



Умный городской гид

Текунов Антон
Котов Алексей
Белогай Дарья
Грибов Дмитрий
Мясников Дмитрий

Введение

Проблема:

Посещение музеев, городов и новых мест не всегда является увлекательным. Отсутствие персонализированного опыта и интерактивности часто ведет к утрате интереса.

Примеры трудностей:

01

Посетителям трудно определить достопримечательности или экспонаты без пояснений.

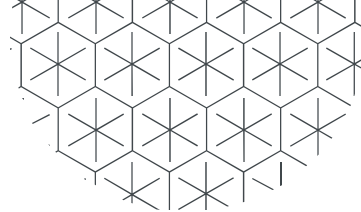
02

Недостаток информации о сфотографированных объектах ведет к упущению важных деталей.

03

Сложности в нахождении релевантных сведений о культурных и исторических объектах.

Актуальность проекта



01

Рост интереса к путешествиям

Увеличение числа туристов создает запрос на удобные инструменты взаимодействия с культурными объектами.

03

Доступность технологий

Использование компьютерного зрения упрощает доступ к информации о культурных объектах.

02

Потребность в интерактивности

Люди ищут новые способы вовлечения, которые делают посещение более увлекательным.

04

Ожидания мгновенной информации

Современные посетители хотят быстро получать сведения о достопримечательностях.



Цель и задачи проекта

Цель:

Повышение вовлеченности и улучшение опыта посещения музеев и городов через предоставление быстрых, интерактивных и персонализированных решений для определения достопримечательностей и объектов по фотографии.

Задачи:

1. Создать чат-бот в Telegram для определения достопримечательностей по фото.
2. Интегрировать модели компьютерного зрения для распознавания объектов.
3. Разработать базу данных с информацией о ключевых достопримечательностях.
4. Провести тестирование функционала бота с реальными объектами.

Техническая реализация

Используемая модель:

Для распознавания достопримечательностей используется готовая **предобученная модель**, которая была интегрирована в систему.

Преимущества использования готовой модели:

1. Ускоренная интеграция: модель готова к использованию и требует минимальной настройки.
2. Высокая точность предсказаний, так как модель уже обучена на обширном наборе данных.
3. Возможность дообучения или обновления модели при необходимости добавления новых объектов.

Общий процесс реализации:

1. Интеграция модели:

- Модель загружается с использованием модуля **model_loader**.
- Она предсказывает, какой объект изображен на фотографии, и возвращает соответствующий результат.

2. Разработка Telegram-бота:

- Бот разработан на Python с использованием библиотеки **python-telegram-bot**.
- Реализованы обработчики команд **/start** и сообщений с фотографиями. Пользователь отправляет фото, бот анализирует его и возвращает результат.

Ключевые элементы реализации:

- Telegram API: Управляет обменом сообщениями между пользователем и ботом.
- Модуль **model_loader**: Отвечает за загрузку модели и предсказание.
- Библиотеки Python: Используются для обработки данных, логирования и взаимодействия с платформами.
- Облачное хранилище: Хранит изображения и временные данные для обработки.

код с обработкой фотографии:

```
19 v async def handle_image(update: Update, context: CallbackContext) -> None:
20     """Обрабатывает изображение, которое прислал пользователь."""
21     try:
22         # Получаем фотографию, отправленную пользователем
23         file = await update.message.photo[-1].get_file()
24         file_path = await file.download_to_drive('image.jpg')
25
26         # Загружаем модель и предсказываем результат
27         model = load_model()
28         prediction = predict_image('image.jpg', model)
29
30         # Отправляем результат пользователю
31         await update.message.reply_text(f"На изображении: {prediction}")
32     except Exception as e:
33         # Обрабатываем ошибки и отправляем сообщение пользователю
34         await update.message.reply_text(f"Произошла ошибка: {str(e)}")
35     --
```

Работа бота

Процесс взаимодействия с ботом:

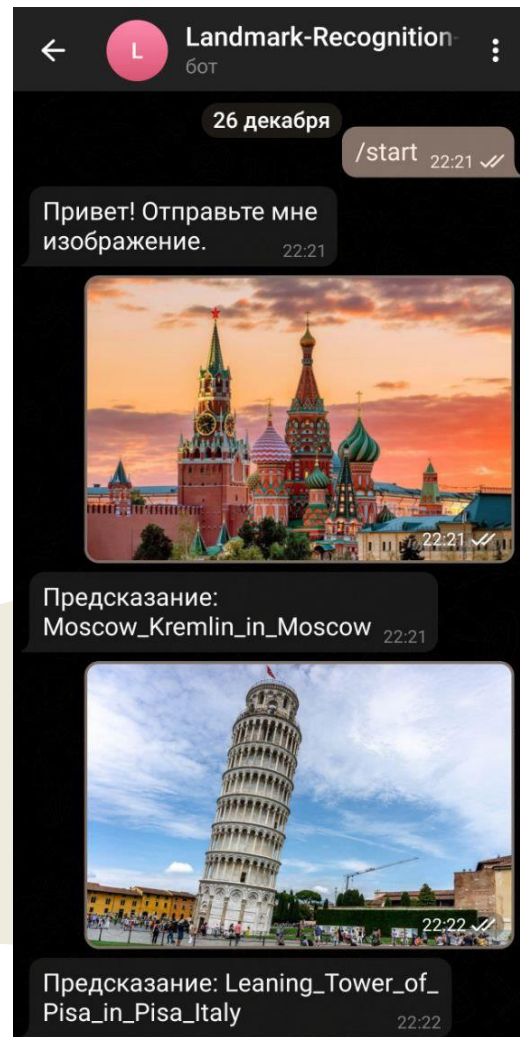
- Пользователь отправляет изображение.
- Бот скачивает его и передает в модель для анализа.
- После получения результата модель возвращает описание объекта (например, название достопримечательности), которое отображается пользователю.

