

Секция 4

ОБРАБОТКА СИГНАЛОВ, ИЗОБРАЖЕНИЙ И ВИДЕО

УДК 004.032.26

СИСТЕМА УПРАВЛЕНИЯ ПАРКОВКАМИ НА ОСНОВЕ АНАЛИЗА ВИДЕОДАНЫХ

*магистрант А. В. ХОДОСЕВИЧ, канд. техн. наук, доц. Р. П. БОГУШ
(Полоцкий государственный университет, Беларусь)*

Аннотация. *Представлена клиент-серверная система для контроля и управления парковками на основе данных систем видеонаблюдения. Для сегментации парковочных мест разработан алгоритм, позволяющий автоматически определять координаты парковочных мест на изображении, а для их классификации используется сверточная нейронная сеть ResNet50. Клиентская часть системы представляет собой web-сайт, доступ к которому можно получить с помощью персонального компьютера или мобильного устройства.*

Ключевые слова: *парковочные места, сверточная нейронная сеть, Smart City.*

Для удаленного мониторинга за открытыми и закрытыми парковками чаще всего применяются системы видеонаблюдения, использующие IP-видеокамеры. Эффективность таких может быть существенно повышена за счет внедрения нового программного обеспечения, реализующего интеллектуальные функции обработки видеоданных [1]. Для парковок с большим количеством мест особенно актуальным является применение подобных систем. В [2] представлена структура построения такой системы, которая представляется перспективной и может быть интегрирована в комплексную систему управления городским имуществом «Smart City» («умный город»). Однако, для ее организации необходима реализация и исследование алгоритмов сегментации и классификации парковочных мест, разработка и реализация клиент-серверной архитектуры

С учетом специфики работы системы управления большими парковками к программному обеспечению сформированы следующие требования: работа в режиме реального времени; минимизация передаваемого трафика по каналам связи; кроссплатформенность; обеспечение стабильной работы при сбоях в сети передачи данных; возможность подключения большого числа камер наблюдения; централизованная архитектура системы видеонаблюдения.

Для сегментации парковочного места и определения его координат разработан алгоритм, который состоит из следующих основных этапов: обнаружение линий разметки парковки на изображении без автомобилей; бинаризация; связывание линий с использованием алгоритма Хафа; определение координат парковочных мест; сохранение координат и идентификаторов парковочных мест в базу данных. На рисунках 1–2 представлен пример некоторых этапов сегментации парковочных мест.

Для классификации парковочных мест используется достаточно быстродействующая и точная сверточная нейронная сеть ResNet50. Для ее обучения использована система машинного обучения ML.NET [3] и синтезированная база данных, пример размеченного изображения парковки для обучения.

Таким образом, в программной реализации выделены три этапа: обнаружение парковочных мест на кадре, распознавание автомобиля в видеопотоке, находящихся на парковочных местах, обновление статусов парковочных мест. Серверная часть системы включает хранение базы данных изображений парковок и координат локализованных парковочных мест для каждой из них, функционирование программного модуля для обнаружения свободных парковочных мест, хранение прайс-листа цен на услуги, хранение информации пользователей, обработку запросов и выдачу данных клиентской части. Обмен между компонентами осуществляется с помощью HTTP-запросов и текстового формата обмена данными – JSON. Для передачи видеопотока используется RTSP протокол. Программная реализация использует OpenCV для классических процедур обработки изображений.

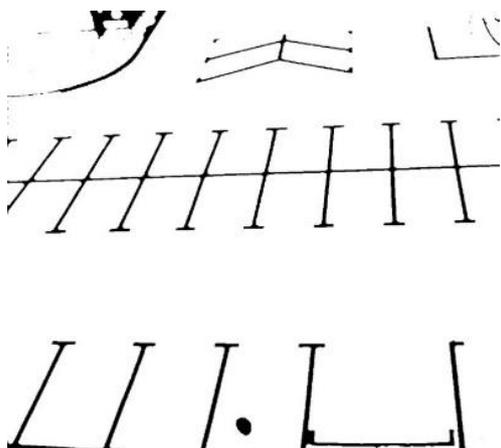


Рисунок 1 – Результат выделения линий и бинаризации



Рисунок 2 – Сегментированные места

Диаграмма классов серверной части системы представлена на рисунке 3. К основным классам относятся:

ResultData класс, хранящий результат анализа нейронной сети: Конструктор ResultData производит инициализацию свойств класса;

MLChecker класс для работы с моделью нейронной сети: Check определяет, находится ли автомобиль на входном фото;

Camera класс, реализующий работу с камерой: Start метод, запускающий поток для функции ReadFromCamera; End метод, останавливающий поток для ReadFromCamera; ReadFromCamera метод, реализующий фоновое чтение видеопотока;

