

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ ДОНЕЦКОЙ НАРОДНОЙ РЕСПУБЛИКИ  
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ДИПЛОМНЫЙ ПРОЕКТ

на тему:

**«Разработка системы распознавания печатного текста»**

АВТОР ДИПЛОМНОГО ПРОЕКТА

ст. гр. ПОС-13 Шумский Александр Александрович

РУКОВОДИТЕЛЬ ДИПЛОМНОГО ПРОЕКТА

ст. преп. Бычкова Елена Викторовна

Донецк – 2017

## ПОСТАНОВКА ЗАДАЧИ

Целью дипломного проекта является разработка системы распознавания печатного текста «OCR».

В данной работе поставлена задача создания мобильной системы распознавания печатного текста с открытым исходным кодом.

Программная реализация системы распознавания печатного текста «OCR» разделена на основные модули:

- модуль интерфейса;
- модуль распознавания текста;
- модуль работы с файлами;
- модуль фильтров.

На текущий момент большинство подобных приложений и сервисов платное, с закрытым исходным кодом и имеет ограничения по объему обрабатываемых данных. В связи с этим создание бесплатной системы для распознавания печатного текста с открытым исходным кодом является крайне востребованной задачей.

## АНАЛИЗ ПРОГРАММ-АНАЛОГОВ

В качестве конкурентов были выделены пять основных приложений в Google play: FineScanner от компании ABBYY, Image to Text (OCR Scanner) от компании HMA Labs, Text Scanner [OCR] от компании Peace, CamScanner Phone PDF Creator от компании INTSIG и The Simple Cam Scanner (OCR) от компании Generic.

В результате анализа рынка систем по распознаванию печатного текста была выявлена нехватка бесплатных и легковесных приложений с поддержкой нескольких языков.

Из этого следует, что разрабатываемый продукт должен соответствовать следующим критериям:

- качественно распознавать текст;
- поддерживать несколько языков;
- занимать мало места в памяти;
- иметь простой интерфейс;
- поддерживать один или несколько фильтров;
- сохранять результирующий текст в файл;
- быть бесплатным (монетизация за счет рекламы).

Подобный продукт способен составить конкуренцию существующим приложениям на рынке мобильного программного обеспечения по распознаванию печатного текста.

## ЭТАПЫ РАБОТЫ ПРОГРАММЫ



Для максимально эффективного распознавания текста необходимо выполнить действия по предобработке и постобработке.

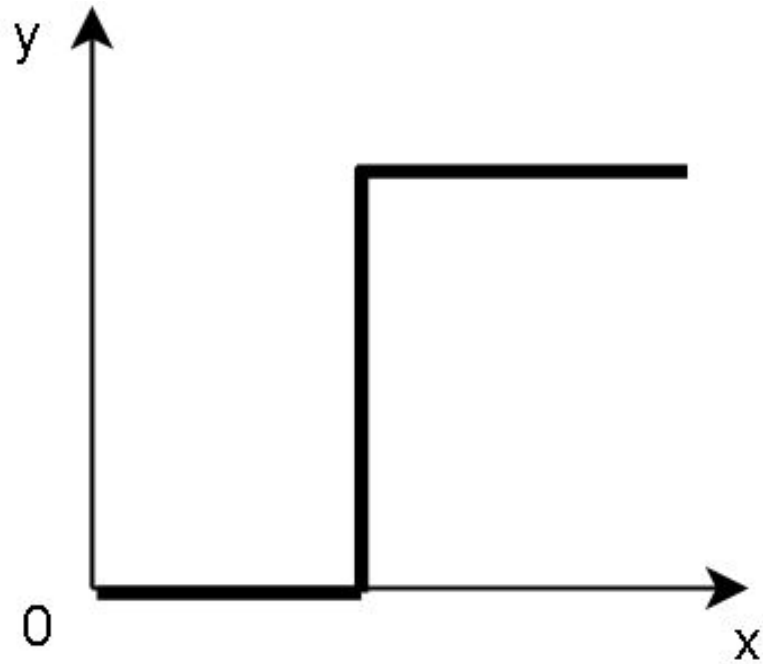
Входными данными являются фотографии исходного текста. Для повышения точности распознавания необходимо провести обработку изображения. Обычно применяется монохромный фильтр или обычный чёрно-белый. Далее обработанное изображение поступает на модуль сегментации, где выделяются сгустки пикселей в потенциальные буквы, а сами сгустки объединяются в потенциальные слова. Данные буквы классифицируются с помощью нейронной сети, а затем идет поиск и сравнение итоговых слов со словарем.

