Разработка механизма автоматического перевода текста на фотографии для мобильных устройств

**Введение**

С развитием технологий человечество вошло в новую эпоху - эру мобильных устройств. Их количество исчисляется миллиардами, люди проводят часы жизни за ними и уже вряд ли могут представить себя без них. В данной ситуации крайне важно использовать мобильные устройства как помощников в быту или в решении технических задач. Сейчас вычислительные мощности Вашего смартфона или планшета уже превосходят мощности стационарных компьютеров 2-3 летней давности. Таким образом, с их помощью можно решать и ресурсоемкие задачи, которые раньше были под силу только мощным системам. Примером такой задачи является распознавание текста на фотографии. Но только распознать текст мало - его можно и нужно использовать для дальнейшей обработки и получения информации. Но, если текст на незнакомом для Вас языке, то он нуждается в переводе. Такая ситуация часто встречается в путешествиях или на конференциях, которых проводится все больше благодаря глобализации. Таким образом и родилась идея данной работы - разработка механизма автоматического перевода текста на фотографии для мобильных устройств. **Целью** данной работы является развитие прикладных навыков в разработке мобильного приложения для платформы iOS и применение знаний в области машинного обучения для использования модели распознавания непосредственно на мобильном устройстве. Для достижения поставленной цели необходимо было решить следующие **задачи**:

1. Разработка UI приложения с учетом интуитивности и получения пользователем положительного UX
2. Разработка архитектуры приложения и взаимодействия его компонент
3. Исследование и применение фреймворков и библиотек сторонних разработчиков
4. Выбор модели для распознавания и ее обучение
5. Портирование полученной модели в формат, пригодный для использования на мобильном устройстве
6. Тестирование полученного прототипа

**Терминология**

UI (user interface) - интерфейс программы, непосредственно с которым взаимодействует пользователь.

UX (user experience) - опыт пользователя, полученный вследствие взаимодействия с программой.

**Главы**

1. Разработка UI, макета приложения
2. Разработка архитектуры (Core, StyleKit, Routing, Data…)
3. Разработка приложения
4. Сторонние библиотеки и фреймворки
5. Обучение и использование ML модели
6. Портирование нейронной сети в CoreML model.
7. C++ -> Obj-C -> Swift bridging
8. Тестирование
9. Результаты и выводы

**1) Дизайн и разработка UI**

Основная задача при разработке UI приложения – создание макета и дальнейший перенос его элементов в само приложения. Сейчас на рынке достаточно хороших инструментов для дизайна и создания макета (Zeplin, Sketch, Figma…). В данной работе использовался инструмент Figma (<https://www.figma.com/>) по нескольким причинам:

* Поддержка всех платформ (Windows, MacOS, браузерный интерфейс)
* Возможность прототипирования и создания интерактивных переходов между экранами
* Хранение проекта в облаке
* Возможность создания публичной ссылки для просмотра проекта
* Интуитивность и наличие всех необходимых инструментов

Начальным этапом дизайна является выбор цветовой палитры приложения. В качестве цветовой палитры было решено использовать спокойной сочетание цветов, на фоне которых довольно контрастно бы смотрелось фотография, сделанная пользователем. Основные три цвета палитры – черный, серо-синий и бирюзовый.

В качестве основного шрифта приложения был выбран шрифт Rationale за свою футуристичность. Одна из ключевых компонент дизайна – логотип приложения. В нем крайне важно лаконично и понятно отразить суть приложения, чтобы заинтересовать пользователя уже на этапе просмотра чартов в AppStore или Google Play. Итоговая идея – «облако» по образу подложки сообщений в мессенджерах, внутри которого схематично изображены реплики с языками, на котором они написаны. Таким образом показаны главные цели приложения – передача и получение информации и возможность перевода, интернациональность.



