

УДК 004.42:81'221.2

ТЕХНОЛОГИИ МАШИНОГО ОБУЧЕНИЯ И ЗАХВАТ ИЗОБРАЖЕНИЙ ДЛЯ РАСПОЗНАВАНИЯ ЯЗЫКА ЖЕСТОВ



А.А. Кунцевич
инженер-программист
Rail SkyWay Systems Ltd



Г.В. Кулик
инженер-программист
LemelLabs LLC



М.Е. Житник
инженер-программист
ITechArt Group

Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь
ЗАО «Струнные технологии», ООО "ЛемелЛабс", ООО «Техартгруп», Республика Беларусь
E-mail: kuntsevichandrey@gmail.com, kulik.dev@gmail.com, m.zhytnik@gmail.com

А.А Кунцевич

Окончил Белорусский государственный университет информатики и радиоэлектроники. Обучается в магистратуре БГУИР. Работает в Rail SkyWay Systems Ltd в должности инженера-программиста.

Г.В Кулик

Окончил Белорусский государственный университет информатики и радиоэлектроники. Обучается в магистратуре БГУИР. Работает в LemelLabs LLC в должности инженера-программиста.

М.Е Житник

Окончил Белорусский государственный университет информатики и радиоэлектроники. Обучается в магистратуре БГУИР. Работает в ITechArt Group в должности инженера-программиста.

Аннотация. В разработке данного приложения преследуется благая цель - помощь в налаживании контакта между людьми использующими язык жестов и людьми говорящими на английском языке. В работе представлены инструменты и методы разработки мобильных приложений с технологиями машинного обучения и захвата изображений, рассматривается создание модели CoreML с помощью Microsoft Custom Vision, интеграция модели в приложение, отправка запросов в модель и обработка результата в реальном времени. В результате был получен прототип приложения, распознающий отдельные буквы языка жестов с точностью более 80% в реальном времени.

Ключевые слова: ARKit, CoreML, Microsoft Custom Vision, NSLinguisticTagger, мобильное приложение, язык жестов, распознавание изображений.

Введение. Согласно лингвистическим принципам, жестовые языки обладают всеми компонентами полноценных языков, собственными морфологическими и синтаксическими структурами и постоянно развиваются.

Жестовые языки практически не зависят от звуковых: например, между амсленом и русским языком жестов немало общего, так как оба основаны на французской сурдопедагогической методике.

Однако, язык жестов для немых, не смотря на свою универсальность, достаточно сложен для понимания обычным человеком. Задумавшись именно об этом мы решили разработать приложение, которое помогло бы убрать языковой барьер между немым человеком и пользователем.

Целью данной работы является создание мобильного приложения для упрощения понимания языка жестов и общения с немыми людьми.

Достаточно простое на первый взгляд приложение имеет под собой достаточно сложную модульную структуру, поэтому было решено разделить процесс разработки на несколько основных этапов:

1. Создание обучающей и тестовой выборки;
2. Обучение модели;
3. Создание графического интерфейса мобильного приложения
4. Получение графических данных
5. Обработка графических данных
6. Форматирование данных после обработки.
7. Вывод результата в графический интерфейс.

Материалы и методы. В На WWDC'17 Apple представила новый фреймворк для работы с технологиями машинного обучения Core ML. На основе него в iOS реализованы собственные продукты Apple: Siri, Camera и QuickType. Core ML позволяет упростить интеграцию машинного обучения [1]. Модели Core ML работают локально и оптимизированы для работы на мобильных устройствах, сводя к минимуму объем используемой памяти и энергопотребление.

Создание Core ML модели происходило с помощью Microsoft Custom Vision.

Для Custom Vision потребовалось не более 60 изображений для обучения распознаванию одного объекта [2].

Обучение модели требовало качественно подготовить исходные данные. Использовались изображения жестов с разных ракурсов, при хорошем и не очень освещении и прочих факторах. Модель используется для инициализации запроса VNCoreMLRequest. Запрос использует масштабирование изображения и гарантирует, что оно вписывается в размеры входного параметра для алгоритма. Для получения запросу назначается classificationCompleteHandler.

