Оглавление

[ВВЕДЕНИЕ 2](#_Toc192932590)

[1. Теоретическая часть 4](#_Toc192932591)

[1.1. Терминология по проекту и глоссарий 4](#_Toc192932592)

[1.2. Распределение ролей и работы 4](#_Toc192932593)

[2. Практическая часть 7](#_Toc192932594)

[2.1. Архитектура ПО 7](#_Toc192932595)

[2.2. Разработка проекта по ролям 7](#_Toc192932596)

[2.3. Контроль выполнения плана 8](#_Toc192932597)

[Заключение 8](#_Toc192932598)

# ВВЕДЕНИЕ

Компьютерное зрение становится все более важным направлением исследований и разработок. OpenCV предоставляет мощные инструменты для реализации базовых и продвинутых алгоритмов обработки изображений, что делает его популярным выбором среди разработчиков.

Актуальность – добавить

С развитием технологий автоматизации и искусственного интеллекта растет спрос на системы, способные обрабатывать и интерпретировать визуальную информацию.

OpenCV является одной из наиболее популярных библиотек для обработки изображений благодаря своей открытости, доступности и обширному сообществу разработчиков. Это облегчает внедрение новых решений и ускоряет процесс разработки.

Основная цель проекта заключается в создании программного обеспечения, способного автоматически определять и классифицировать объекты на основе их визуальных характеристик.

Задачи

Обьект и предмет исследования

# ТЕОРИТИЧЕСКАЯ ЧАСТЬ

# Терминология по проекту и глоссарий

Задача состоит в том, чтобы разработать приложение, которое будет принимать изображение в качестве входных данных и возвращать информацию о наличии определенных объектов на этом изображении. Для упрощения мы ограничимся распознаванием таких объектов, как круги, квадраты и треугольники.

Используемые технологии:

Python – основной язык программирования.

OpenCV – библиотека для обработки изображений и компьютерного зрения.

NumPy – библиотека для работы с многомерными массивами и матрицами.

# Распределение ролей и работы

Оконешников Родион – Специалист по данным.

Роль:

Сбор, подготовка и предварительная обработка данных для обучения модели.

Основные задачи:

FRONT

Сбор данных: Сбор изображений объектов, которые будут использоваться для тренировки модели. Это включает поиск и скачивание изображений из открытых источников, а также, возможно, создание собственных снимков.

Организация данных: Создание структуры каталогов для хранения изображений, разделение данных на тренировочные, валидирующие и тестовые наборы.

Предварительная обработка: Преобразование изображений (обрезка, изменение размеров, поворот, нормализация яркости и цвета). Использование методов предварительной фильтрации, таких как шумоподавление, улучшение контрастности и другие операции, повышающие качество входных данных.

Поддержка разметки: Помощь второму участнику в процессе разметки изображений (если потребуется)

Былчахов Алексей – Специалист по разметке. - FRONT

Роль:

Разметка изображений для обучения модели.

Основные задачи:

Выбор инструмента для разметки: Определение подходящего программного обеспечения для разметки изображений (например, LabelImg, VGG Image Annotator и др.). Настройка интерфейсов и рабочих процессов.

Разметка объектов: Нанесение меток на объекты на изображениях, указание границ рамкой (bounding box) или контурной областью (segmentation mask). Выбор категорий объектов и поддержание согласованности разметки.

Проверка качества: Периодическая проверка правильности нанесённых меток, выявление ошибок и внесение необходимых правок.

Создание аннотаций: Генерация файлов аннотаций в нужном формате (например, XML для YOLO, JSON для COCO и др.), совместимых с используемым фреймворком для обучения.

Мандарова Артем – Инженер машинного обучения.

Роль:

Настройка и обучение модели, тестирование и интеграция решения.

Основные задачи:

Выбор архитектуры модели: Исследование и выбор подходящей архитектуры для распознавания объектов (YOLO, SSD, Faster R-CNN и др.) исходя из специфики задачи и характеристик данных.

Настройка гиперпараметров: Подбор значений гиперпараметров модели (количество эпох, размер батча, оптимизатор, скорость обучения и т.д.) для достижения наилучших результатов.

Обучение модели: Запуск процесса обучения модели на размеченном наборе данных, мониторинг прогресса и контроль за переобучением/недообучением.

Оценка и тестирование: Проведение оценки модели на валидирующих и тестовых наборах данных, измерение ключевых метрик (точность, полнота, F1-score и др.), анализ результатов.

Оптимизация модели: Поиск способов увеличения точности и уменьшения времени выполнения модели (например, использование прецессинга, оптимизированных архитектур, ускорителей вроде CUDA).

Интеграция: Интеграция готовой модели в приложение или веб-сервис для автоматической обработки новых изображений и вывода предсказанных меток.

Документация: Описание всех этапов работы, создание документации по проекту, включая инструкции по использованию модели.

* 1. Стек технологий

1. OpenCV (Open Source Computer Vision Library)

Назначение: OpenCV — это мощная библиотека для работы с изображениями и видео, созданная специально для задач компьютерного зрения. Она содержит множество функций для обработки изображений, выделения особенностей, отслеживания объектов и даже глубокого обучения.

Что делает: С помощью OpenCV вы сможете легко загружать изображения, изменять их размеры, применять фильтры, находить края, выделять области интереса и многое другое. Эта библиотека поддерживает множество языков программирования, включая Python, C++, Java и другие.

Зачем нужна: OpenCV идеально подходит для задач предварительного анализа изображений перед распознаванием объектов. Например, вы можете использовать её для удаления шума, изменения контрастности, масштабирования изображений или выделения интересующих областей.

2. Python

Назначение: Python — это высокоуровневый язык программирования общего назначения, известный своей читаемостью и легкостью освоения. Он широко используется в научных исследованиях, разработке ПО и анализе данных.

Что делает: Python позволяет писать чистый и понятный код, который легко поддерживать. Благодаря большому количеству библиотек и модулей, таких как NumPy, Pandas и Matplotlib, этот язык отлично подходит для работы с данными, машинного обучения и визуализацией.

Зачем нужен: Python станет основным языком для написания скриптов, обработки данных и работы с OpenCV. Вы будете использовать его для написания основного кода проекта, настройки и вызова функций OpenCV, а также для работы с обучающими данными.

3. NumPy (Numeric Python)

Назначение: NumPy — это библиотека для работы с многомерными массивами и матрицами в Python. Она обеспечивает эффективные вычислительные операции с числами и векторами, что делает её незаменимой для обработки больших объемов данных.

Что делает: NumPy позволяет вам хранить и манипулировать данными в виде массивов, что значительно ускоряет выполнение арифметических операций. Многие функции OpenCV работают именно с массивами NumPy, поэтому эта библиотека тесно связана с ними.

Зачем нужна: NumPy необходима для представления изображений в виде матриц чисел (пикселей), а также для проведения математических операций над этими матрицами. Без неё сложно было бы эффективно обрабатывать данные и передавать их в OpenCV.

Таким образом, комбинация OpenCV, Python и NumPy обеспечит основу для эффективного распознавания объектов на изображениях.

# Практическая часть

# Архитектура ПО

# Разработка проекта по ролям

# Контроль выполнения плана

# Заключение