## МОНОХРОМНЫЙ ФИЛЬТР

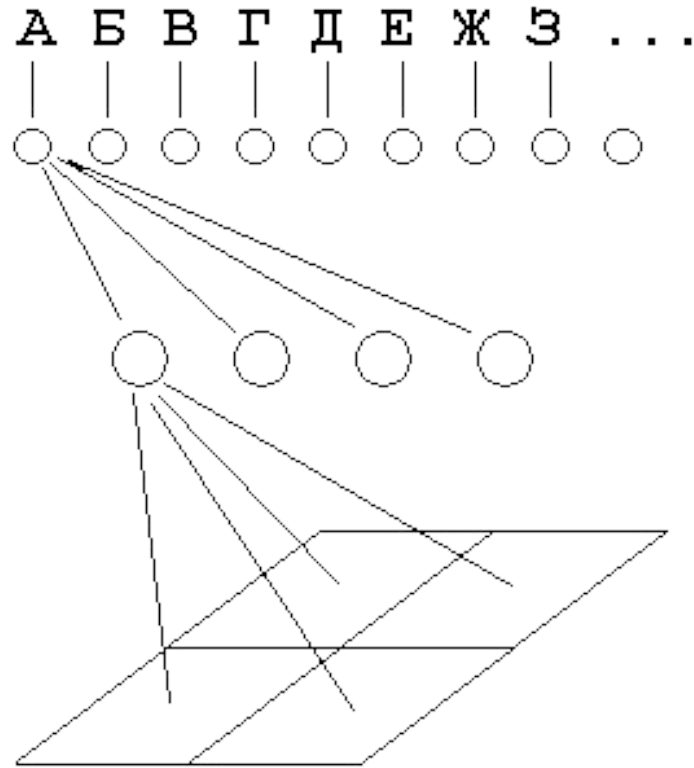
Монохромный фильтр преобразует исходное цветное изображение в новое, содержащее свет одного цвета (длины волны), воспринимаемый как один оттенок.

Монохромный фильтр необходимо реализовать с применением метода препарирования, в котором граница перехода высчитывается путем нахождения среднего значения яркости всех пикселей изображения. Такой подход позволяет при сегментации выделить буквы даже при плохом, но равномерном освещении даже на цветной бумаге.

В качестве точки перехода выбрано среднее значение яркости для всего изображения.

