

**И. И. КИРЕЕВ, И. В. КУЛДАРЕВ, З. Р. САФИУЛЛИН,  
П. В. ПЛОТСКИХ, А. В. МЕЛЬНИКОВА**

*Тюменский государственный университет, г. Тюмень*  
**УДК 004.93**

## **РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ПОМОЩИ В ИЗУЧЕНИИ РУССКОГО ЖЕСТОВОГО ЯЗЫКА**

***Аннотация.** В работе представлен процесс разработки мобильного приложения с базовыми функциями, которое помогает в обучении русского жестового языка, и с функцией распознавания жеста пользователя с помощью камеры и методом классификации случайные леса.*

***Ключевые слова:** мобильное приложение, классификатор, случайные леса, русский жестовый язык, клиент-серверная архитектура, прототип.*

**Введение.** В современном обществе все больше людей стало заниматься самообразованием, хотят приобрести или дополнить навыки без чьей-либо помощи. Главное преимущество самостоятельного обучения навыкам с помощью гаджетов и/или поиска в доступной интернет-среде заключается в том, что ограниченные знания не ограничиваются одним человеком. Однако при таком преимуществе возникает проблема контроля правильности полученных знаний. Информация, произношение или жесты обычно контролируются живым человеком, а именно учителем/преподавателем, но на просторах открытых источников возможна фальсификация информации.

Эта проблема может быть решена путем создания приложений для контроля вышеуказанных сфер изучений. Рассматриваемая нами проблема поднимается в изучении русского жестового языка и контроля правильного показа жеста.

Целью данной работы является разработка мобильного приложения с базовыми функциями для изучения жестового языка, добавить уровни заданий и реализовать один из уровней как задание: «Покажите жест на камеру», в ответ которого показывается сообщение «Правильно/Неправильно» с помощью методов классификации.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Исследовать и найти методы классификации руки.
2. Найти и/или создать базу данных фотографий русского жестового алфавита.
3. Написать программу распознавания жеста с выводом соответствующего текста.
4. Исследовать приложения для обучения навыкам на функционал, вследствие чего выстроить дизайн и функционал для удобства пользователя.
5. Изучить способы передачи изображения, сделанное пользователем, в программу распознавания жеста, а после получение ответа.
6. Изучить способы обнаружения руки на видеопотоке.

### **Материалы и методы.**

#### *Постановка задачи классификации.*

Постановка задачи звучит следующим образом: дано  $x$  — изображение (фото жеста). Пусть  $X$  — множество описаний изображений жестов,  $Y$  — конечное множество имен классов. Существует неизвестная целевая зависимость — отображение  $y: X \rightarrow Y$ , значения которой известны только на объектах конечной обучающей выборке  $X^m = (x_1, y_1), \dots, (x_m, y_m)$ . Требуется: определить оптимальный алгоритм  $a: X \rightarrow Y$  способный классифицировать произвольное изображение  $x \in X$ .

#### *Анализ жестов и выбор метода классификации*

Первым этапом реализации проекта являлся поиск источников информации, в том числе и статьи на тему анализа жестов [6].

По итогу изучения открытых источников было установлено следующее: для разработки приложения была выбрана клиент-серверная архитектура (рис. 1). В качестве помощи реализации клиентской части использована библиотека OkHttp [3], серверной — фреймворк Flask [4]. Клиентское приложение делает запрос на сервер, в котором передается изображение. Далее на сервере в два этапа происходит анализ изображения. На первом этапе, в глубокой сверточной нейронной сети (на основе архитектуры Single Shot Multibox Detector) [1], находится кисть руки и ее ключевые точки (рис. 2). На втором этапе координаты  $x$ ,  $y$ ,  $z$  ключевых точек передаются в классификатор, который распознает жест и возвращает значение в клиентское приложение.

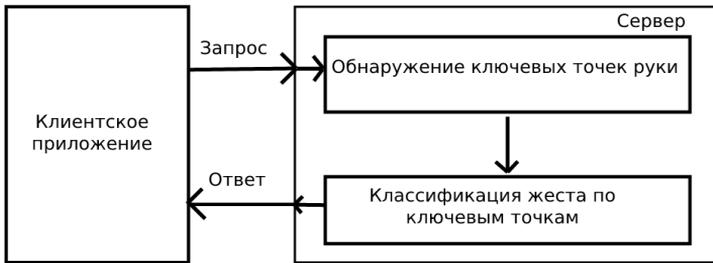


Рис. 1. Клиент-серверная архитектура приложения

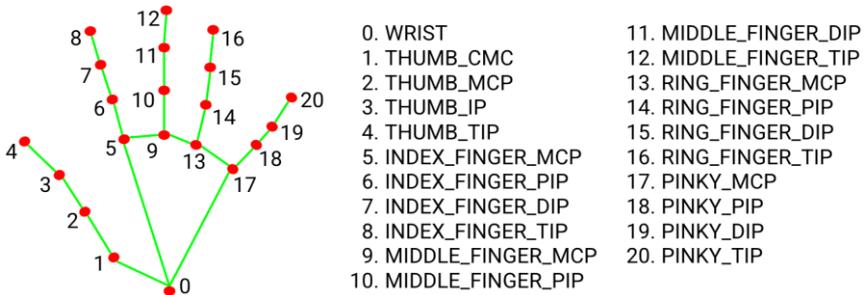


Рис. 2. Двадцать одна ключевая точка руки

Для обучения классификатора был создан датасет русского жестового языка объемом 2571 элемент. Структура датасета представлена на рис. 3.

