ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

**Московский институт электроники и математики**

Козьмин Андрей Викторович, группа БИВ247

Корсаев Артемий Батаевич, группа БИВ247

**БУДИЛЬНИК**

**С ТЕХНОЛОГИЕЙ РАСПОЗНАВАНИЯ ПОЗЫ ЧЕЛОВЕКА**

Междисциплинарная курсовая работа

по направлению 09.03.01 Информатика и вычислительная техника

студентов образовательной программы бакалавриата

«Информатика и вычислительная техника»

|  |  |  |
| --- | --- | --- |
| Студент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | подпись | И.О. Фамилия |

|  |  |  |
| --- | --- | --- |
| Студент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | подпись | И.О. Фамилия |

Руководитель

Старший преподаватель

|  |
| --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| И.О. Фамилия |

Москва 2024 г.

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

МОСКОВСКИЙ ИНСТИТУТ ЭЛЕКТРОНИКИ И МАТЕМАТИКИ

**ЗАДАНИЕ**

**на междисциплинарную курсовую работу бакалавра**

студенту группы БИВ247 Козьмину Андрею Викторовичу

1. Тема работы

Будильник с технологией распознавания позы человека.

1. Требования к работе.
   1. Устройство может воспроизводить звуковые сигналы.
   2. Устройство может отправлять/получать данные по Wi-Fi.
   3. Устройство может снимать видео.
2. Содержание работы
   1. Написание программы для отправки данных микроконтроллером по http.
   2. Написание программы для получения данных с камеры.
   3. Написание программы для воспроизведения звуковых сигналов.
   4. Проектирование и разработка устройства (электрическая схема и корпус).
3. Сроки выполнения этапов работы

Первый вариант МКР предоставляется студентом в срок до «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2024г.

Итоговый вариант МКР предоставляется студентом в срок до «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2024г.

|  |  |  |  |
| --- | --- | --- | --- |
| Задание выдано | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2024г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | А.М. Елисеенко |
|  |  | подпись руководителя |  |
| Задание было принято к исполнению | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2024г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | А.В. Козьмин |
|  |  | подпись студента |  |

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

МОСКОВСКИЙ ИНСТИТУТ ЭЛЕКТРОНИКИ И МАТЕМАТИКИ

**ЗАДАНИЕ**

**на междисциплинарную курсовую работу бакалавра**

студенту группы БИВ247 Корсаеву Артемию Батаевичу

1. Тема работы

Будильник с технологией распознавания позы человека.

1. Требования к работе.
   1. Программа может обрабатывать фотографии.
   2. Программа может отправлять/получать данные по Wi-Fi.
   3. Программа может сравнивать фотографии.
2. Содержание работы
   1. Написание программы загрузки данных о позе.
   2. Написание программы для передачи данных на устройство.
   3. Написание программы для сравнения с загруженной позой.
3. Сроки выполнения этапов работы

Первый вариант МКР предоставляется студентом в срок до «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2024г.

Итоговый вариант МКР предоставляется студентом в срок до «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2024г.

|  |  |  |  |
| --- | --- | --- | --- |
| Задание выдано | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2024г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | А.М. Елисеенко |
|  |  | подпись руководителя |  |
| Задание было принято к исполнению | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2024г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | А.Б. Корсаев |
|  |  | подпись студента |  |

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

МОСКОВСКИЙ ИНСТИТУТ ЭЛЕКТРОНИКИ И МАТЕМАТИКИ

**График выполнения междисциплинарной курсовой работы бакалавра**

студента группы Козьмина Андрея Викторовича

Тема работы

Будильник с технологией распознавания позы человека.

|  |  |  |  |
| --- | --- | --- | --- |
| Дата согласования первого варианта МКР | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2024г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | А.М. Елисеенко |
|  |  | подпись руководителя |  |
| Дата согласования итогового варианта МКР | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2024г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | А.В. Козьмин |
|  |  | подпись студента |  |

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

МОСКОВСКИЙ ИНСТИТУТ ЭЛЕКТРОНИКИ И МАТЕМАТИКИ

**График выполнения междисциплинарной курсовой работы бакалавра**

студента группы Корсаева Артемия Батаевича

Тема работы

Будильник с технологией распознавания позы человека.