Название:

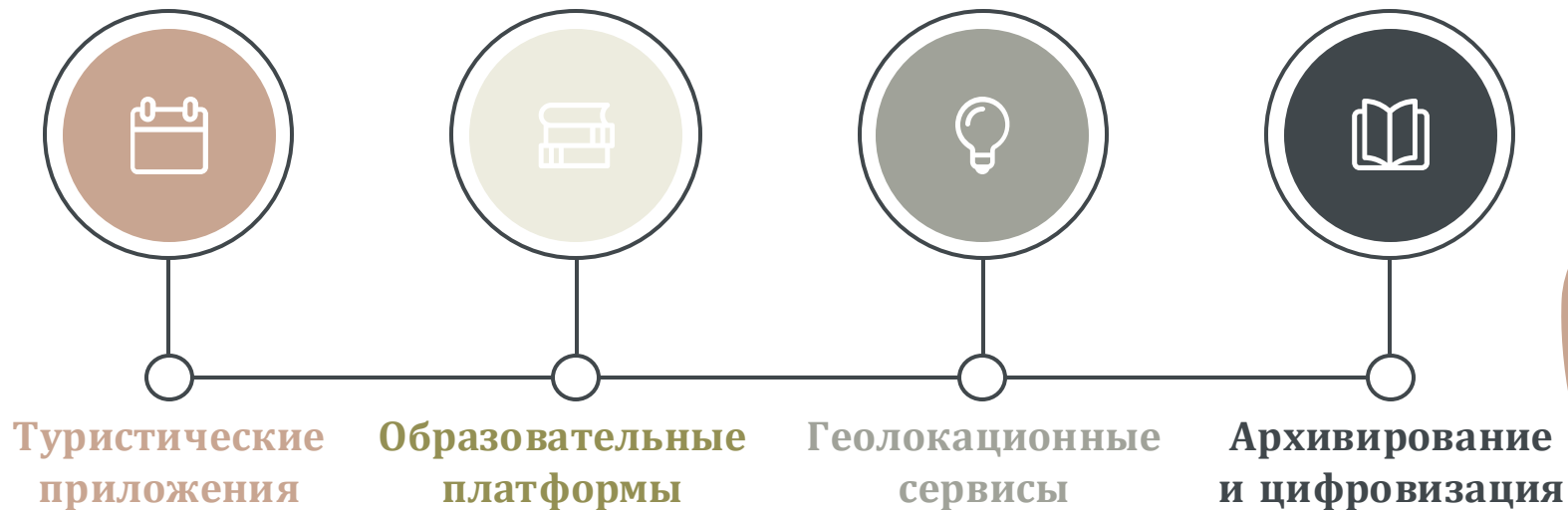
Landmark-Recognition-Bot
@Hackaton_Facts_bot

Поддерживаемые команды:

/start



Возможные области применения



Бизнес-модель проекта

Целевая аудитория:

- Туристы, путешественники и местные жители.
- Культурные учреждения и музеи.
- Городские администрации.

Стратегия продвижения:

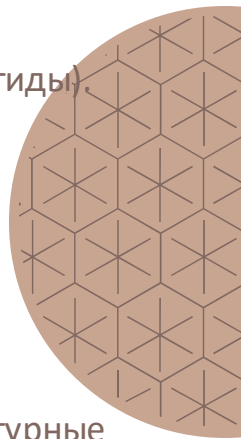
- Партнёрство с туристическими агентствами.
- Интеграция с популярными платформами (например, TripAdvisor).
- Реклама на тематических площадках и форумах.

Источник дохода:

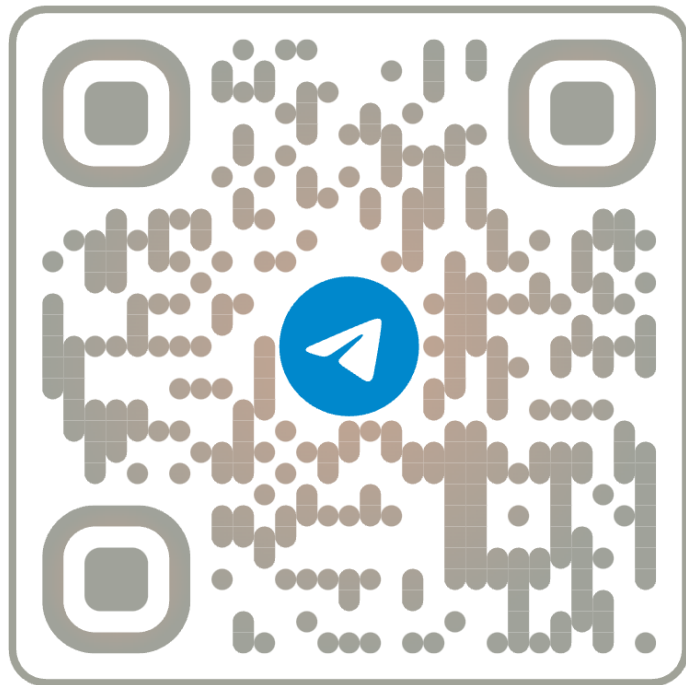
- Лицензионные **соглашения** с музеями и городскими администрациями.
- **Подписка** для туристов с расширенными функциями (например, подробные аудиогиды).
- **Реклама** локальных сервисов и бизнесов через бот.

Потенциальные рынки:

- Туристически популярные города и культурные объекты.
- Регионы с большим количеством музеев и памятников.



Наш ТГ-бот:



Ссылка на Git:

