Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра математической
кибернетики и компьютерных наук

ПОСТРОЕНИЕ СИСТЕМЫ АУТЕНТИФИКАЦИИ В СРЕДЕ ЭЛЕКТРОННЫХ ЗАМКОВ С ИСПОЛЬЗОВАНИЕМ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ НА ПЛАТФОРМЕ IOS, IBEACON, NFC, МАШИННОГО ОБУЧЕНИЯ И ДАЛЬНЕЙШЕЕ ВНЕДРЕНИЕ В КОМПАНИИ EPAM SYSTEMS

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные технологии
факультета КНиИТ
Рокитянского Артема Алексеевича

Научный руководитель
доцент, к. ф.-м. н. _____ В. М. Соловьев

Заведующий кафедрой
к. ф.-м. н. _____ С. В. Миронов

Саратов 2018

ВВЕДЕНИЕ

Целью настоящей работы является построение системы аутентификации в среде электронных замков с использованием мобильных приложений на платформе iOS, машинного обучения и дальнейшее внедрение в компании EPAM SYSTEMS.

Будучи разработчиком в компании EPAM SYSTEMS, я столкнулся со следующей проблемой - доходя от входа в офисное здание до моего рабочего места мне необходимо пройти через множество турникетов и дверей с электронными замками. Причем перемещаясь непосредственно по территории самой компании заходя на каждый этаж нужно снова открывать дверь с таким замком. Основное неудобство заключается в том, что при себе всегда нужно иметь пропуск для перемещения по территории компании и пропуск от офисного здания. Носить их всегда с собой, а тем более постоянно доставать и открывать с их помощью двери/турникеты неудобно. Эта задача и послужила темой моей работы. Каким образом можно решить данную проблему? Первое что нужно - это избавиться от пропуска через турникеты офисного здания. Второе - избавиться от пропуска к дверям самой компании. Если пропуск можно забыть, то практически всегда все берут с собой телефон. Было решено разработать приложение, которое работая в фоновом режиме само определяет когда сотрудник компании подходит к двери/турникету и они должны открываться. Также для дополнительной безопасности, а также в качестве исследования и нейронных сетей, имеющих в настоящее время широкое применение и большую популярность, было решено построить систему идентификации сотрудников через камеру, стоящую у дверей/турникетов.

Разбивая эти задачи на более мелкие были определены следующие:

— Подготовка шаблона iOS приложения;
— Изучение технологии iBeacon для детерминации замка;
— Исследование и реализация механизма NFC;
— Изучение нейронных сетей для построения системы распознавания лиц;
— Построение и обучение модели для машинного обучения;
— Интеграция выше изученных технологий в iOS приложение

Структура работы:

— Введение
— Теоритическая часть

— Экспериментальная часть
— Заключение
— Приложение

Теоритическая и экспериментальная части состоят из пяти параграфов, а именно:

— Шаблон iOS приложения
— Сервер
— iBeacon
— NFC
— Машинное обучение для распознавания лиц

1 Шаблон iOS приложения

Разрабатывая приложение всегда необходимо начинать с архитектуры. Существует множество различных архитектур, например - MVC(Model-View-Controller), MVVM(Model-View-ViewModel), MVP(Model-View-Presenter), VIPER(View-Interactor-Presenter-Entity-Router). Первые три содержат в себе такие сущности как - Model - модель данных. View - отвечает за слой представления (GUI) Controller/Presenter/ViewModel - является связующим звеном между моделью и представлением, логика определенного окна приложения находится именно тут, хотя конечно если приложении крупное, то логика отсюда выносится в другие классы, например синглтон NetworkManager, который отвечает за все сетевые запросы. [?] В данном проекте было решено использовать Apple's MVC архитектуру, так как она является стандартной и выбрана Apple для реализации приложения любого типа. [?]

Структурируя приложения получим несколько моделей данных. Одна из них это Person. У этой модели два поля: name типа String и emploeyeeId типа String. Эти данные придут с сервера после успешного логина в систему. При создании экрана настроек понадобится использовать класс UITableView, по сути это таблица с ячейками, как раз для такой ячейки создадим класс PropertiesTableViewCell. Полями этого класса будут: rssi типа UILabel, name типа UILabel, типа uuid UILabel.