|  |  |  |  |
| --- | --- | --- | --- |
| Дата согласования первого варианта МКР | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2024г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | А.М. Елисеенко |
|  |  | подпись руководителя |  |
| Дата согласования итогового варианта МКР | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2024г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | А.Б. Корсаев |
|  |  | подпись студента |  |

Содержание

[Введение 7](#_Toc200113807)

[Актуальность 7](#_Toc200113808)

[Цель работы 7](#_Toc200113809)

[Задачи 7](#_Toc200113810)

[Анализ существующих решений 7](#_Toc200113811)

[Используемые компоненты 8](#_Toc200113812)

[Глава I: Проектирование и реализация программного продукта 11](#_Toc200113813)

[Постановка задачи 11](#_Toc200113814)

[Анализ требований к программному продукту 11](#_Toc200113815)

[Проектирование интерфейса и программной архитектуры 11](#_Toc200113816)

[Выбор технологий 12](#_Toc200113817)

[Реализуемая архитектура согласно паттернам проектирования 12](#_Toc200113818)

[Реализация основных функций приложения 12](#_Toc200113819)

[Реализация функции преобразования изображения человека в вектора-кости. 12](#_Toc200113820)

[Тестирование и отладка 14](#_Toc200113821)

[Глава II: Анализ результатов и возможные доработки 15](#_Toc200113822)

[Оценка полученного результата 15](#_Toc200113823)

[Дальнейшие шаги по улучшению 15](#_Toc200113824)

[Возможности масштабирования 15](#_Toc200113825)

[Проблемы, с которыми мы столкнулись 15](#_Toc200113826)

[Заключение 15](#_Toc200113827)

[Выводы по результатам работы 15](#_Toc200113828)

[Достижение цели и выполнение задач 15](#_Toc200113829)

[Личный опыт и приобретённые навыки 16](#_Toc200113830)

[Перспективы продолжения работы 16](#_Toc200113831)

[Список использованных источников 16](#_Toc200113832)

[Приложения 16](#_Toc200113833)

[Приложение 1 - Листинг кода 16](#_Toc200113834)

Введение

Актуальность

Сон является неотъемлемой частью жизни каждого человека. Многие пользуются будильниками и изо дня в день просыпаются под однообразную музыку, что может раздражать или надоедать (рис. 1). В связи с этим возникает необходимость создания устройствa, позволяющего разнообразить ежедневную рутину пробуждения. В нашей работе мы разработали будильник, который отключается в момент, когда человек принимает необходимую позу. Это способствует более осознанному и активному пробуждению, а также уменьшает вероятность повторного засыпания.

Цель работы

Создать будильник и необходимое ПО для его функционирования, которые предоставляют следующий функционал: отключение музыки после принятия человеком необходимой позы.

Задачи

1. Разработать приложение для удобного и интуитивного взаимодействия с будильником.
2. Разработать ПО для обработки пользовательских поз.
3. Разработать устройство и написать ПО для его функционирования.

Анализ существующих решений

Проведя анализ открытых источников аналогичных решений найдено, не было. Но были выявлены решения, со схожими идеями:

1. Alarmy – android приложение, которое при срабатывании требует от пользователя совершение какой либо активности: решить математическую задачу, сделать фотографию заданного объекта или же небольшая физическая активность – потрясти телефон / дойти до определённого места.
2. Barcode Alarm Clock и QRAlarm – IOS и Android приложения соответственно, идея которых заключается в том, что для выключения звукового сигнала требуется просканировать определённый QR код, который пользователь заранее распечатал и поместил в помещении в определённое место.
3. Также существует множество будильников, основанных на инфракрасном датчике. Используя различные устройства – зачастую это пистолет – необходимо попасть инфракрасным лазером в приёмник, после чего будильник выключится.

Исходя из полученных данных, можно сделать вывод, что функционал нашего будильника не имеет прямых аналогов, но в то же время уже предпринимались попытки создания нестандартных решений в данной области.

