ при использовании АС ОКУЛУС,

– систематическая планомерная деятельность по улучшению осведомленности персонала о порядке работы с АС и обеспечения ИБ.

9.6.Модернизация и внесение изменений в АС ОКУЛУС

Модернизация и внесение изменений в АС ОКУЛУС осуществляется на основе приказа с последующим внесением изменений в проектную и эксплуатационную документацию на АС и систему ее защиты.

Модернизация и внесение изменений в АС ОКУЛУС могут быть осуществлены в случае:

– изменений требований действующего законодательства РФ в области регулирования ЗИ,
– по результатам контроля качества функционирования АС,
– согласованного руководством предложения по совершенствованию АС, вынесенного эксплуатирующими АС лицами.
– выявления несоответствий функциональных характеристик АС установленным в проектной и эксплуатационной документации,

9.7.Обеспечение ИБ в рамках всего ЖЦ АС ОКУЛУС

Обеспечение ИБ АС ОКУЛУС должно осуществляться на стадиях всего ЖЦ цикла ИБ, с учётом всех сторон, вовлеченных в процессы ЖЦ (разработчиков, заказчиков, поставщиков продуктов и услуг, эксплуатирующих и обслуживающих организаций).

Разработка технических заданий, проектирование, создание, тестирование, приемка средств и систем защиты ИС проводится при участии АИБа и системного администратора. Порядок разработки и внедрения ИС должен быть регламентирован и контролироваться.

При разработке и эксплуатировании АС необходимо придерживаться требований и методических указаний, определенных следующими нормативно-правовыми актами и документами (включая, но не ограничиваясь):

- Федеральный закон от 27.07.2006 № 149-ФЗ (ред. от 19.07.2018) «Об информации, информационных технологиях и о защите информации».
- Приказа ФСТЭК №17-2013 «Об утверждении Требований о защите информации не содержащей государственной тайны, содержащейся в государственных информационных системах»,
- Методический документ ФСТЭК России от 11.02.2014 «Меры защиты информации в государственных информационных системах».
- ГОСТ 34.601-90. Автоматизированные системы. Стадии создания;
- ГОСТ 34.602-2020 Информационные технологии (ИТ). Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы;
- ГОСТ Р 59792-2021 Информационные технологии (ИТ). Комплекс стандартов на автоматизированные системы. Виды испытаний автоматизированных систем;
- ГОСТ Р 57100-2016 СПИ. Описание архитектуры;
- ГОСТ 34.201-89. Виды комплектность и обозначение документов при создании Автоматизированных систем;
- ГОСТ Р 57193-2016 СПИ. Процессы жизненного цикла систем;
- ГОСТ Р ИСО-МЭК 12207-2010 СПИ. Процессы жизненного цикла программных средств;
- ГОСТ Р ИСО-МЭК 14764-2002 ИТ. Сопровождение программных средств;
- ГОСТ Р 59277-2020. Системы искусственного интеллекта. Классификация систем искусственного интеллекта;
- ГОСТ Р 54593-2011. Свободное программное обеспечение;
- ГОСТ 19.401-78. Единая система программной документации. Текст программы. Требования к содержанию и оформлению;

Обеспечение ИБ в рамках всего ЖЦ АС ОКУЛУС осуществляется реализацией следующих мер:

- обеспечение физической безопасности технических средств (ТС) и программных компонентов АС ОКУЛУС,
- идентификация и аутентификация субъектов доступа и объектов доступа,
- управление доступом к ТС и программным компонентам АС ОКУЛУС,
- ограничение программной среды АС ОКУЛУС,
- обеспечение безопасной эксплуатации ТС АС ОКУЛУС, включая регламентное обслуживание ТС и программных компонентов АС ОКУЛУС,
- оснащение средствами антивирусной защиты ТС АС ОКУЛУС, включая ТС пользователей и администраторов, настройка и эксплуатирование в соответствии с эксплуатационной документацией,
- обеспечение бесперебойной эксплуатации ТС и программных компонентов АС ОКУЛУС. Регулярное резервирование баз данных, реестра моделей, датасетов и иных составляющих АС ОКУЛУС, в т.ч. программных компонентов,
- обеспечение сетевой безопасности ТС и программных компонентов АС ОКУЛУС на всех уровнях взаимодействия: при работе с АС пользователей, при администрировании или техническом обслуживании АС, при взаимодействии АС с внешними ИС,
- обеспечение безопасной разработки ПО АС ОКУЛУС,
- регистрация событий безопасности АС ОКУЛУС,
- обеспечение реагирования на инциденты ИБ,
- управление обновлениями ПО,
- контроль (анализ) защищенности информации в АС ОКУЛУС. Периодический контроль уровня защищенности информации в

АС ОКУЛУС, периодический контроль эффективности работы системы защиты информации.