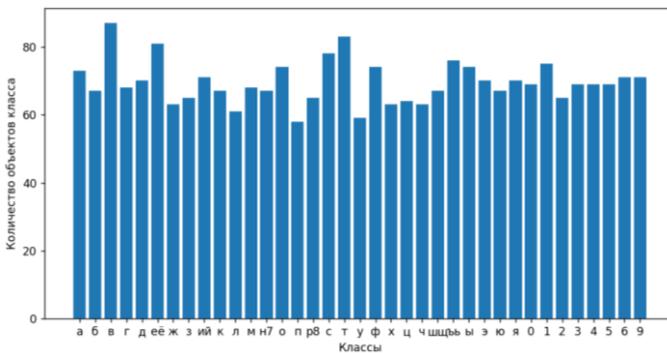


Рис. 3. Объем по каждой букве

Данные были разделены на обучающую и тестовую выборку в примерном соотношении 4:1. Для выявления оптимального результата классификации жестов были протестированы различные модели классификаторов [2]:

- логистическая регрессия;
- k ближайших соседей;
- дерево принятия решений;
- случайный лес;
- метод опорных векторов;
- градиентный бустинг;
- наивный байесовский классификатор.

А также модификации данных моделей. Для оценки классификаторов использовались такие метрики как recall и precision [7].

Precision можно интерпретировать как долю объектов, названных классификатором положительными и при этом действительно являющимися положительными, а recall показывает, какую долю объектов положительного класса из всех объектов положительного класса определил алгоритм.

Результаты исследования различных моделей классификаторов представлены на рис. 4.

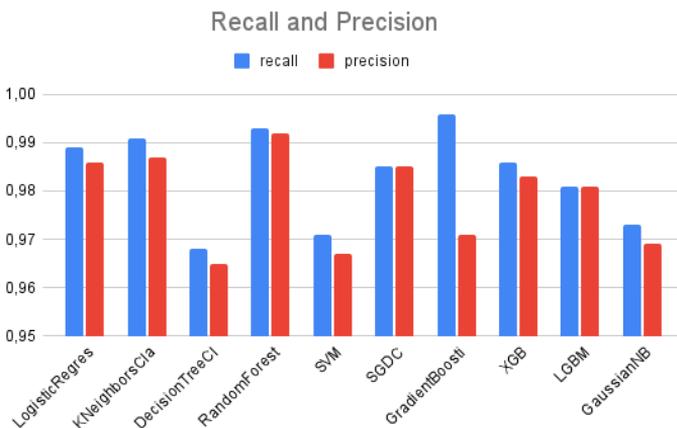


Рис. 4. Статистика по моделям классификатора

По итогам исследования в качестве итогового варианта была выбрана модель — случайный лес [5]. Средняя точность распознавания на тестовой выборке составила 99.2%. Ошибки приходились на жесты формы рук которых очень схожи, такие как е-с и ж-ф (рис. 5).

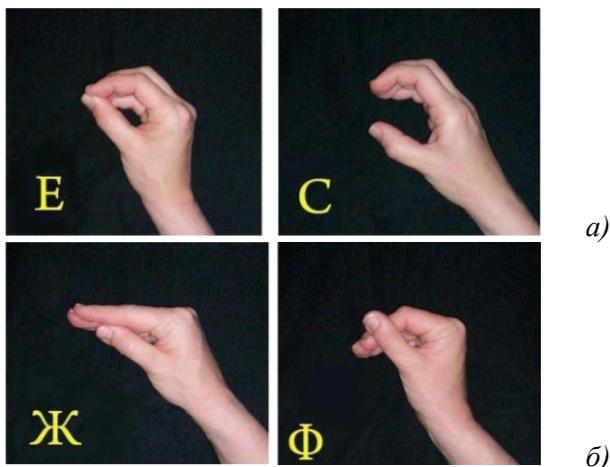


Рис. 5. Жесты со схожей формой рук е-с (а) ж-ф (б)

## Результаты

### *Клиентская часть и дизайн приложения*

Приложение было решено делать на платформе OS Android, т. к. она является популярной операционной системой среди мобильных устройств. Разработка клиентской части происходит в интегрированной среде разработки (IDE) Android Studio на языке программирования Kotlin.