Используемые компоненты

Было принято решение разрабатывать проект, используя компоненты, указанные в таблице 1. Выбор данных компонентов основан в большинстве на личном опыте, а также по другим причинам, которые указаны в таблице.

Таблица - Используемые компоненты.

|  |  |  |
| --- | --- | --- |
| Название и описание | Причины | Изображение |
| Микроконтроллер ESP32S3 – необходим для создания самого устройства будильника. | 1. Большое количество обучающего материала. 2. Достаточное количество GPIO. 3. Переходник на камеру. 4. Достаточная производительность для многозадачной работы. |  |
| Android studio – официальная среда разработки приложений на Android. | 1. Большое количество встроенных инструментов для разработки. 2. Наличие опыта работы в данной среде. | Выпуск Android Studio Hedgehog: множество функций, направленных на  повышение эффективности разработки | AppMaster |
| ESP IDF – официальный фреймворк для разработки ПО для микроконтроллеров ESP32. | 1. Обширная документация. 2. Большое количество примеров. 3. Наличие всех необходимых инструментов для разработки. | Getting Started With ESP-IDF | Set Up Espressif IDE – RoboticWorx |
| FastAPI – фреймворк для разработки RESP API сервисов на python. | 1. Высокая производительность. 2. Понятная документация. 3. Большое количество примеров и статей. 4. Асинхронная обработка запросов. | Python API deployment with RStudio Connect: FastAPI | R-bloggers |

|  |  |  |
| --- | --- | --- |
| Название и описание | Причины | Изображение |
| Docker – инструмент для контейнеризации приложений. | 1. Наличие опыта использования данного инструмента. 2. Обширная документация. 3. Множество обучающих материалов и примеров. 4. Простота развёртывания приложения. | Что такое Docker? Структура и преимущества | Serverspace |
| OpenCV – мощная библиотека для обработки изображений. | 1. Наличие опыта использования этого инструмента. 2. Популярное решение для множества задач. 3. Хорошая производительность. |  |
| TensorFlow – фреймворк для обучения и использования моделей искусственного интеллекта. | 1. Имеется опыт использования данного фреймворка. 2. Предоставляет большой функционал для построения моделей, обрабатывающих изображения. | TensorFlow — Википедия |
| Git – самая популярная система контроля версий. | 1. Большая распространённость. 2. Имеется опыт использования этой системы контроля версий | Git | Git | Opsera Ecosystem |

Глава I: Проектирование и реализация программного продукта

Постановка задачи

Требуется разработать прототип устройства и сервер. Устройство будет включаться в заданное пользователем время, делать снимки человека и отправлять их на сервер. Сервер будет анализировать изображение на то, в правильной ли позе находится человек. После чего отправлять вердикт на устройство, которое в свою очередь будет отключать звуковой сигнал, если человек принял правильную позу.

Анализ требований к программному продукту

Это who?

Проектирование интерфейса и программной архитектуры

Было принято разделить задачу на две части: сервер и устройство. Как было описано в постановке задачи на сервере изображения будут обрабатываться, а устройство будет реализовывать вердикт сервера. Для решения такой задачи на сервере был реализован HTTP сервер, который получает POST запрос с изображением, после чего прогоняет его через модель MoveNet. На выходе из модели получается набор векторов-костей, которые сравниваются с эталонным и в качестве ответа сервер отправляет пустой ответ с кодом 200, если человек в правильной позе или пустой ответ с кодом 400 если человек в неправильной позе.

Программная архитектура на устройстве выглядит следующим образом. Основная задача была разбита на самостоятельные подзадачи, которые выполняются параллельно и общаются между собой посредством каналов, через которые передаются команды и данные. Для этого было принято решение использовать операционную систему FreeRTOS. Были созданы следующие подзадачи:

1. BLE – задача, отвечающая за взаимодействие с bluetooth low energy.
2. HTTP – задача, отвечающая за взаимодействие с сервером по HTTP протоколу.
3. LOGIC – задача, отвечающая за логику работы устройства. Она отправляет команды другим задачам.
4. CAM – задача, отвечающая за взаимодействие с камерой.
5. MUSIC – задача, отвечающая за проигрывание музыки.

