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 – Компьютерного зрение (Computer Vision, CV) является одной из наиболее динамично развивающихся областей искусственного интеллекта. OpenCV (Open Source Computer Vision Library) — это открытая библиотека для работы с компьютерным зрением, которая активно используется как в академических исследованиях, так и в коммерческих проектах. Рассмотрим актуальность OpenCV в контексте современных тенденций развития технологий. Современные приложения компьютерного зрения все чаще используют технологии глубокого обучения (Deep Learning). OpenCV используется в самых разных отраслях благодаря своей универсальности. OpenCV поддерживает множество операционных систем (Windows, Linux, macOS) и архитектур (x86, ARM), что делает его доступным для разработки как настольных, так и мобильных приложений. Библиотека также совместима с различными языками программирования, включая Python, C++, Java и MATLAB. Это обеспечивает широкую доступность и удобство использования.

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

Для процесса обнаружения простых объектов на изображениях с использованием методов компьютерного зрения включает в себя – Алгоритмы предварительной обработки изображения, методы сегментации и выделения объектов, техники классификации и распознавания объектов, а также взаимодействия между алгоритмами Opencv и использованием десктопного приложения.

Сперва разрабатывается алгоритм для обнаружения объектов

Исследовать и реализовать методы обнаружения объектов используя классические методы такие как контурный анализ, поиск границ, сопоставления шаблонов и цветовой сегментации. А также используя библиотеки и модели opencv

Создания интерфейса для приложения

Разработать удобный графический интерфейс пользователя для взаимодействия с алгоритмами обнаружения объектов. Реализовав следующие функции – Загрузка изображения, настройка параметров обработки, отображения результатов обнаружения объектов, сохранение результатов.

Тестирования и валидация

Проверка приложения используя простые объекты, например, геометрические фигуры или лица глаза. Предмет исследования — процесс обнаружения простых объектов на изображениях/видео с использованием OpenCV. Основные задачи включают разработку алгоритмов обнаружения, создание десктопного приложения, оптимизацию производительности и тестирование точности.

# 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.

Терминология:

Юзабилити - это степень удобства использования продукта или системы, насколько легко и эффективно пользователи могут достичь своих целей при взаимодействии с ним. Этот термин часто применяется в контексте разработки интерфейсов (веб-сайтов, приложений, программ), где важно обеспечить, чтобы пользователи могли быстро и без затруднений находить нужную информацию, выполнять задачи и получать желаемые результаты.

# Распределение ролей и работы

Оконешников Родион – Специалист по данным.

Роли:

* Сбор, подготовка и предварительная обработка данных для обучения модели.
* Разметка изображений для обучения модели.

Основные задачи:

1. Сбор данных:

* Сбор изображений объектов, используемых для тренировки модели.
* Поиск и скачивание изображений из открытых источников.

1. Организация данных:

* Создание структуры каталогов для хранения изображений.
* Разделение данных на тренировочные, валидирующие и тестовые наборы.

1. Предварительная обработка:

* Преобразование изображений (обрезка, изменение размеров, поворот, нормализация яркости и цвета).
* Применение методов предварительной фильтрации, таких как шумоподавление, улучшение контрастности и другие операции для повышения качества входных данных.

1. Разметка изображений:

* Выбор инструмента для разметки (LabelImg, VGG Image Annotator и др.).
* Настройка интерфейса и рабочих процессов.
* Нанесение меток на объекты на изображениях (рамки bounding box или контурные области segmentation mask).
* Проверка качества разметки, выявление ошибок и корректировка.
* Создание аннотационных файлов в нужных форматах (XML для YOLO, JSON для COCO и др.).

Былчахов Алексей — Специалист по FRONT

Роль:

* Специалист по фронтенду занимается разработкой пользовательского интерфейса (UI), который обеспечивает взаимодействие пользователей с системой распознавания объектов.

Основные задачи:

1. Проектирование и разработка UI:

* Создание макетов и прототипов интерфейса.
* Разработка интуитивно понятного и удобного пользовательского интерфейса для загрузки изображений, и отображения результатов работы модели.

1. Реализация функционала загрузки изображений:

* Разработка механизма загрузки изображений через интерфейс.
* Обеспечение поддержки различных форматов изображений.
* Отображение результатов работы модели:
* Интеграция фронтенда с бэкендом для получения результатов распознавания объектов от модели.
* Отображение результатов распознавания на загруженных изображениях (например, рамки вокруг распознанных объектов, подписи и другая информация).

1. Работа с API:

* Интеграция фронтенда с API, предоставляемым инженером машинного обучения для передачи изображений и получения результатов распознавания.
* Обработка запросов и ответов от сервера.

1. Тестирование и отладка:

* Тестирование разработанного интерфейса на наличие багов и ошибок.

1. Документирование:

* Составление технической документации по фронтенд-разработке.
* Предоставление инструкций по использованию интерфейса.

1. Совместная работа с командой:

* Координация действий с инженерами машинного обучения и специалистами по данным для обеспечения правильной интеграции компонентов системы.
* Взаимодействие с дизайнерами для соответствия интерфейса корпоративному стилю и требованиям юзабилити.

Мандаров Артем – Инженер машинного обучения.

Роль:

* Настройка и обучение модели, тестирование и интеграция решения.

Основные задачи:

1. 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.

1. Настройка гиперпараметров:

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

1. Обучение модели:

* Запуск процесса обучения модели на размеченном наборе данных.
* Мониторинг прогресса и контроль за переобучением/недообучением.

1. 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).

1. Интеграция:

* Интеграция готовой модели в приложение или веб-сервис.

1. Документирование:

* Составление технической документации по машинные обучения.
* Предоставление инструкций по использованию интерфейса.
  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

# Код на opencv которая будет обнаруживать простые объекты по их формам и определять, как фигуры

# 

# Рисунок 1 – код для распознания

# Сперва читаем изображение, преобразовав ее в оттенок серого для лучшего чтения «image = cv2.imred(image\_path), gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY).

# 

# Рисунок 2 – преобразование в бинарные изображения

# Применяя пороговое преобразование для получения бинарного изображения. THRESH\_BINARY\_INV инвертирует цвета, так что фигуры становятся белыми, а фон черным. Также находим контуры на изображении.

# Далее находит контур многоугольника чтобы определить количество вершин.

# 

# Рисунок 3 – Определение форм

# Определяем форму по количеству вершин. Для четырехугольников дополнительно проверяем соотношение сторон для различия между квадратом и прямоугольником.

# 

# Рисунок 4 – Отрисовка контуров

# Отрисовываем контур и добавляем метку с названием фигуры.

# 

# Рисунок 5 – Распознания простых фигур

# Для проверки мы используем изображение, где нарисованы простые фигуры.

# Архитектура ПО

# Разработка проекта по ролям

# Контроль выполнения плана

# Заключение