Для работы с сетью необходим класс NetworkManager, котоый по своей сути является синглтоном. Он использует стороннюю библиотеку Alamofire.

Так как приложение должно отправлять уведомления пользователю, необходимо создать класс отвечающий за это. NoficationManager, он использует библиотеку от Apple – UserNofication. Уведомления отправляются, например, в тот момент когда сотрудник приходит в офис. Он получает приветственное уведомление. Так же возможны варианты отправки сообщений об ошибках и т.д.

За хранение данных в приложении отвечает класс DataManager типа NSObject. Он использует библиотеку от Apple – UserDefaluts. Хранится могут следующие данные - например данные пользователя авторизировавшегося в приложении и данные сессии.

В данном приложении были разработаны следующие экраны: 1. Main - экран, на котором расположены все элементы управлению приложением,

такие как кнопки открыть/закрыть, переход к экрану настроек, переход к экрану, на котором проводятся снимки пользователя, для дальнейшего обучения нейронной сети для распознавания лиц, и тд 2. Settings – экран настроек 3. PhotoPreparation - экран для создания снимков пользователя 4. Login - экран авторизации пользователя в системе

2 Сервер

Серверная часть использует следующие технологии:

— Gradle
— Bootstrap
— NodeJS
— Docker Compose
— Spring

Основными вляются Spring и NodeJS. Node.js - это среда с открытым исходным кодом, кросс-платформенная среда выполнения JavaScript, которая выполняет JavaScript-код на стороне сервера. Spring Framework - это платформа приложений и инверсия контейнера управления для платформы Java. Основные функции платформы могут использоваться любым Java-приложением, но есть расширения для создания веб-приложений поверх платформы Java EE (Enterprise Edition).

Для данного проекта сервер разработал другой разработчик компании, ответственный за это, так что моя задача была только поднять его на локальной машине и обеспечить соединение устройства с REST API.

3 iBeacon

Что такое iBeacon? iBeacon работает с службами определения местоположения в iOS. С iBeacon устройство iOS может предупреждать приложения, когда пользователь приближается или покидает место. В дополнение к мониторингу местоположения, приложение знает, когда пользователь близко к iBeacon, как счетчик выписки в розничном магазине. Вместо использования широты и долготы для определения местоположения iBeacon использует сигнал низкой энергии Bluetooth, который обнаруживается устройством iOS. [**?**]

В данном проекте маячок iBeacon используется следующим образом - настроив приложение на определенный маячок запускается бесконечный цикл в фоновом режиме, суть которого заключается в обнаружении маяка. Когда маячок найден нужно понять как близко он находится, чтобы избежать ложных срабатывание когда он далеко. Итак, если iBeacon расположен в определенном радиусе, то приложение отправляет запрос на сервер, передавая данные маяка как параметр, далее сервер после обработки данных открывает нужный нам турникет. Клиент получивший ответ с сервера с кодом - 200, отправляет уведомление пользователю об успешном открытии и приветствии в офисе.

Для работы с iBeacon необходимо подключтиь библиотеку CoreBluetooth, так как iBeacon по своей сути является Bluetooth-устройством.

4 NFC

Near-field communication (NFC) представляет собой набор протоколов связи, которые позволяют использовать два электронных устройства, один из которых обычно представляет собой переносное устройство, например смартфон, для установления связи путем приведения их в пределах 4 см друг от друга. Устройства NFC используются в бесконтактных платежных системах, аналогичных тем, которые используются в кредитных карточках и смарткартах электронных билетов, и позволяют мобильным платежам заменять / дополнять эти системы. В данном проекте данную технологию нужно применить следующим образом: пользователь, подходя к двери, на которой установлен NFC модуль, подносит устройство с приложением к модулю и далее срабатывает механизм двери.

Для работы с NFC необходимо подключить библиотеку CoreNFC, разработанную компанией Apple для работы с подобными типами устройств. [?]