10.Оценка класса АС в соответствии с требованиями Приказа ФСТЭК России от 13 февраля 2013 года №17

В целях предварительной оценки класса АС в соответствии с требованиями Приказа ФСТЭК России от 13 февраля 2013 года № 17 (далее – Приказ № 17) следует рассмотреть следующее мероприятие для обеспечения защиты информации, содержащиеся в автоматизированной системе «Окулус» - «Формирование требований к защите информации, содержащейся в информационной системе»

Ввиду реализации автоматизированной системы как комплекса программных и аппаратных решений, а также разработки ПО при реализации данных процедур необходимо ориентироваться на перечень государственных стандартов:

1. ГОСТ Р 51583 «Защита информации. Порядок создания автоматизированных систем в защищенном исполнении. Общие положения»
2. ГОСТ Р 51624 «Защита информации. Автоматизированные системы в защищенном исполнении. Общие требования»
3. ГОСТ Р 56939 «Защита информации. Разработка безопасного программного обеспечения. Общие требования»

На основании пункта 2 настоящего документа предполагается разработка непосредственно исходного кода программного обеспечения, а как следствие возможного наличия уязвимостей кода. Для защиты от угроз, связанных с эксплуатацией автоматизированной системы и особенностями её архитектуры предполагается выполнение требований Приказа № 17. В то же время для обеспечения необходимого уровня защиты информации требуется реализация мер, направленных на предотвращение появления и устранение уязвимостей программ в процессах жизненного цикла программного обеспечения, связанных с уязвимостями непосредственно исходного кода разрабатываемой автоматизированной системы.

В соответствии с вышеуказанными мероприятиями необходимо выполнить следующие процедуры:

1. Принятие решения о необходимости защиты информации;
2. Классификация информационной системы по требованиям защиты информации;
3. Определение угроз безопасности информации, реализация которых может привести к нарушению безопасности информации в информационной системе, и разработку на их основе модели угроз безопасности информации;
4. Определение требований к системе защиты информации информационной системы
5. Реализация цикла общих требований к содержанию и порядку выполнения работ, связанных с созданием безопасного программного обеспечения и формированием среды обеспечения оперативного устранения выявленных пользователями ошибок программного обеспечения и уязвимостей программы

Для корректного принятия решения о необходимости защиты информации следует проанализировать цели создания автоматизированной системы, выявить информацию, подлежащую обработке, рассмотреть нормативно правовые акты, методические документы и национальные стандарты подлежащие выполнению и принять решение о необходимости создания системы защиты информации информационной системы, а также определение целей и задач защиты информации в информационной системе, основных этапов создания системы защиты информации информационной системы и функций по обеспечению защиты информации, содержащейся в информационной системе, обладателя информации (заказчика),оператора и уполномоченных лиц.

Цели создания данной автоматизированной системы определены в разделе 5 настоящего документа. Информация, подлежащая обработке исходя из пунктов 2 и 3 является противоправный видеоконтент и результаты его обработки данной автоматизированной системой, а также иная информация, связанная с её функционированием (конфигурация системы, содержание её элементов и т.д.)

В части выполнения данной автоматизированной системой задач, обозначенных на основании федеральных законов, законов субъектов Российской Федерации и иных правовых актов государственных органов (раздел 1) в соответствии с Федеральным законом от 27 июля 2006 г. № 149-ФЗ «Об информации, информационных технологиях и о защите информации» данная автоматизированная система является государственной информационной системой и подлежит выполнению требований Приказа ФСТЭК России № 17 от 11 февраля 2013 года. Также ввиду наличия при реализации данной автоматизированной системы процедуры разработки программного обеспечения в соответствии разделом 1 ГОСТ 56939 необходимо его выполнение с целью нейтрализации уязвимостей программного кода.

Следующим этапом необходимо документально определить цели и задачи защиты информации в данной автоматизированной системе в соответствии с внутренними нормативными документами её обладателя. Основными этапами для создания системы защиты информации в соответствии с вышеопределёнными стандартами и нормативно-правовыми актами рекомендуется определить:

1. формирование требований к защите информации, содержащейся в информационной системе;
2. разработка системы защиты информации информационной системы;
3. реализация процедуры разработки безопасного ПО;
4. внедрение системы защиты информации информационной системы;
5. аттестация информационной системы по требованиям защиты информации (далее - аттестация информационной системы) и ввод ее в действие;
6. обеспечение защиты информации в ходе эксплуатации аттестованной информационной системы;
7. обеспечение защиты информации при выводе из эксплуатации аттестованной информационной системы или после принятия решения об окончании обработки информации.