Были так же реализованы отдельные модули, не являющиеся самостоятельными задачами, но необходимые для работы устройства:

1. WIFI – модуль отвечает за подключение устройства к wifi.
2. CHANNELS – модуль предоставляет каналы для общения между задачами.
3. STORAGE – модуль предоставляет удобный функционал для работы с энергонезависимой памятью.
4. SNTP – модуль отвечает за получение реального мирового времени.

Выбор технологий

Используемые технологии и причины их выбора были детально описаны во введении. Так же с ними можно ознакомится в таблице 1.

Реализуемая архитектура согласно паттернам проектирования

На устройстве была реализована архитектура Orchestration + Event Channel. За оркестрацию отвечает задача LOGIC, и все задачи обмениваются между собой данными и событиями по каналам.

На сервере реализован паттерн Chain of Responsibility. После получения POST запроса сервер прогоняет его через цепочку обработчиков, после чего возвращает итоговый результат.

Реализация основных функций приложения

Реализация функции преобразования изображения человека в вектора-кости.

Одной из важнейших функций обработки изображения является movenet(), которая отвечает за нахождение ключевых точек на изображении.



Рисунок листинг функции movenet()

Эта функция используется уже в get\_keypoints(), которая преобразует изображения в вид, необходимый для функции movenet(). Так, в список преобразований входит возможность получения на вход картинки как строки, символизирующий путь к изображению, так и с numpy-массивами. Вход преобразуется тип данный Tensor(), используемых библиотекой tensorflow, а затем проходит масштабирование Tensor(), который посылается уже в функцию movenet(), результат которой возвращается из get\_keypoints().