Для захвата изображения используется CVPixelBuffer, данные для которого обеспечивает ARKit. ARKit захватывает пиксельные буферы в плоском формате YCbCr полного диапазона (также известном как YUV) в соответствии со стандартом ITU R. 601-4. В отличие от некоторых применений этого стандарта, ARKit захватывает значения цветового пространства полного диапазона [3].

Получив изображения в пиксельный буфер, происходит инициализация объекта UIImage, который в дальнейшем будем отправлен в VNImageRequestHandler.

После обработки изображения, VNImageRequestHandler отправляет результат в classificationCompleteHandler VNCoreMLRequest, параметрами которого являются VNRequest и Error. Обработав ошибки, если они возникли, можно приступить к обработке результатов классификации изображений.

VNRequest содержит массив объектов VNObservations, которые отдадут результат классификации по 26 буквам английского алфавита.

Результат классификации содержит ключ буквы и вероятность соответствия. В дальнейшем приложение сможет запоминать последовательность букв, формируя из них слова, с помощью NSLinguisticTagger [4].

На рисунке 1 изображен графический интерфейс мобильного приложения с результатами анализа жеста, соответствующего букве "W" [5].

Графический интерфейс мобильного приложения представляет собой сцену, окно вывода точности распознавания жеста, окно вывода результата обработки.



Рисунок 1. – Графический интерфейс мобильного приложения с результатами анализа жеста, соответствующего букве “W”

Заключение. На основании проделанной работы, были сформированы обучающая и тестовые выборки, обучена модель на основе вышеописанных данных, создано графическое приложение для работы с моделью. Удалось достичь результатов распознавания изображений с точностью свыше 80% на каждый объект.

Список литературы

- [1.] Документация разработчика Apple [Электронный ресурс] Электронные данные. – Режим доступа: <https://developer.apple.com/documentation/coreml>
- [2] Документация Microsoft Azure [Электронный ресурс] Электронные данные. – Режим доступа: <https://docs.microsoft.com/en-us/azure/cognitive-services/custom-vision-service/>
- [3] Документация разработчика Apple [Электронный ресурс] Электронные данные. – Режим доступа: <https://developer.apple.com/documentation/arkit/>
- [4] Документация разработчика Apple [Электронный ресурс] Электронные данные. – Режим доступа: <https://developer.apple.com/documentation/foundation/nslinguistictagger>
- [5] Алфавит глухонемых [Электронный ресурс] Электронные данные. – Режим доступа: <https://blogs.ucl.ac.uk/library-rnid/2017/07/14/a-manual-alphabet-for-the-deaf-and-dumb-circa-1870s/>

MOBILE APPLICATION FOR GESTURE LANGUAGE RECOGNITION

A.A. Kuntsevich
Software engineer
Rail SkyWay Systems Ltd

G.V. Kulik
Software engineer LemelLabs
LLC

M.E. Zhytnik
Software engineer
Itechart group

*Belarusian State University Informatics and Radioelectronics, Republic of Belarus
Rail SkyWay Systems Ltd, LemelLabs LLC, ITechArt Group, Republic of Belarus
E-mail: kuntsevichandreya@gmail.com, kulik.dev@gmail.com, m.zhytnik@gmail.com*

Abstract. The development of this application pursues a good goal - to help establish contact between people using sign language and people who speak English. This project describes and uses tools and methods for developing mobile applications with machine learning and image capture technologies, discusses the creation of a CoreML model using Microsoft Custom Vision, integrating the model into an application, sending requests to the model and processing the result in real time. As a result of our work we got a prototype application that recognizes individual sign language letters with an accuracy of more than 80% in real time.

Key words: ARKit, CoreML, Microsoft Custom Vision, NSLinguisticTagger, mobile application, gesture language, image recognition.