Для корректного разделения ответственности и выполнения законодательства в части лицензируемых видов деятельность требуется определить и документально определить уполномоченными на то сотрудниками разработчика данной информационной системы следующих лиц: обладателя информации (заказчика), оператора и уполномоченных лиц (в том числе для обеспечения выполнения требований по защите информации с учетом действующего законодательства в части лицензируемых видов деятельности).

В части классификации автоматизированной системы рекомендуется использовать следующие материалы:

1. На основании раздела 3 данного документа и наличия существующего процесса ручной реализации процесса выполняемого данной автоматизированной системой в соответствии с приложением №1 Приказа № 17 уровень значимости информации определить как низкий по причине возможности выполнения процессов данной автоматизированной системы с недостаточной эффективностью в результате нарушения одного из ключевых свойств безопасности информации.
2. По причине использования данной автоматизированной системы на объектах одного органа государственной власти и отсутствия сегментов территориальных органах, представительствах, филиалах, подведомственных и иных организациях масштаб признать как объектовый.
3. По результатам признания уровня значимости обрабатываемой информации низким, а масштаба объектовым, в соответствии с приложением № 1 Приказа № 17 минимально необходимый класс защищенности автоматизированной системы «Окулус» признать третий.

По результат работ необходимо оформить актом классификации.

Для корректного определения угроз безопасности информации необходимо выполнить моделирование угроз в соответствии с «Методикой оценки угроз безопасности информации» утвержденной ФСТЭК России 5 февраля 2021 года для

непосредственно архитектуры информационной системы и ЦОД в котором её планируется разместить, а также иные методики в соответствии с ГОСТ 56939 для определения угроз уровня разрабатываемого программного кода. Под моделированием угроз безопасности информации понимается процесс формирования модели угроз безопасности информации. Моделирование угроз безопасности информации выполняют с целью выявления потенциальных угроз безопасности информации, которые возникают вследствие применения АС и вызваны его функциональными особенностями (например, из-за ошибок проектирования, эксплуатации, обслуживания), и уточнения проекта архитектуры автоматизированной системы до реализации. Моделирование угроз безопасности информации выполняется путем применения методологии моделирования (перечисления) угроз безопасности информации.

Существующие методологии моделирования угроз безопасности информации, как правило, позволяют перечислять угрозы безопасности информации на основе анализа:

- потоков данных, передаваемых между компонентами программы и (или) элементами среды ее эксплуатации;
- перечня (библиотеки) типовых угроз безопасности информации;
- деревьев угроз безопасности информации.

В зависимости от используемой методологии, исходными данными для моделирования угроз безопасности информации являются:

- информация, связанная с типовыми сценариями компьютерных атак и типовыми угрозами безопасности информации, актуальными для разрабатываемой АС;
- сценарии использования разрабатываемой АС и предъявляемые пользователями требования к ней;
- сведения о проекте архитектуры АС (предполагаемые компоненты программы и их интерфейсы, концепция их совместного функционирования, перечень заимствованных у сторонних разработчиков ПО компонентов).

В зависимости от опыта и практических навыков работников, осуществляющих моделирование угроз безопасности информации, ими может использоваться исходная информация одного или нескольких типов. На ранних стадиях внедрения мер следует использовать как минимум информацию, связанную с типовыми сценариями компьютерных атак и типовыми угрозами безопасности информации, актуальными для разрабатываемой АС. По мере получения опыта и практических навыков в моделировании угроз безопасности информации исходную информацию следует дополнять сценариями использования разрабатываемого ПО, требованиями пользователей к разрабатываемому ПО, сведениями о проекте архитектуры программы.

Типовые сценарии компьютерных атак и угрозы безопасности информации следует анализировать с целью определения их применимости к разрабатываемому ПО с учетом его характеристик (например, используемые при разработке языки программирования и технологии) и характеристик предполагаемой среды его эксплуатации.

В качестве источников информации, содержащих типовые сценарии компьютерных атак и угрозы безопасности информации, можно привести: Банк данных угроз безопасности информации ФСТЭК России, публикации проекта Open Web Application Security Project (OWASP), например, публикация «Top 10 Most Critical Web Application Security Risks».

