Должиков Валерий Михайлович
бакалавр, студент
Ветчинов Владислав Олегович
бакалавр, студент
Научный руководитель
Лебедев Константин Андреевич
д-р физ.-мат. наук, профессор
ФГБОУ ВО «Кубанский государственный университет»
г. Краснодар, Краснодарский край

ИНТЕГРАЦИЯ ТЕХНОЛОГИЙ КОМПЬЮТЕРНОГО ЗРЕНИЯ В МОБИЛЬНЫЕ ПРИЛОЖЕНИЯ НА FLUTTER: ПОДХОДЫ, ИНСТРУМЕНТЫ И ОБРАЗОВАТЕЛЬНЫЙ ПОТЕНЦИАЛ

Аннотация: *в статье рассматриваются способы интеграции технологий компьютерного зрения в мобильные приложения, разрабатываемые на фреймворке Flutter. Проанализированы три подхода: использование предобученных моделей через Google ML Kit, запуск пользовательских моделей TensorFlow Lite на устройстве и обращение к облачным API (Google Cloud Vision, Firebase ML). Для каждого подхода описаны архитектура взаимодействия, ограничения и типичные сценарии применения. Особое внимание уделено образовательному потенциалу: возможности создания учебных проектов, объединяющих мобильную разработку и машинное обучение в рамках одного приложения.*

Ключевые слова: *компьютерное зрение, Flutter, машинное обучение, Google ML Kit, TensorFlow Lite, мобильное приложение, распознавание объектов, образовательные технологии.*

Технологии компьютерного зрения за последнее десятилетие прошли путь от исследовательских лабораторий до повседневных приложений: камера смартфона распознаёт лица, сканирует документы, переводит вывески и определяет породу собаки по фотографии. Для студентов, изучающих мобильную разработку, интеграция подобных возможностей в собственное приложение

представляет практический интерес и одновременно расширяет представление о том, какие задачи способно решать программное обеспечение на мобильном устройстве [1].

Фреймворк Flutter, благодаря механизму platform channels и развитой экосистеме пакетов, позволяет подключать средства компьютерного зрения без перехода к нативному коду. В настоящей статье рассмотрены три принципиально различных подхода к такой интеграции, каждый из которых предъявляет свои требования к инфраструктуре и квалификации разработчика.

Подход 1: *Google ML Kit*. Google ML Kit – набор готовых API машинного обучения, работающих непосредственно на устройстве без подключения к интернету. Для Flutter доступен через семейство пакетов google_mlkit_* (отдельный пакет для каждой задачи): распознавание текста (google_mlkit_text_recognition), детекция лиц (google_mlkit_face_detection), распознавание штрихкодов (google_mlkit_barcode_scanning), классификация изображений (google_mlkit_image_labeling), определение позы тела (google_mlkit_pose_detection) и другие [2].

Архитектура взаимодействия выглядит следующим образом: камера Flutter (через пакет camera) передаёт кадры в формате InputImage, который поступает на вход соответствующего детектора ML Kit. Детектор возвращает результат – список распознанных объектов с координатами, текстом или метками – в виде Dart-объектов, которые интерфейс отображает поверх изображения камеры.

Для образовательного контекста ML Kit привлекателен минимальным порогом входа: студенту не нужно разбираться в архитектурах нейронных сетей, собирать датасеты и обучать модели. Достаточно подключить пакет, настроить камеру и обработать результат. При этом обработка происходит на устройстве, что означает работу без интернета и отсутствие задержек сетевого обмена. Типичный учебный проект – сканер текста с камеры или считыватель QR-кодов – реализуется за одно-два занятия.

Ограничения ML Kit: набор задач фиксирован (нельзя обучить модель на собственных данных), точность распознавания текста на русском языке ниже,

чем на латинице, а детекция объектов ограничена несколькими сотнями общих категорий (Google Base Classifier) без возможности расширения [3].

Подход 2: *TensorFlow Lite*. TensorFlow Lite (TFLite) – фреймворк для запуска моделей машинного обучения на мобильных устройствах и встраиваемых системах. В отличие от ML Kit, TFLite позволяет использовать любую модель, обученную в TensorFlow, PyTorch (через конвертацию) или другом фреймворке, при условии её конвертации в формат.tflite. Для Flutter доступен пакет tflite_flutter, предоставляющий низкоуровневый API для загрузки модели и выполнения инференса [4].

Типичный сценарий: студент обучает модель классификации изображений в Google Colab (например, распознавание рукописных цифр MNIST или классификация растений), экспортирует её в формат TFLite, помещает в папку assets Flutter-проекта и загружает при запуске приложения. Входное изображение с камеры преобразуется в тензор нужного размера, подаётся на вход модели, а результат (вектор вероятностей по классам) отображается в интерфейсе.

Образовательная ценность этого подхода существенно выше, чем у ML Kit: студент проходит полный цикл от подготовки данных и обучения модели до её развёртывания на мобильном устройстве. Это формирует понимание pipeline машинного обучения, а не только потребление готовых API. Однако порог входа значительно выше: необходимы базовые знания Python, TensorFlow и линейной алгебры, а также понимание форматов данных (размеры тензоров, нормализация пикселей, квантизация модели).