Финальный вариант логотипа

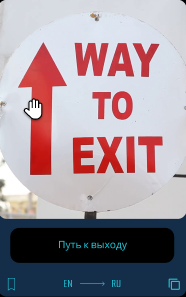
Приветственный экран приложения было решено сделать стандартным образом – логотип посередине, название приложения прибито к нижней части экрана. Для дальнейшей разработки дизайна нужно было понять и сформулировать функциональность приложения, в результате чего был составлен список базовой функциональности:

* Возможность создания фотографии, с которой будет осуществляться перевод текста
* Отображение результата перевода, языка текста на фотографии и языка, на который произошел перевод
* Возможность копирования результата перевода в буфер обмена
* Хранение переводов с возможностью удаления
* Настройки и их переключение
* Информация о приложении
* Обратная связь, возможность написать автору

На основе этого списка был составлен список основных экранов приложения:

* Приветственный экран
* Основной экран
* Экран настроек
* Экран конкретной настройки с выбором возможных значений
* Экран сохраненных переводов
* Экран «О приложении» с кнопкой «Написать разработчику»

Для основного экрана и для экрана сохраненных переводов было решено создать похожий интерфейс – основную часть занимает карточка перевода (колода карточек переводов в случае экрана с сохраненными переводами). В карточке перевода необходимо отобразить основную информацию – непосредственно фотографию, итоговый результат, языки и кнопки для реализации функционала (копирование в буфер, сохранение или удаление из списка переводов). Большую часть данной карточки, что логично, занимает фотография текста, которая в случае основного экрана служит отображением изображения с камеры. Под фотографией расположено поле, содержащее текст итогового перевода, а под ним кнопка сохранения/удаления из списка сохраненных переводов, целевой и конечный языки перевода, кнопка копирования итога в буфер. При оформлении карточки было решено использовать скругленные края и все три основные цвета палитры.



Итоговый макеты карточки перевода

На основном экране должна быть возможность перехода к экрану настроек и экрану сохраненных переводов. Так же должны быть возможность снять фотографию, чтобы инициировать процесс перевода. Для этого в Tab Bar основного экрана добавлены три кнопки, выполняющие соответствующие задачи.



Tab bar основного экрана

Каждый экран, кроме основного и экрана «О Приложении», подписан заголовком. Для навигации среди экранов предусмотрена кнопка «Назад» в левой верхней части в виде стрелки влево.

Так как карточка перевода занимает почти весь экран, то на экране сохраненных переводов было решено сделать навигацию между переводами с помощью смахиваний перевода наверху стека в сторону. Получилось своеобразная колода карточек с переводами. В правом верхнем углу предусмотрена кнопка для отмены последнего действия, а по истечению карточек в колоде появляется надпись о том, что переводы закончились. Если же список изначально пуст, надписать посередине экрана оповещает об этом.

Экраны настроек традиционно сделаны в единообразном стиле – это таблица с ячейками. Ячейки в приложении было решено сделать двух типов, а именно ячейка с дополнительной информацией и без нее. Для ячейки, в которой отображается только основная информация, текст центрируется, аналогично остальному интерфейсу приложения использованы скругленные края и общая палитра.



Пример ячейки с основной информацией

В ячейке, содержащей дополнительную информацию (помимо основной) центрирование не происходит. Основная информация располагается по левому краю, дополнительная – по правому краю.



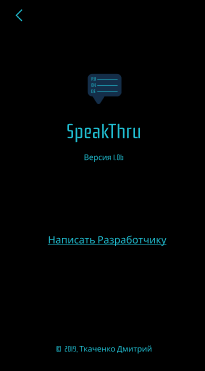
Пример ячейки с основной и дополнительной информацией

Остальные экраны конкретных настроек представляют собой таблицу ячеек с основной информацией. Для перехода с экрана настроек к экрану «О приложении» отдельной ячейки не предусмотрено. За переход на данный экран отвечает надпись внизу экрана настроек.



Переход к экрану «О приложении»

Сам экран содержит основную информацию – логотип, название, версию, копирайт и возможность написать разработчику на почту. Для отличия кнопки обратной связи от остальных надписей было решено выделить ее подчеркиванием:



Экран «О приложении»

В итоге, разработанный интерфейс приложения решает поставленные перед ним задачи:

* Отсутствие перегруженности элементами
* Реализация базовой функциональности
* Возможность изменения настроек
* Спокойная палитра и шрифт, улучшающие UX
* Удобство просмотра перевода в силу расположения почти по всей поверхности экрана карточки с переводом
* Возможность обратной связи

**2) Разработка архитектуры**