Для разработки удобного пользовательского интерфейса было исследовано несколько приложений-аналогов по помощи в изучении языков такие как: DuoLingo, Lingualeo и т. п.

После экспериментального пути подбора цветов и расположения кнопок, приложение стало выглядеть следующим образом (рис. 6-10):



Рис. 6. Главное меню приложения.  
Часть: Основы

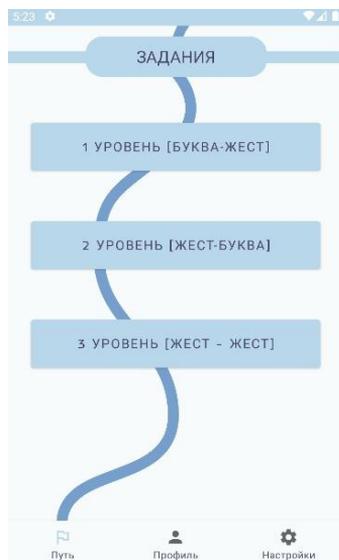


Рис. 7. Главное меню приложения.  
Часть: Задания



Рис. 8. Экран приложения после правильного ответа пользователя



Рис. 9. Условие задания

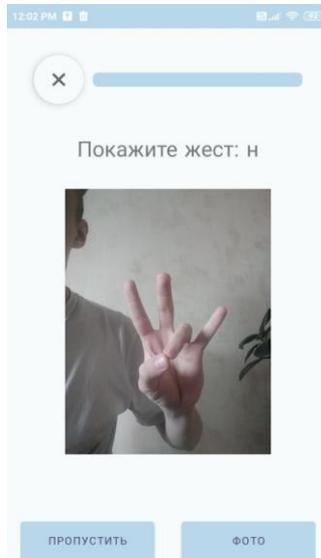


Рис. 10. Условие задания с фронтальной камерой.

Главное меню приложения состоит из двух частей, а именно: основные знания и тренировочные уровни. Уровни организованы на данном этапе следующим образом (рис. 8):

- всего 10 вопросов;
- при правильном ответе показывается диалоговое окно с похвальной фразой и кнопкой продолжения;
- при правильном ответе прогресс линия заполняется;
- при 10 правильных ответах возвращается на главное меню;
- имеется кнопка пропустить вопрос;
- при неправильном ответе высвечивается соответствующее сообщение;
- на 1 и 2 уровнях при новых прохождениях всегда разные варианты вопросов. Без повторов.

**Заключение и перспективы.** В ходе проделанной работы было реализовано мобильное приложение с минимальным функционалом для пользования. Написан сервер для принятия запроса (фото жеста)

с клиентской частью (пользователя) и итогового возвращения запроса (определенный жест).

В перспективе данного проекта лежат следующие задачи:

- определять жест на видеопотоке;
- создать адаптивную верстку пользовательского интерфейса;
- клонировать данное приложение на iPhone OS;
- реализовать кнопки раздела главного меню «Основы русского жестового языка»;
- оптимизация размера приложения;
- переход от статических жестов (Алфавит) к динамическим жестам.

## СПИСОК ЛИТЕРАТУРЫ

1. Документация фреймворка MediaPipe. — URL: <https://google.github.io/mediapipe/solutions/hands> (дата обращения: 28.05.2022). — Текст : электронный.
2. Документация библиотеки Scikit-Learn. — URL: <https://scikit-learn.org/stable> (дата обращения: 28.05.2022). — Текст : электронный.
3. Документация HTTP клиента OkHttp. — URL: <https://square.github.io/okhttp> (дата обращения: 28.05.2022). — Текст : электронный.
4. Документация фреймворка Flask. — URL: <https://flask.palletsprojects.com/en/2.1.x> (дата обращения: 28.05.2022). — Текст : электронный.
5. Чистяков С. П. Случайные леса: обзор / С. П. Чистяков. — Текст : непосредственный // Труды Карельского научного центра Российской академии наук. — 2013. — № 1. — С. 117-136.
6. Рюмин Д. Метод автоматического видеоанализа движений рук и распознавания жестов в человеко-машинных интерфейсах. — Текст : непосредственный // Научно-технический вестник информационных технологий, механики и оптики. — 2020. — Т. 20, № 4. — С. 525-531.
7. Мухамедиев Р. И. Таксономия методов машинного обучения и оценка качества классификации и обучаемости / Р. И. Мухамедиев, Е. Л. Мухамедиева, Я. И. Кучин. — Текст : электронный // Cloud of science. — 2015. — Т. 2, № 3. — URL: [http://cloudofscience.ru/publications/archive/cos\\_2\\_3/](http://cloudofscience.ru/publications/archive/cos_2_3/) (дата обращения: 28.05.2022).