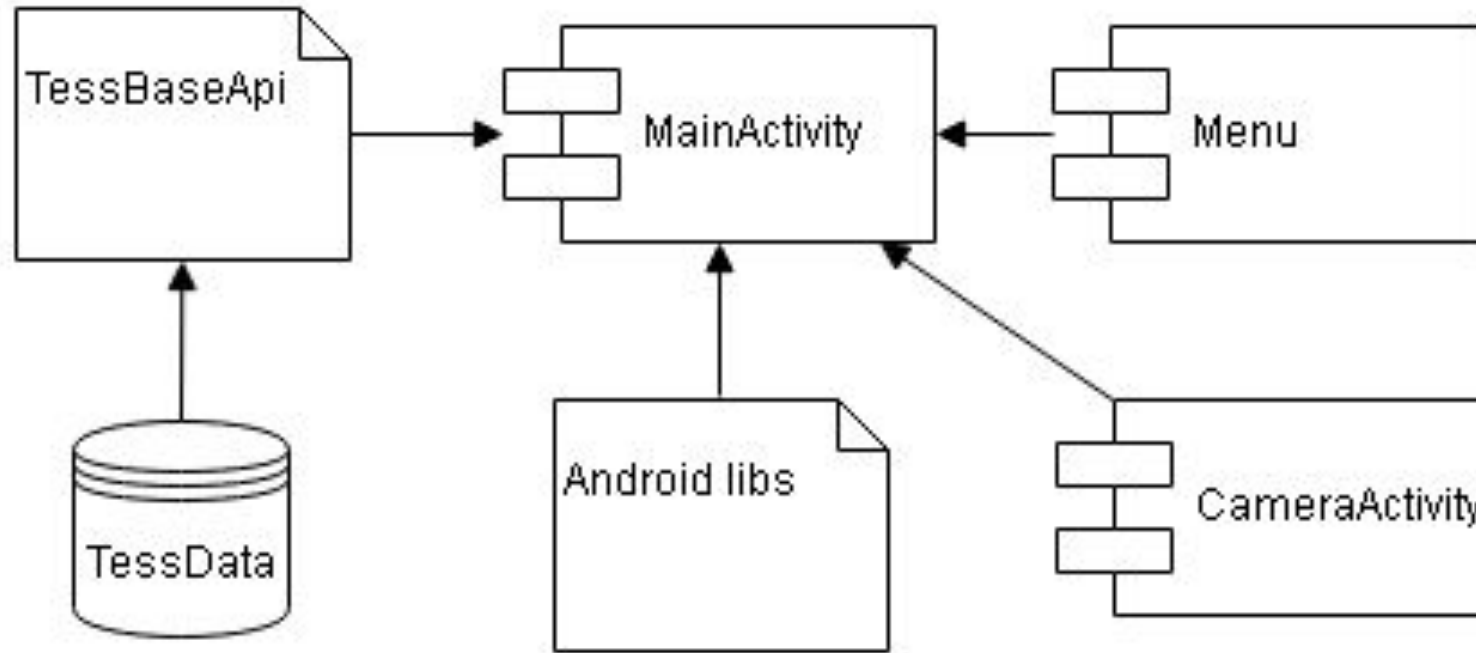


## СВЯЗИ И УЗЛЫ НЕЙРОННОЙ СЕТИ



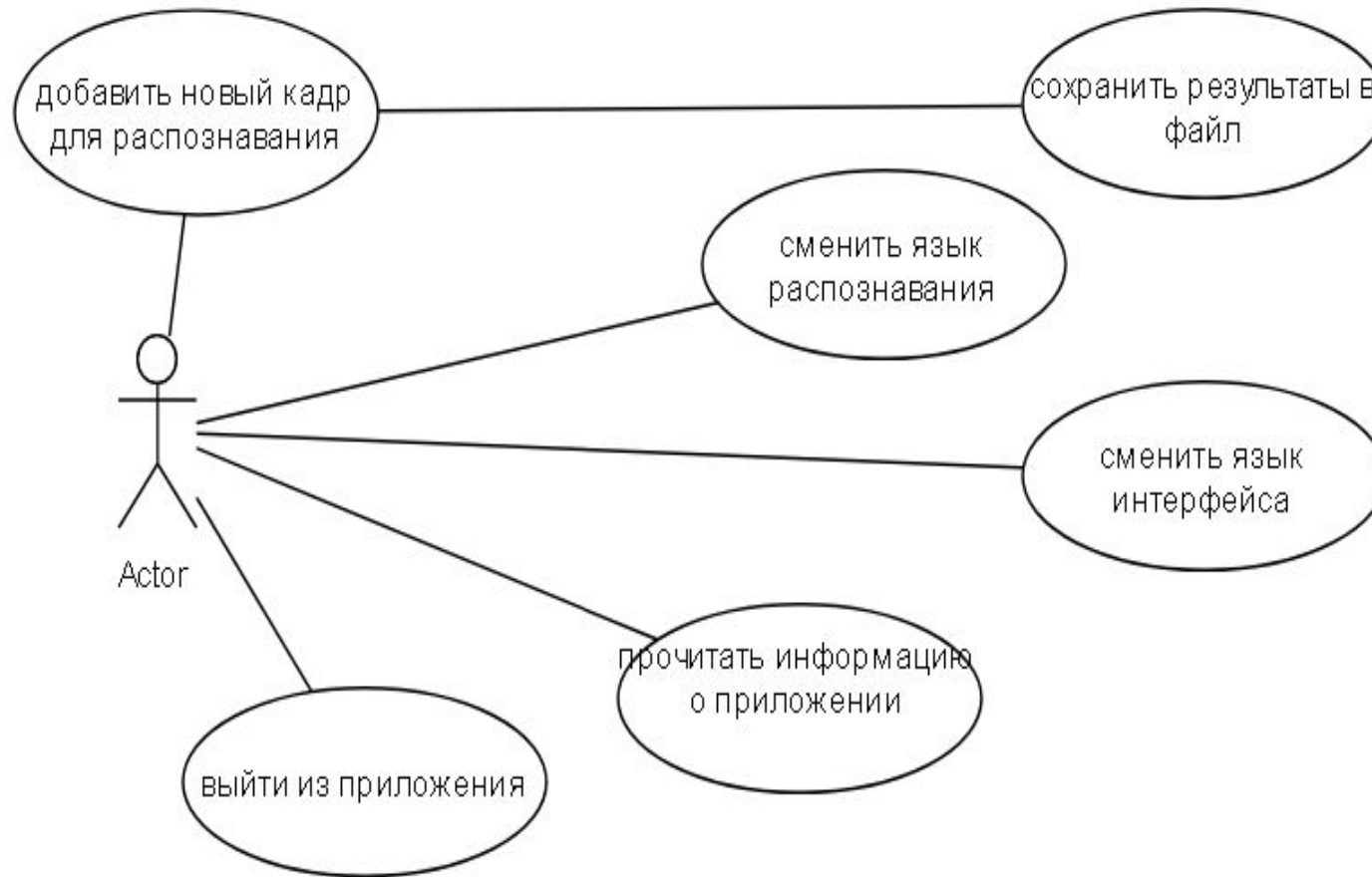
Число выходных параметров нейронной сети равняется числу распознаваемых символов. Результатом распознавания является символ, которому соответствует наибольшее из значений выходного вектора нейронной сети. Повышение надежности таких алгоритмов связано, как правило, либо с поиском более информативных входных признаков, либо с усложнением структуры нейронной сети.