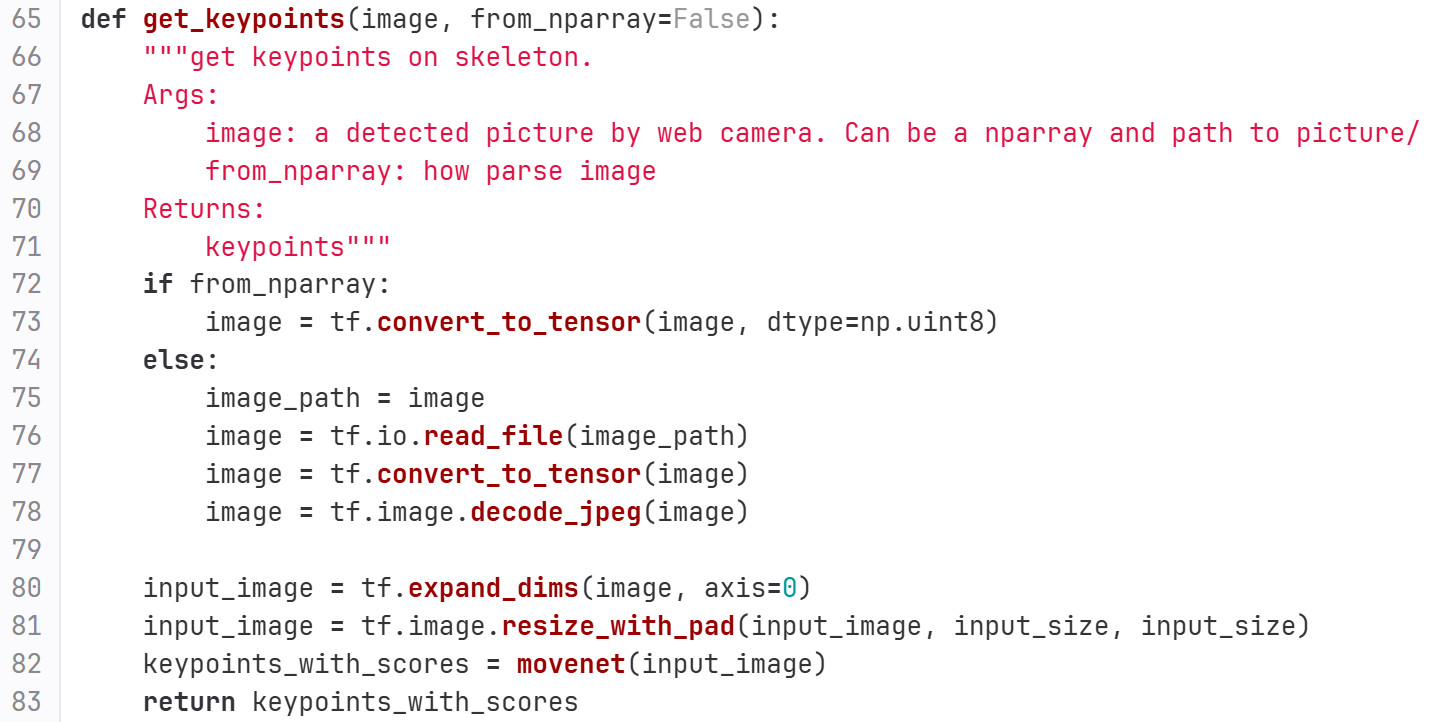


Рисунок - листинг функции get\_keypoints()

Одной из ключевых функций является keypoints\_and\_edges\_for\_display() (Приложение 1), которая на вход получает ключевые точки, возвращая информацию для представления позы на изображении в виде линий(костей).

Реализация HTTP сервера

HTTP сервер написан на языке программирования Python с использованием фреймворка FastAPI. Он реализует один эндпоинт, который ожидает POST запрос с изображением. Изображение передаётся в модель, после чего результат обработки изображения сравнивается с эталонным значением, после чего возвращается ответ клиенту – 200, если поза совпала и 400, если поза не совпадает. Эталонные значения получаются с помощью инъекции зависимостей (рис. 3).

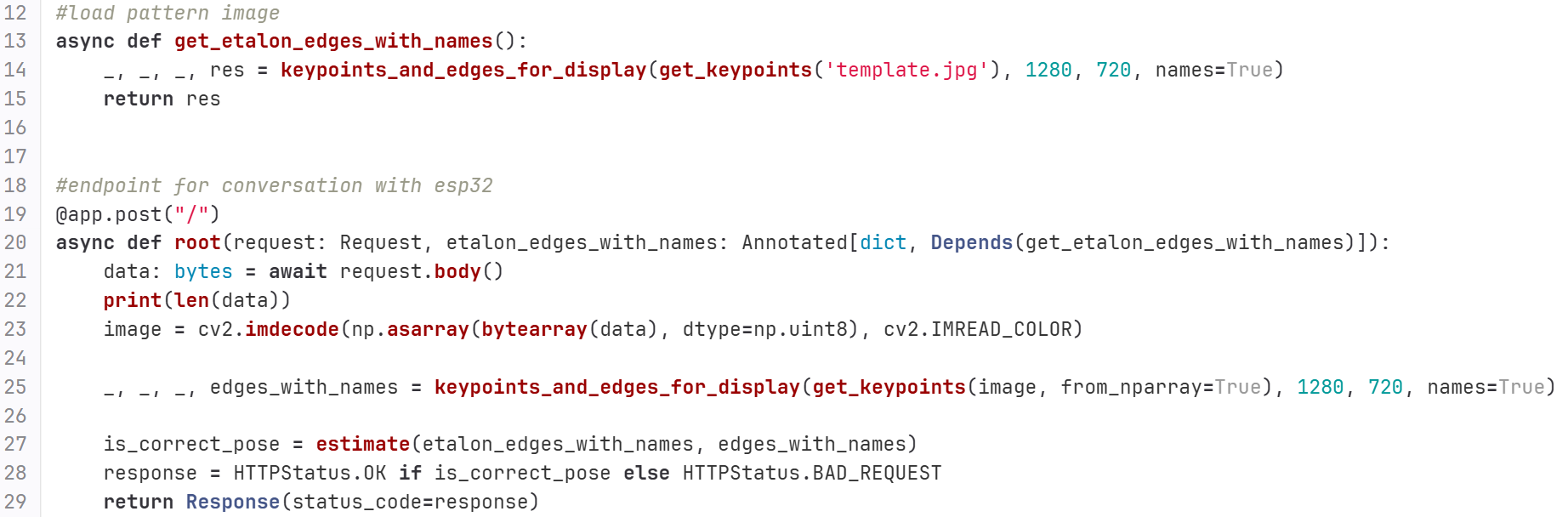


Рисунок – Реализация эндпоинта и инъекции зависимостей.