Сценарии использования разрабатываемой АС и требования к ней предъявляемые пользователями, следует анализировать с целью выявления угроз безопасности информации, связанных с выполнением этих требований и сценариев. Анализ следует выполнять с учетом разработанной модели нарушителя.

Выявление угроз безопасности информации на основе сведений о проекте архитектуры программы выполняется путем анализа потоков данных, передаваемых между компонентами программы и (или) элементами среды ее эксплуатации, или информации об известных уязвимостях в заимствованных у сторонних разработчиков ПО компонентов.

При подготовке к реализации мер необходимо:

1. исследовать существующие процессы в границах области действия мер, связанные с моделированием угроз безопасности информации;
2. выбрать (уточнить) и описать методологию моделирования (перечисления) угроз безопасности информации; перечисления) угроз безопасности информации, При описании используемой методологии моделирования угроз безопасности информации разработчик может дать ссылку на источник информации, содержащий описание методологии моделирования угроз безопасности информации.
3. выбрать и установить в среду реализации АС инструментальные средства для реализации меры по защите информации;
4. определить порядок сбора исходной информации, необходимой для выполнения моделирования угроз, с учетом необходимости выполнения следующих типовых действий:
   - собрать информацию об области применения разрабатываемой АС и типе обрабатываемой информации;
   - собрать сведения о проекте архитектуры (предполагаемые компоненты и их интерфейсы, концепция их совместного функционирования) и сведения об элементах среды эксплуатации, с которыми должно интегрироваться (совместно функционировать) разрабатываемая АС;
   - сформировать перечень заимствованных у сторонних разработчиков ПО компонентов, предполагаемых к использованию при разработке ПО;
5. определить порядок разработки и документирования модели нарушителя. Разработка модели нарушителя выполняется с целью определения видов нарушителей, их мотивации и возможностей по реализации угроз безопасности информации. Модель нарушителя в дальнейшем используется при выявлении угроз безопасности

информации, которые могут возникнуть вследствие применения ПО. При разработке модели нарушителя следует учитывать:

- область применения (например, класс защищенности информационной системы, в которой планируется использование разрабатываемой АС);
- тип информации, обрабатываемой АС (например, персональные данные, информация, содержащая сведения, составляющие государственную тайну или общедоступные данные).

Разработка модели нарушителя с учетом особенностей разрабатываемой АС и среды его эксплуатации является более предпочтительным, чем использование ранее созданных и немодифицированных моделей нарушителей, разработанных без учета этих особенностей.

6. определить порядок анализа защищенности заимствованных у сторонних разработчиков ПО компонентов, предполагаемых к использованию при разработке ПО, с учетом необходимости выполнения следующих типовых действий:
   - оценить предлагаемые к использованию при разработке ПО компоненты, заимствованные у сторонних разработчиков ПО, и выбрать компоненты, использование которых не приведет к ухудшению общей защищенности разрабатываемого ПО;
   - документировать и поддерживать в актуальном состоянии (т.е. при изменении используемых сторонних компонентов и/или их версий, соответствующую информацию следует задокументировать) результаты анализа и обоснование выбора;
7. определить порядок идентификации и документирования угроз безопасности информации с учетом необходимости выполнения следующих типовых действий:

– в соответствии с выбранной методологией моделирования угроз безопасности информации идентифицировать и документировать угрозы безопасности информации;
– проверить формулировки документированных угроз безопасности информации с точки зрения их адекватности;

При документировании списка выявленных потенциальных угроз безопасности информации для каждой выявленной угрозы безопасности информации следует указывать: уникальный идентификатор и формулировку угрозы безопасности информации. Для документирования угроз безопасности информации рекомендуется использовать созданные шаблоны моделей угроз безопасности информации (при их наличии).

8. определить порядок обработки документированных угроз безопасности информации с учетом необходимости выполнения следующих типовых действий:
   – проанализировать каждую документированную угрозу безопасности информации с целью определения стратегии ее обработки;
   – для каждой угрозы безопасности информации выполнить документирование стратегии ее обработки;
   – проверить формулировки документированных стратегий обработки угроз безопасности информации с точки зрения их адекватности;
9. определить процедуру периодического анализа и пересмотра документированных угроз безопасности информации с учетом необходимости выполнения следующих типовых действий:
   – определить события, при наступлении которых выполняется анализ и пересмотр документированных угроз безопасности информации;

– при наступлении событий анализировать документированные угрозы безопасности информации и пересматривать выбранную стратегию их обработки;
– уточнять документацию разработчика ПО (проект архитектуры программы, планы тестирования программы) по результатам анализа и пересмотра документированных угроз безопасности информации.