## ДИАГРАММА МОДУЛЕЙ



Данная диаграмма содержит в себе 6 условных модулей. Три из них являются программными компонентами: MainActivity, Menu, CameraActivity. Одна – пакетом библиотек Android. Одна – внешней библиотекой TessBaseApi. Также имеется одна база данных – TessData.

## ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ



Диаграммы вариантов использования или диаграммы прецедентов относятся к той группе диаграмм, которые представляют динамические или поведенческие аспекты системы. Это отличное средство для достижения взаимопонимания между разработчиками, и конечными пользователями продукта.



## ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТЕЙ

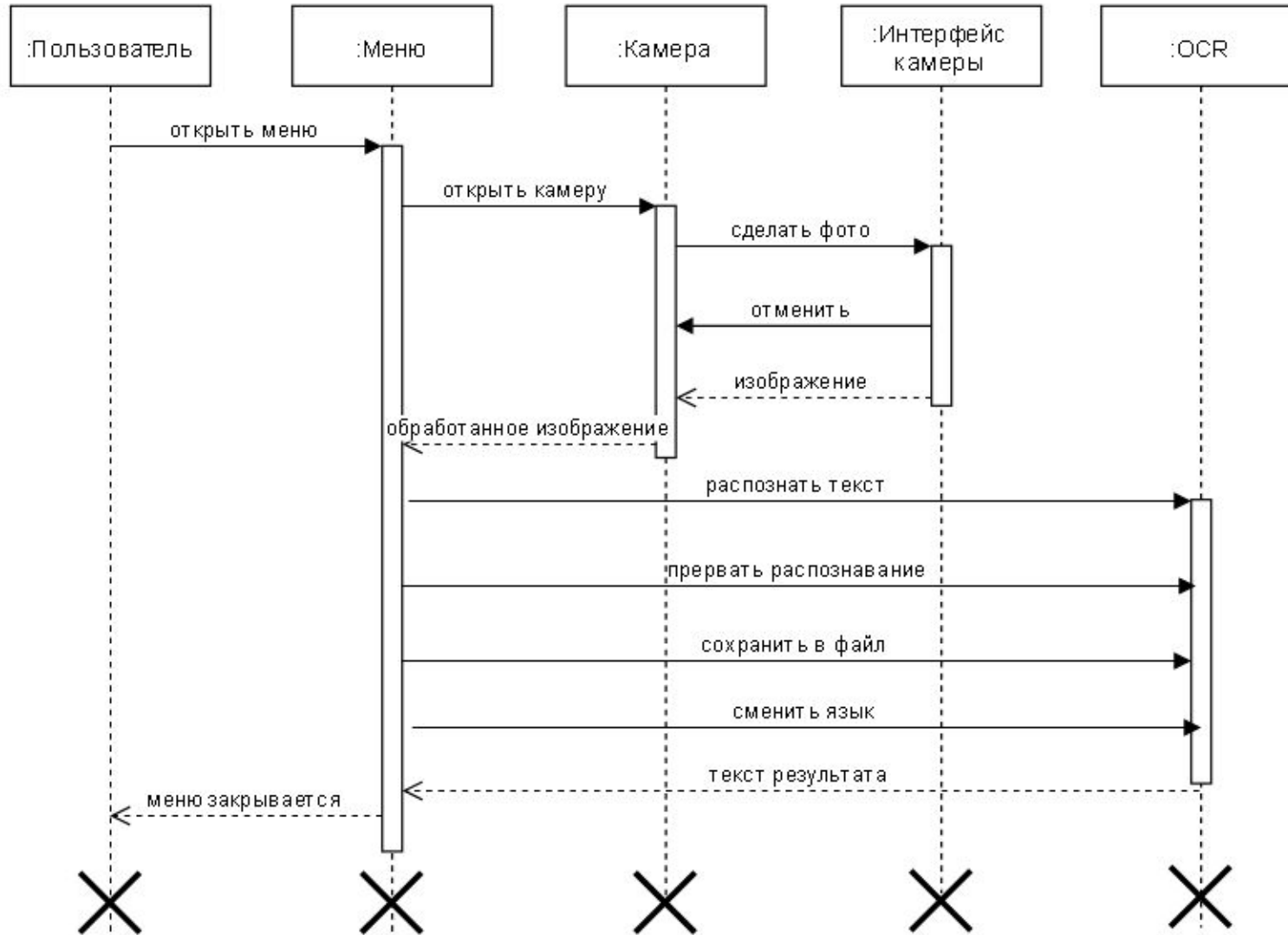