Так же для удобства и воспроизводимости запуска сервер был контейнеризирован с использованием Docker (рис. 4).



Рисунок 4 – Dockerfile для HTTP сервера.

Так же для менеджмента зависимостей используется специальный инструмент uv от компании astral-sh. Он позволяет сохранять конкретные версии используемых библиотек, автоматически управляет виртуальным окружением, а также предоставляет некоторые другие возможности. Все эти возможности в совокупности позволяют удобно работать с виртуальным окружением, зависимостями, версией python и т.п. Это в значительной степени ускоряет разработку программного обеспечения.

Тестирование и отладка

Для тестирования модели распознавания позы на начальном этапе была написана программа для её отладки. В качестве камеры, откуда берется поза, используется веб-камера компьютера.

Следующим этапом тестирование модели распознавания позы было использования камеры устройства на базе ESP32S3 вместо веб-камеры компьютера. (рассказать тест мобильного приложения и есп)

Глава II: Анализ результатов и возможные доработки

Оценка полученного результата

(12/10. А вообще, нужно что-то написать)

Дальнейшие шаги по улучшению

Планируется сделать загрузку пользователем собственного шаблона позы, поддержку распознавания нескольких поз на фотографии, улучшение графического интерфейса, а также улучшить функцию сравнения позы: например, чтобы функция оценивала не просто картинку, а видео длиною в 5 секунд.

Возможности масштабирования

(А вообще, нужно что-то написать)

Проблемы, с которыми мы столкнулись

Были проблемы с определением функции оценки позы и подбором порогового значения для неё. (А вообще, нужно что то написать)

Заключение

Выводы по результатам работы

(12/10. А вообще, нужно что-то написать)

Достижение цели и выполнение задач

Мы считаем, что разработка минимально жизнеспособного продукта проведена успешна. Проект имеет большой потенциал засчёт интеграции с системой “умный дом”. (нужно что-то еще написать)

Личный опыт и приобретённые навыки

Артемий Корсаев приобрел навыки работы в команде, работа с фреймворком FastAPI, а также разработкой с помощью инструментов tensorflow(А вообще, нужно что то написать)

Перспективы продолжения работы

У проекта огромный потенциал. Пока что он находится на стадии прототипа и у него присутствуют недостатки. В перспективе можно избавится от недочётов и перейти от прототипа к полноценному устройству.

Список использованных источников

1. ESP-IDF Examples Repository. URL: https://github.com/espressif/esp-idf/tree/master/examples (дата обращения: 06.06.2025).
2. ESP-IDF Programming Guide for ESP32-S3. URL: https://docs.espressif.com/projects/esp-idf/en/stable/esp32s3/index.html (дата обращения: 06.06.2025).
3. FastAPI Documentation. URL: https://fastapi.tiangolo.com/ (дата обращения: 06.06.2025).
4. Matplotlib Documentation. URL: https://matplotlib.org/stable/index.html (дата обращения: 06.06.2025).
5. Mido: MIDI Objects for Python. URL: https://mido.readthedocs.io/en/stable/ (дата обращения: 06.06.2025).
6. MoveNet: Ultra fast and accurate pose detection. URL: https://www.tensorflow.org/hub/tutorials/movenet?hl=ru (дата обращения: 06.06.2025).
7. NumPy Documentation. URL: https://numpy.org/doc/stable/ (дата обращения: 06.06.2025).
8. OpenCV Documentation. URL: https://docs.opencv.org/5.x/ (дата обращения: 06.06.2025).
9. TensorFlow Python API. URL: https://www.tensorflow.org/api\_docs/python/tf/all\_symbols (дата обращения: 06.06.2025).

Приложения

Приложение 1 - Листинг кода