Range класс «диапазона»: ToUp реализующий сортировку по возрастанию; ToDown реализующий сортировку по убыванию; In проверяет вхождение диапазона в другой диапазон;

Line класс «линии»: Clone копирует линию; GetOrientation возвращает ориентацию линии; ToUpOnX реализующий сортировку точек по X; ToUpOnY реализующий сортировку точек по Y; Length возвращает длину линии; GetAngle возвращает угол наклона линии; GetDistance возвращает расстояние от линии до линии; InLine проверяет, входит ли линия в линию;

LineEx класс-расширение класса «Line»; Unite объединяет близкие линии или те, которые входят друг в друга; CompareLines сравнивает линии по X/Y; RectSort реализующий сортировку массива точек; GetRects возвращает преобразованный массив полигонов из массива линий;

PrepareImage класс реализует работу с изображением: GetExt возвращает расширение изображения; Save сохранение изображения; ToCanny поиск граней с использованием Cv2; HoughLines поиск линий с использованием Cv2; Load загрузка изображения; ToMask возвращает маску изображения с использованием Cv2;

ImageComparer класс, реализующий сравнение изображений: IsFree метод определяющий, свободно ли место; GetImage возвращает картинку полигона из входного фото.

Клиентская часть представляет собой web-сайт, доступ к которому можно получить с помощью персонального компьютера или мобильного устройства, имеющих доступ в сеть Интернет, и обеспечивает основные возможности: вывод информации о парковках, существующих ценах, организации-собственнике; выбор места среди свободных при бронировании, временном интервале, автомобиля и парковки при бронировании; регистрация и авторизация пользователя; управление своими автомобилями; просмотр брони пользователя; управление таблицами парковок и цен для администратора сайта.

Взаимодействие с системой могут осуществлять три типа пользователей: системный администратор (обслуживающий персонал) серверной части, пользователь клиентской части и администратор клиентской части. Системный администратор имеет доступ к серверу, на котором установлены и запущены система управления базами данных и программный продукт, с возможностью прямого доступа к файлам системы. К основным его функциям относятся: настройка на стороне сервера; редактирование основного файла настроек проекта; настройка подключения

к базе данных и ее редактирование. Администратор клиентской части имеет возможности: добавления, изменения и удаления в базе данных информации о парковках и ценах; просмотра броней пользователей для мониторинга работы системы.

Для пользователя представляется: получение общей информации; ознакомление с актуальными ценами на бронирование; регистрация/авторизация в системе с возможностями изменения данных своего профиля; выбор парковки, свободного места на дату и время бронирования; просмотр и редактирование своего профиля; просмотр своих броней на парковках в системе.

Тестирование разработанного модуля выполнено с использованием персонального компьютера Google Chrome, Mozilla Firefox, Internet Explorer и мобильных устройств под управлением ОС Android. Для мобильных устройств под управлением Android тестирование выполнено с использованием РОСО Х3 в браузере устройства. Скриншоты страниц веб-приложения в мобильной адаптации представлены на рисунках 4–6.