Диаграмма последовательностей относится к диаграммам взаимодействия UML, описывающим поведенческие аспекты системы, но рассматривает взаимодействие объектов во времени.

## ДИАГРАММА АКТИВНОСТИ

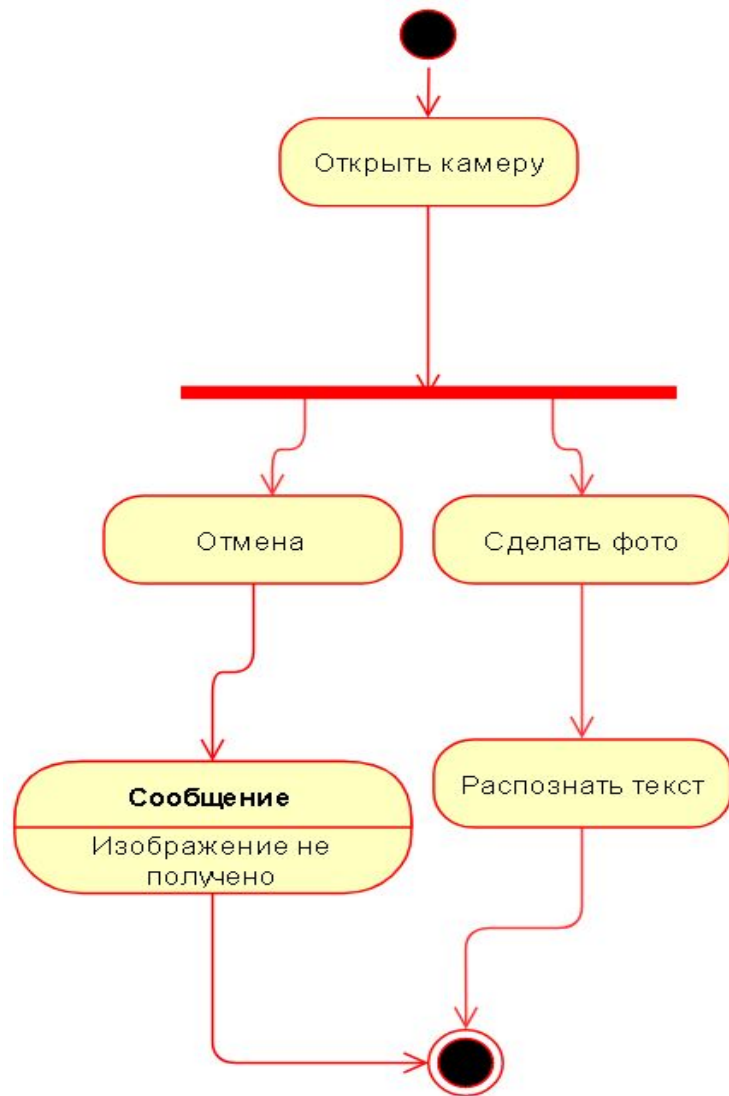


Диаграмма активности – диаграмма, на которой показаны действия, состояния которых необходимо согласовывать.

Под активностью понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов – вложенных видов деятельности и отдельных действий, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого.

## ТЕСТИРОВАНИЕ И ОТЛАДКА

С целью выявления ошибок было проведено тестирование методом черного ящика.