Документированные угрозы безопасности информации следует периодически пересматривать и уточнять, руководствуясь актуальной информацией, связанной с типовыми сценариями компьютерных атак и угрозами безопасности информации, сценариями использования разрабатываемой АС и требованиями к нему, предъявляемыми пользователями, проектом архитектуры программы.

1. определить общую структуру процесса моделирования угроз безопасности информации, включая общий перечень процедур и действий, фиксируемые результаты, время начала и временные рамки процедур и действий;
2. назначить работников, ответственных за реализацию мер защите информации, ознакомить их с документацией, относящейся к реализации меры.

При реализации мер защиты необходимо:

1. выполнить сбор исходной информации, необходимой для выполнения моделирования угроз;
2. с учетом полученной информации выполнить разработать модель нарушителя;
3. выполнить анализ защищенности заимствованных у сторонних разработчиков ПО компонентов;
4. идентифицировать и документировать угрозы безопасности информации;

5. обработать документированные угрозы безопасности информации и уточнить планы тестирования программы и проект архитектуры программы с учетом стратегий, выбранных при обработке угроз безопасности информации:
   - выявить возможные проектные решения, направленные на нейтрализацию выявленных угроз безопасности информации;
   - проанализировать выявленные проектные решения с учетом их реализуемости и оценки экономической целесообразности;
   - определить используемое при разработке проектное решение, уточнить проект архитектуры программы и сформулировать соответствующее требование по безопасности, предъявляемое к АС (при необходимости);
   - включить в план тестирования программы тесты, проверяющие нейтрализацию документированных угроз безопасности информации;
6. периодически анализировать и пересматривать документированные угрозы безопасности информации.

Как результат выполнения классификации автоматизированной системы выраженной в акте классификации, а также моделирования угроз как с точки зрения и архитектуры, так и с точки зрения исходного кода автоматизированной системы должны быть представлены требования к системе защиты информации содержащие меры определенные в Приказе № 17 минимально 3 класса защищенности, а также процедуры направленные на минимизацию актуальных угроз безопасности информации, в том числе посредством выполнения ГОСТ 56939.

Требования к системе защиты информации информационной системы включаются в техническое задание на создание информационной системы и (или) техническое задание (частное техническое задание) на создание системы защиты

информации информационной системы, разрабатываемые с учетом ГОСТ 34.602, ГОСТ Р 51583 и ГОСТ Р 51624, и должны в том числе содержать:

1. цель и задачи обеспечения защиты информации в информационной системе;
2. класс защищенности информационной системы;
3. перечень нормативных правовых актов, методических документов и национальных стандартов, которым должна соответствовать информационная система;
4. перечень объектов защиты информационной системы;
5. требования к мерам и средствам защиты информации, применяемым в информационной системе;
6. стадии (этапы работ) создания системы защиты информационной системы;
7. требования к поставляемым техническим средствам, программному обеспечению, средствам защиты информации;
8. функции заказчика и оператора по обеспечению защиты информации в информационной системе;
9. требования к защите средств и систем, обеспечивающих функционирование информационной системы (обеспечивающей инфраструктуре);
10. требования к защите информации при информационном взаимодействии с иными информационными системами и информационно-телекоммуникационными сетями, в том числе с информационными системами уполномоченного лица, а также при применении вычислительных ресурсов (мощностей), предоставляемых уполномоченным лицом для обработки информации.

При определении требований к системе защиты информации информационной системы учитываются положения политик обеспечения информационной безопасности обладателя информации (заказчика), а также

политик обеспечения информационной безопасности оператора и уполномоченного лица в части, не противоречащей политикам обладателя информации (заказчика).

В случае создания информационной системы, функционирование которой предполагается на базе информационно-телекоммуникационной инфраструктуры центра обработки данных, дополнительно определяются требования по защите информации, подлежащие реализации в информационно-телекоммуникационной инфраструктуре центра обработки данных.

11.Ожидаемый эффект реализации Концепции

В результате реализации Концепции АС ОКУЛУС ожидается:

- сокращение трудозатрат на ручное выявление признаков нарушений законодательства РФ в изображениях и видеоматериалах;
- повышение качества выявления признаков нарушений законодательства РФ в изображениях и видеоматериалах;
- снижение ошибок и исключение человеческого фактора в рамках поиска и распознавания ЗИ;
- увеличение скорости обработки информации и как следствие увеличение пропускной способности процесса выявления признаков нарушений законодательства РФ в изображениях и видеоматериалах.