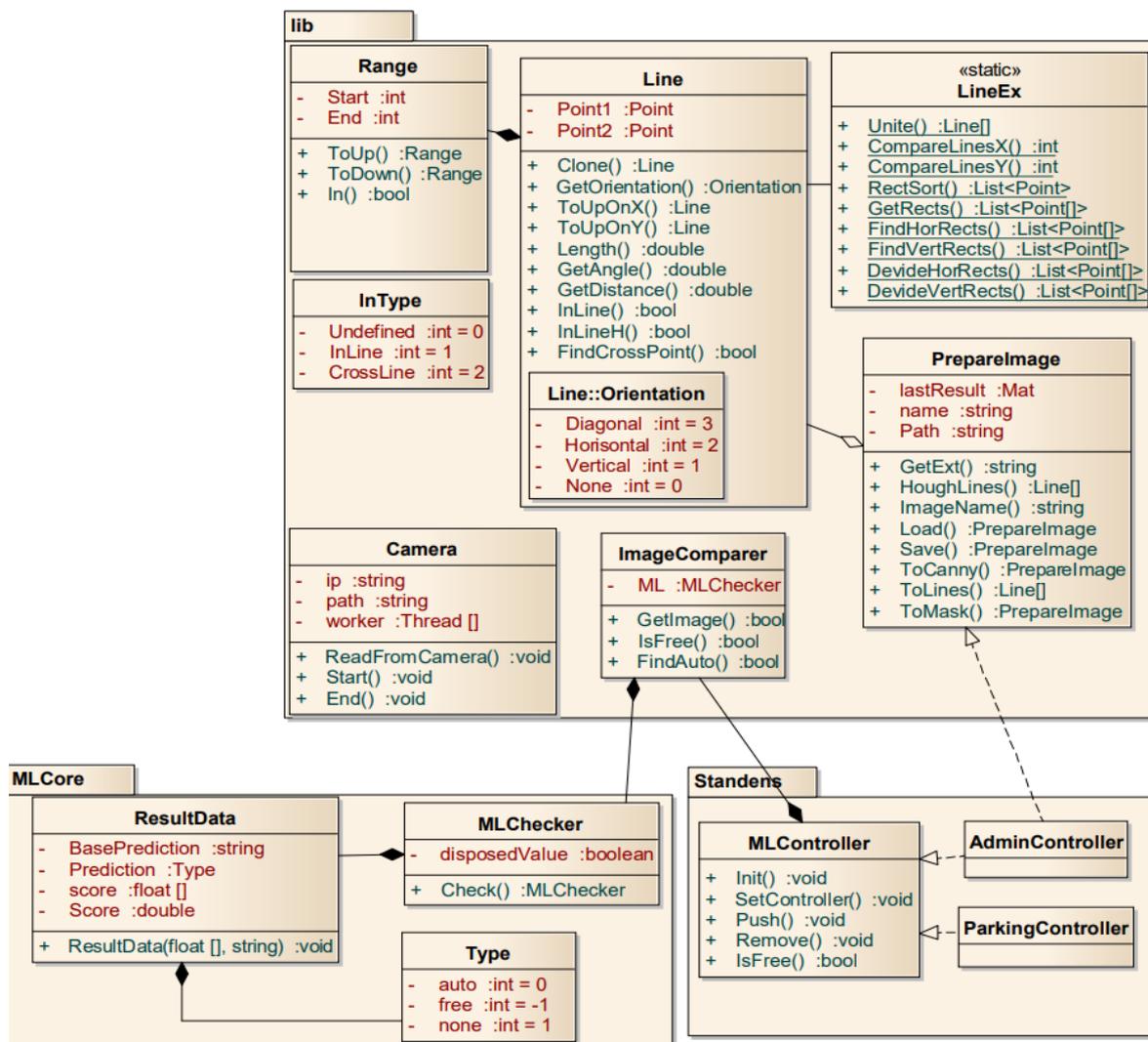


Рисунок 3. – Диаграмма классов серверной части

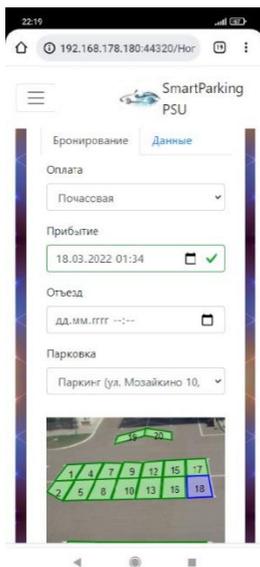


Рисунок 4 – Страница «Бронирование»

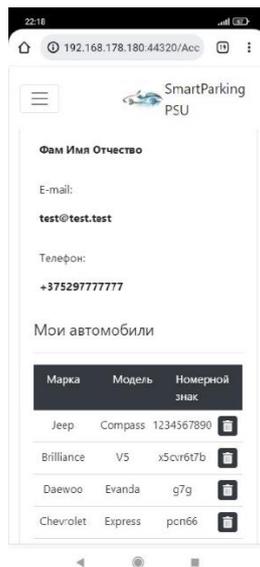


Рисунок 5 – Страница «Профиль пользователя»

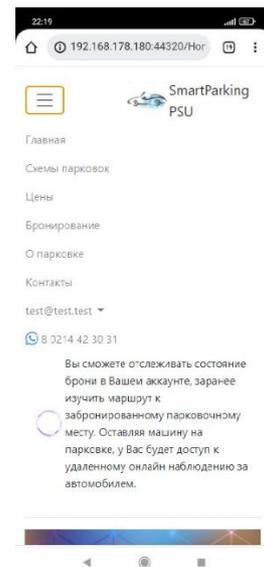


Рисунок 6 – Главная страница с открытым меню

ЛИТЕРАТУРА

1. Smart city services over a future internet platform based on internet of things and cloud: The smart parking case [Electronic resource] / J. Lanza [et al.] // Energies, 2016. – Vol. 9, no. 719. – 2016. – Mode of access: <http://dx.doi.org/10.3390/en9090719>. – Date of access: 20.12.2021.
2. Bohush, R. Extraction of Image Parking Spaces in Intelligent Video Surveillance Systems / R. Bohush, P. Yarashevich, S. Ablameyko, T. Kalganova // Machine Graphics & Vision. – 2018. – Vol. 27, № 1/4. – P. 47-62.
3. ML.NET. An open source and cross-platform machine learning framework [Электронный ресурс]. – Режим доступа: <https://dotnet.microsoft.com/en-us/apps/machinelearning-ai/ml-dotnet>. – Дата доступа: 14.01.2022.