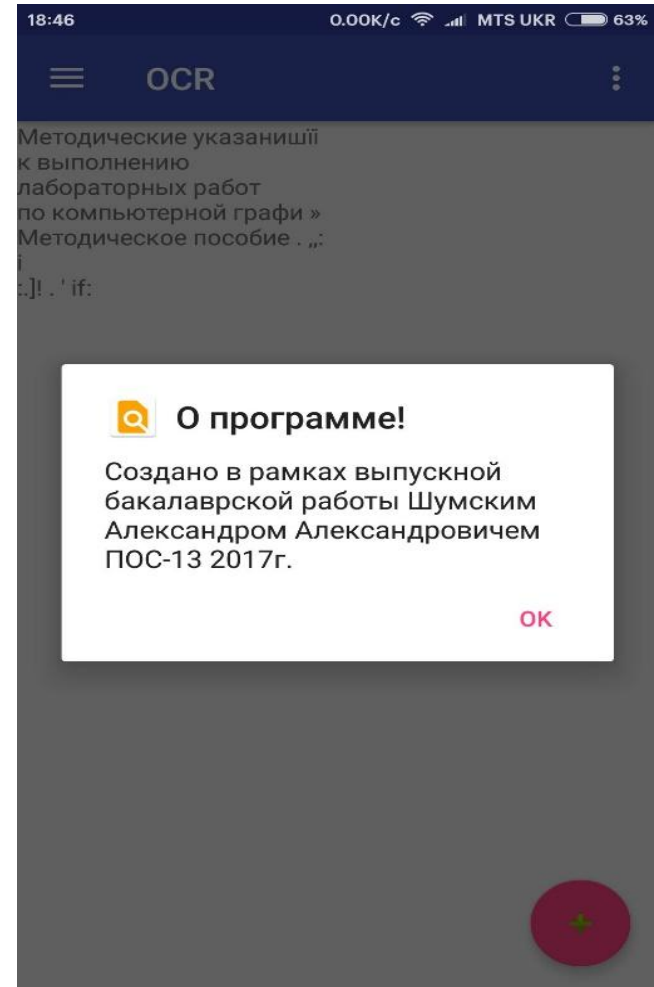
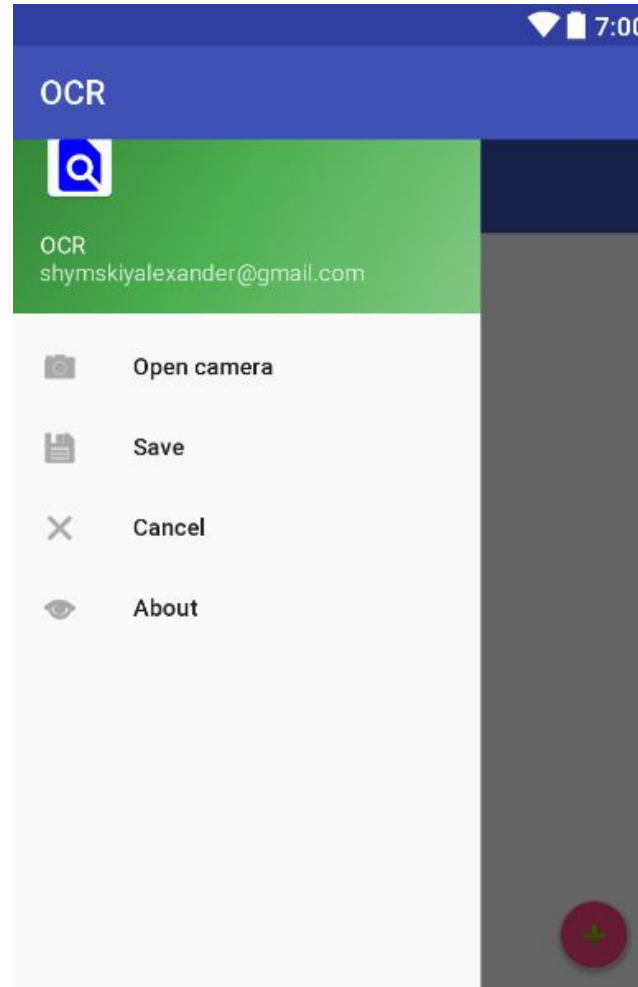
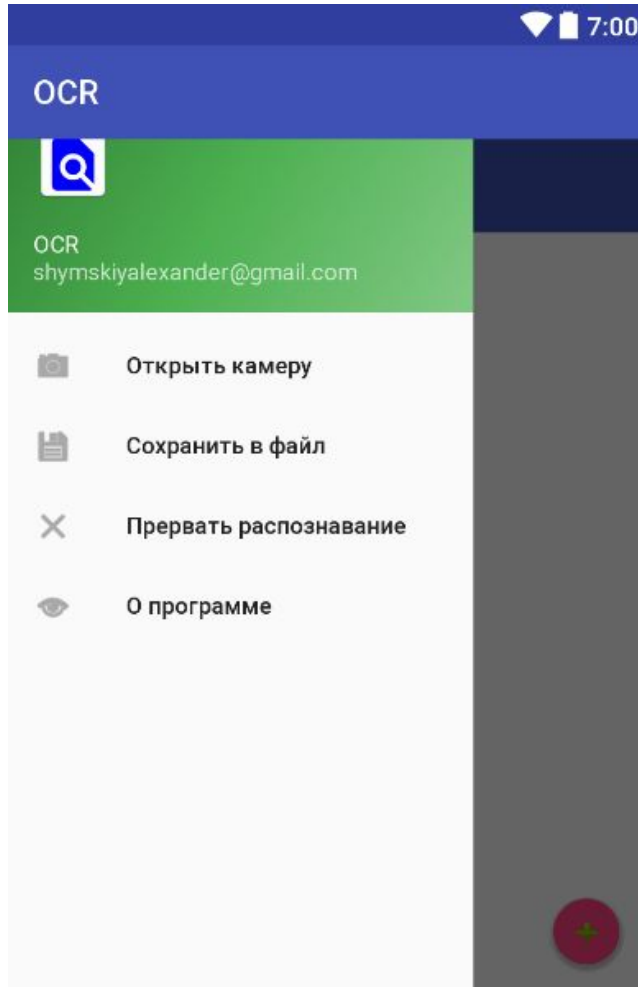
Приложение было протестировано на следующем наборе версий операционной системы Android в эмуляторе:

- Android 5.0 (API 21);
- Android 5.1 (API 22);
- Android 6.0 (API 23);
- Android 7.1.1 (API 25).

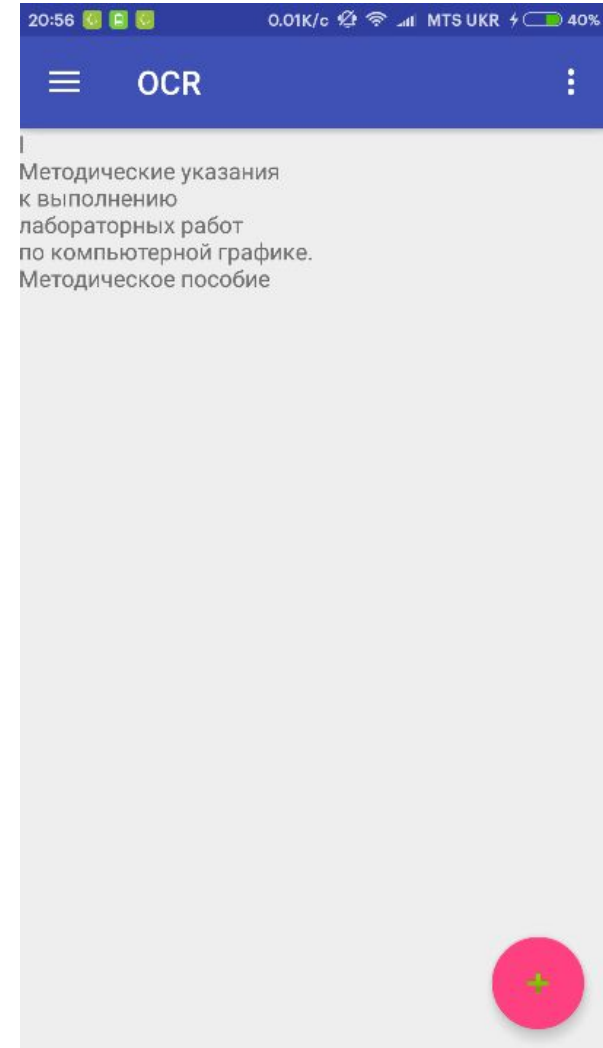
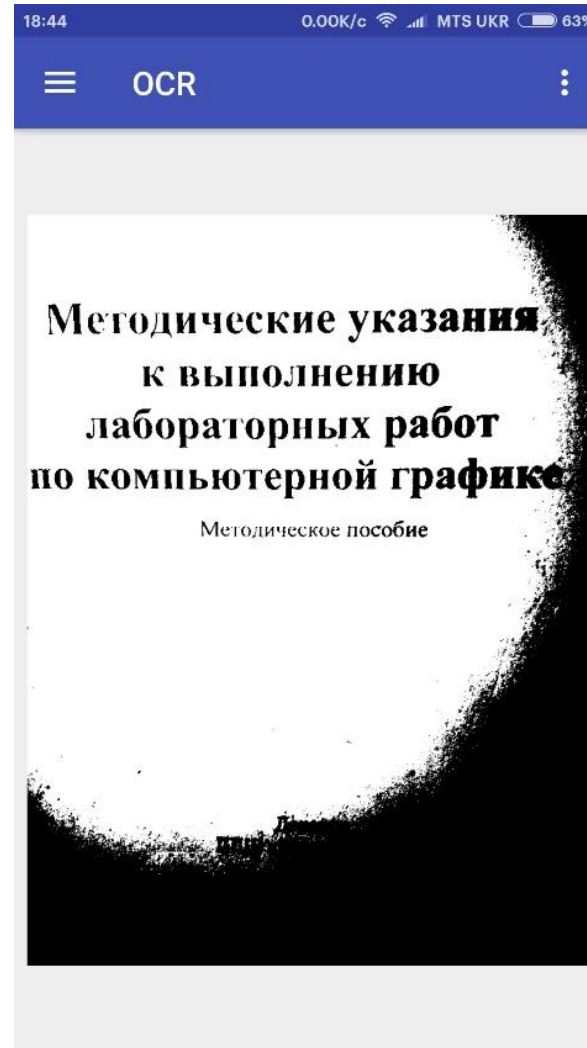
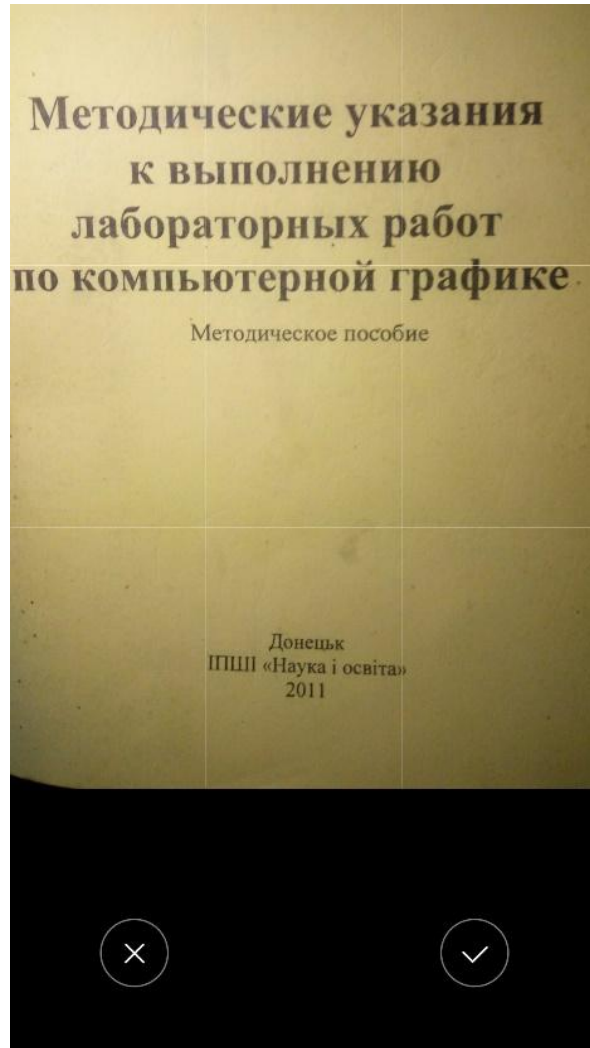
Устройства с API ниже 21 не поддерживаются из-за технических особенностей реализации, а также версии 21 и выше занимают более 67% среди всех версий.

Точность и время распознавания напрямую зависят от качества освещения на фото. Эксперименты показали, что для достижения наилучших результатов необходимо мощное равномерное освещение. Порог работы фильтра был вручную понижен для достижения наилучших результатов.

## ЭКРАННЫЕ ФОРМЫ ГЛАВНОГО МЕНЮ



## ЭКРАННЫЕ ФОРМЫ ПРОЦЕССА РАСПОЗНАВАНИЯ



## ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломного проекта были выявлены основные аспекты и задачи, решение которых требуется для полноценной работы программы. В результате анализа рынка было выявлено, что создание мобильной системы распознавания текста является крайне востребованной задачей.

Были изучены основные принципы фильтрации изображений; рассмотрены различные методы распознавания образов; проведено обоснование выбора языка программирования применительно к поставленной задаче.

Изучена архитектура и особенности разработки приложений под Android, ее библиотеки, ресурсы, принципы работы. Проведен анализ существующих библиотек распознавания текста.

Была создана UML-диаграмма вариантов использования, диаграмма модулей, диаграмма активности и диаграмма последовательностей. На их основе было спроектировано и разработано мобильное приложение.

В результате работы над дипломным проектом было разработано мобильное приложение по распознаванию печатного текста. В дальнейшем предполагается размещение приложения в Google Play и усовершенствование программных модулей.