

Поиск свободных парковочных мест

С помощью искусственного интеллекта

Проект подготовила команда НТК, в
составе:

Возяков Вячеслав, Быстрицкий Дмитрий,
Воронин Илья, Матвеева Дарья (by НТК)

Постановка задачи

Создать продукт, позволяющий в реальном времени анализировать данные с камеры видеонаблюдения для получения данных о свободных парковочных местах на парковке, передача информации с помощью телеграмм бота



Архитектура сервиса

(by НТК)



Разбивка программной части на задачи

получение картинки

(by НТК)

распознавание
парковочных мест

распознавание машин

распознавание
освободившихся мест

Формирование
сообщения для отправки
пользователю

Реализация

Основой для нашей программы мы
решили использовать (by HTK)

- предобученная модель Mask R-CNN
- предобученная модель OpenCV
- open-source моделью от Matterport

(by HTK)

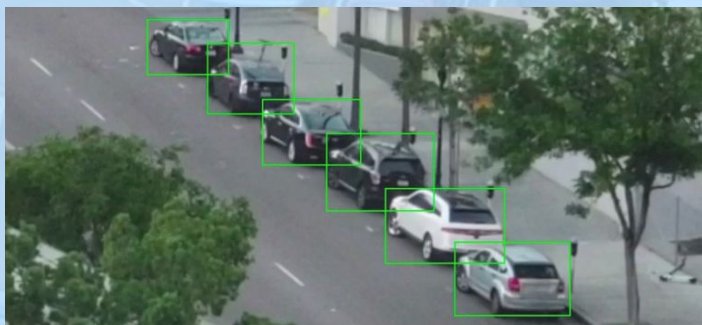
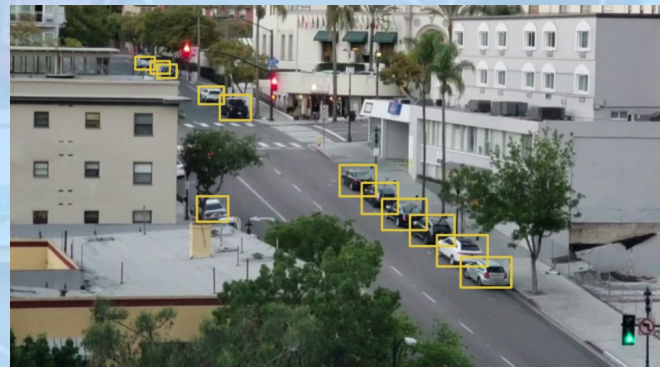
```
1 import numpy as np
2 import cv2
3 import mrcnn.config
4 import mrcnn.utils
5 from mrcnn.model import MaskRCNN
6 from pathlib import Path
7
8
9 # Конфигурация, которую будет использовать библиотека Mask-RCNN.
10 class MaskRCNNConfig(mrcnn.config.Config):
11     NAME = "coco_pretrained_model_config"
12     IMAGES_PER_GPU = 1
13     DEVICE = "cpu"
14     DETECTION_MIN_CONFIDENCE = 0.6
15
16
17 # Список объектов для обнаружения.
18 boxes = []
19
20 # Список объектов для обнаружения.
21 car_boxes = []
22
23 for i, box in enumerate(boxes):
24     # Обработка каждого объекта.
25     # ...
26
27 return np.array(car_boxes)
28
29
30 # Корневая директория проекта.
31 ROOT_DIR = Path(".")
32
33 # Директория с обученными весами.
34 MODEL_PATH = ROOT_DIR / "weights"
35
36 # Локальный путь к файлу с обученными весами.
37 COCO_MODEL_PATH = ROOT_DIR / "mask_rcnn_coco.h5"
38
39
40 # Загружаем датасет COCO при необходимости.
41 if not COCO_MODEL_PATH.exists():
42     mrcnn.utils.download_trained_weights(COCO_MODEL_PATH)
43
44 # Директория с изображениями для обработки.
45 IMAGE_DIR = ROOT_DIR / "images"
46
47 # Видеофайл или камера для обработки – вставьте значение 0, если нужно использовать камеру, а не видеофайл.
48 VIDEO_SOURCE = "test_images/parking.mp4" ## source of video
49
50 # Создаем модель Mask-RCNN в режиме вывода.
```

Что мы использовали

При создании продукта были использованы:

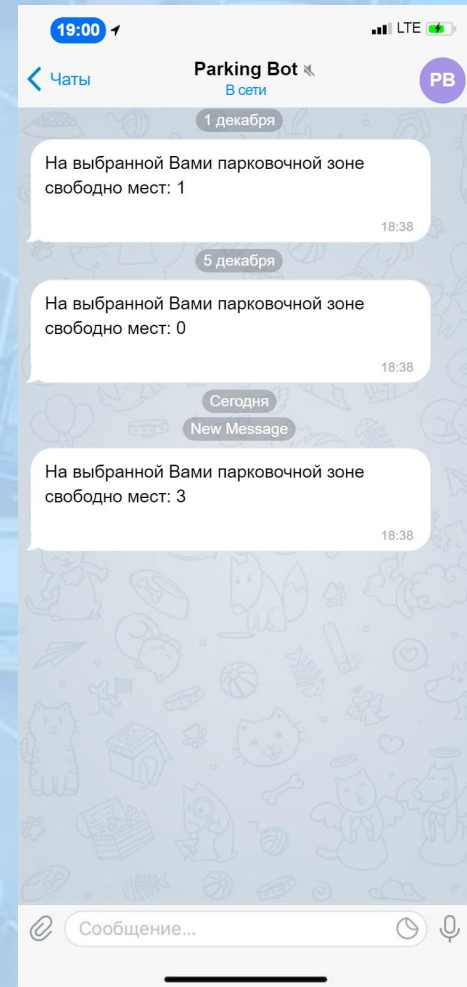
- Google collab (by НТК)
- Jupyter notebook
- Github
- PyCharm JetBrains Community ed
- Python
- aiogram
- Mask R-CNN
- датасет COCO (сокращение от Common Objects In Context) (by НТК)
- OpenCV


Практический принцип работы (by HTK)



Далее, по запросу
пользователь имеет
возможность получить
сообщение, содержащее
кол-во парковочных мест

(by НТК)





Перспективы разработки
или
Чем мы видим финальный продукт

(by НТК)



Спасибо за внимание!

Искренне Ваша, команда НТК

(by НТК)