Алгоритм открытия дверей/турникетов, с помощью NFC несколько отличается от алгоритма для iBeacon. Так как радиус действия NFC сильно меньше чем у iBeacon для открытия дверей/труникетов нужно поднести устройство к дачтику.

5 Машинное обучение для распознавания лиц

Для определения сотрудников по лицам необходимо использовать технологию распознавания лиц, с помощью машинного обучения. Первое, что нужно сделать это определить, где на изображении лицо, для этого нужно воспользоваться библиотекой OpenCV. Все, изображения или видео поток пропускается через черно-белый фильтр, так как цветовая характеристика не важна. Найдя лицо на изображении необходимо центрировать его относительно глаз, рта и т.д. Следующий шаг это обучить модель на распознавание лиц сотрудников, для этого нужно взять фотографии сделанные в приложении, далее они отправляются на сервер, где из них выделяется и выравнивается лицо. Таким образом собирается набор данных для обучения. После того как дынные собраны происходит дообучение модели с использованием библиотеки TensorFlow асинхронно на сервере. Далее пропуская через эту модель изображение сотрудника можно опередилить кто именно нахоидстя на изображении или видео потоке, что по сути в конечном итоге обрабывается как набор изображений. Сотрудник, ворятность совпадения которого выше всего и будет отображен как найденный. [?]

Таким образом система распозвания лиц должна работать в две фазы:
Фаза 1:

— Формирование базы изображений пользователей при помощи приложения, в котором сотрудники делают фотографии себя с разных ракурсов - это происходит автоматически и стандартное количество фотографий равно 10, это значение меняется внутри прложения, например для увеличения точности при фотографировании нужно увелить число фотографий;
— Обработка фотографий на сервере - выделение и выравнивание лиц с фотографий;
— Дообучение модели машинного обучения;

Фаза 2:

— На сервере запускается скрипт с захватом изображения с камеры
— В реальном времени происходит поиск лиц, с помощью библиотеки OpenCV
— Найденные лица выравниваются относительно глаз, рта и т.д.
— Лица пропускаются через ранее обученную модель и на выходе полу-

чаем метку говорящую о том какой сотрудник найден, если схожесть менее 80 процентов, то значит, что сотрудник не найден, иначе одна из ступеней аутентификации пройдена.

ЗАКЛЮЧЕНИЕ

В заключении изложены теоритические и экспериментальные результаты и сформулированы выводы. В начале работы над данным проектом была поставлена задача построения системы аутентификации через приложение для мобильных платформ на iOS и применение машинного обучения для распознавания лиц Задача была разделена на более мелкие, а именно -

— подготовка шаблона iOS приложения;
— изучение технологии iBeacon для детерминации замка;
— исследование и реализация механизма NFC;
— изучение нейронных сетей для построения системы распознавания лиц;
— построение и обучение модели для машинного обучения;
— интеграция выше изученных технологий в iOS приложение

Для обучения нейронной сети были использованы мои фотографии и фотографии сотрудников взятые с внутреннего сервиса. Таким образом перед открытием дверей/турникетов будет происходить проверка лица сотрудника, но как описывается выше в работе есть ряд проблем связанных с распознаванием лиц, а именно, например - возможна подмена лица на изображение его или видео с этим лицом, но для прохождения через турникет/дверь все равно придется быть авторизованным в приложении. Данная технология распознавания лиц, как отмечалось ранее в работе используется, как минимум, в целях исследования процесса распознавания лиц, для обеспечения большей безопасности и возможной дальнейшей интеграции с другими сервисами компании.

В реальной среде предполагается, что данные датчиков iBeacon и NFC будут приходить с сервера, кэшироваться и при каждом заходе в приложение будет осуществляться проверка на обновление данных датчиков, на случай если произойдет замена одного или нескольких, а также при переезде из офиса в офис необходимо обновить список датчиков.

Задача построения системы аутентификации в среде электронных замков с использованием мобильных приложений на платформе iOS, машинного обучения и дальнейшее внедрение в компании EPAM SYSTEMS выполнена, в данный момент проводятся тесты и подготовка данных перед непосредственным запуском данного проекта в компании.