# Отчет по проектной работе по **курсу** “Разработка мобильных приложений»

# Тема: «Распознавание рукописного текста на изображении»

# Работу выполнили студенты ИУ9-71Б:

# Дьячек П.В.

# Ионов Т.Р.

# Захарова В.Д.

# **Введение**

# Целью данной работы является создание мобильного приложения для распознавания текста на изображении, полученного с мобильного устройства. Было принято решение разрабатывать клиент-серверное приложение, так как сам процесс распознавания на телефоне сделать достаточно трудно, в том числе из-за ограничений, связанных с вычислительной мощностью.

# Задачи:

# 1. Подготовить сервер API для модели распознавания рукописного текста.

# 2. Разработать мобильное приложение, осуществляющее выбор и отправку изображения на сервер, с последующим отображением результатов распознавания.

# **Схема приложения**

# Блок-схема процессов в приложении показана на рисунке 1. Клиент передает изображение на сервер в виде последовательности байт, которые сервер конвертирует в необходимый формат для модели. После получения информации о положении текста и его содержании, данные отправляются в виде

# Json-файла обратно на клиент, где пользователю показывают результат в виде текста, а также на исходном изображении отрисовываются прямоугольники с текстом, которые были обнаружены моделью.

# Рисунок 1 — Схема приложения

# **Процесс разработки**

# В силу того, что приложение клиент-серверное, достаточно просто вести параллельную разработку, ведь можно тестировать отдельно клиент или сервер, создавая тестовые запросы/ответы. Список задач, распределенный по участникам команды в виде диаграммы Ганта приведен на рисунке 2. Соответственно Ионов Т. отвечал за серверную часть и подготовку модели, Захарова В. за создание мобильного приложения и Дьячек П. За организацию и работу API.

# Рисунок 2 — диаграмма Ганта

# Выбор изображения на клиенте осуществляется с помощью встроенного в Android API, которое позволяет выбрать изображение из памяти устройства, или же сделать снимок.

# В ходе разработки возник вопрос о том, как хранить изображение, отправленное пользователем, пока сервер не дал ответ. Можно было отрисовывать прямоугольники с текстом прямо на сервере, возвращая отредактированную картинку на клиент вместе с текстом, однако это более затратно в плане траффика. Было принято решение создать простую базу данных из одной таблицы, которая являлась абстракцией записи, имеющей заголовок, распознанный текст, путь к самому изображению, а также список точек для прямоугольников на изображении.

# Для разработки сервера был выбран язык Python, так как это значительно упрощает развертку модели глубокого обучения. В качестве фреймворков для создания API были выбраны Flask и Gunicorn. С их помощью осуществляется получения файла по http в виде байткода, а также последующая отправка результатов на клиент в виде json-файла.

# 

# **Результат**

# В результате приложение оформлено в виде списка заметок, к каждой из которых прикреплено изображение с распознанным текстом. Этот текст можно редактировать, исправляя в таком случае возможные ошибки модели. Список записей, а также кнопки для их добавления или удаления показаны на рисунке 3.

# 

# Рисунок 3 — список заметок

# На рисунке 4 демонстрируется интерфейс заметки: изображение с размеченными прямоугольниками с текстом, редактируемое текстовое поле с результатом, кнопки для загрузки изображения, отправки на сервер и сохранения.

# 

# Рисунок 4 — Интерфейс заметки

# **Выводы**

# В ходе проектной работы было разработано мобильное приложения для распознавания рукописного текста на изображении. Все поставленные задачи успешно выполнены в срок. Были приобретены навыки планирования разработки, командной работы. Получен опыт создания мобильных приложений, а также API-серверов.