Дополнительная сложность – предобработка изображения. Камера Flutter возвращает кадры в формате YUV или BGRA, тогда как модель ожидает RGB-тензор фиксированного размера (например, 224×224×3). Конвертация между форматами требует побайтового преобразования, что для студента, не знакомого с низкоуровневой работой с изображениями, может стать серьёзным препятствием.

Подход 3: *облачные API*. Третий вариант – отправка изображения на облачный сервер, где его анализирует мощная модель, и получение результата по сети.

Наиболее доступные сервисы: Google Cloud Vision API, Firebase ML (облачная версия) и Amazon Rekognition. Во Flutter обращение к облачному API выполняется через HTTP-запрос: изображение кодируется в Base64, отправляется POST-запросом и возвращается JSON с результатами [5].

Преимущества облачного подхода: доступ к наиболее мощным и точным моделям, возможность обработки изображений любого разрешения, отсутствие нагрузки на процессор устройства. Для задач, требующих высокой точности (например, распознавание медицинских изображений или анализ документов), облачные API остаются безальтернативным выбором.

Недостатки: зависимость от интернет-подключения, задержка ответа (100–2000 мс в зависимости от сервиса и размера изображения), стоимость (большинство сервисов тарифицируют количество запросов – бесплатный лимит обычно составляет 1000 запросов в месяц) и необходимость передачи пользовательских изображений на внешний сервер, что поднимает вопросы приватности.

Сравнение трёх подходов по ключевым критериям представлено в таблице 1.

Таблица 1

Сравнение подходов к интеграции компьютерного зрения во Flutter

Критерий	Google ML Kit	TensorFlow Lite	Облачные API
Работа офлайн	Да	Да	Нет
Кастомные модели	Нет	Да	Частично
Порог входа	Низкий	Высокий	Средний
Точность	Средняя	Зависит от модели	Высокая
Латентность	10–50 мс	50–200 мс	100–2000 мс
Стоимость	Бесплатно	Бесплатно	Платно (лимиты)
Приватность данных	На устройстве	На устройстве	Передача на сервер
Образовательная ценность	Средняя	Высокая	Средняя

Образовательные сценарии. На основе проведённого анализа можно предложить градацию учебных проектов по уровню сложности. Начальный уровень: сканер QR-кодов или распознаватель текста на базе ML Kit (1–2 занятия, не

требует знаний ML). Средний уровень: классификатор объектов с камеры, использующий предобученную модель MobileNet через TFLite (3–4 занятия, требует базового понимания нейросетей). Продвинутый уровень: приложение с моделью, обученной на собственном датасете – например, распознавание жестов языка жестов или определение состояния дорожного покрытия – через TFLite с предварительным обучением в Google Colab (проект на 4–6 недель, требует знания Python и TensorFlow) [6].

Отдельный интерес представляет связь компьютерного зрения с тематикой фитнес-приложений. Пакет google_mlkit_pose_detection позволяет отслеживать положение 33 ключевых точек тела человека в реальном времени. Это открывает возможность создания приложения-тренера, который анализирует правильность выполнения упражнений по видео с камеры – проект, естественным образом объединяющий мобильную разработку, компьютерное зрение и предметную область физической культуры.

Таким образом, экосистема Flutter предоставляет достаточный инструментарий для интеграции технологий компьютерного зрения на всех уровнях сложности – от подключения готового API до запуска пользовательских моделей. Для образовательных программ это означает возможность выстроить непрерывную траекторию: от первого знакомства с машинным зрением через ML Kit до полноценного ML-проекта с обучением и развёртыванием собственной модели, и всё это в рамках единого фреймворка и языка программирования.

Заключение.

Проведённый анализ трёх подходов к интеграции компьютерного зрения в мобильные приложения на Flutter показал, что выбор инструмента определяется балансом между простотой освоения, гибкостью и требованиями к инфраструктуре. Google ML Kit обеспечивает быстрый старт и работу в офлайн-режиме за счёт готовых предобученных моделей, не требуя от разработчика знания машинного обучения. TensorFlow Lite предоставляет полный контроль над моделью и максимальную образовательную ценность: студент проходит весь цикл от обучения до развёртывания, формируя глубокое понимание ML-pipeline. Облачные

API открывают доступ к наиболее мощным моделям, однако требуют постоянного интернет-соединения и несут финансовые издержки при масштабировании.

Предложенная градация учебных проектов – от сканера QR-кодов на ML Kit до приложения с собственной обученной моделью через TFLite – позволяет выстроить непрерывную образовательную траекторию в рамках единого фреймворка и языка программирования. Экосистема Flutter тем самым становится эффективной платформой не только для мобильной разработки, но и для практического освоения технологий машинного обучения и компьютерного зрения.

Список литературы

1. Форсайт Д. Компьютерное зрение. Современный подход / Д. Форсайт, Ж. Понс. – М.: Вильямс, 2018. – 960 с. – ISBN 978-5-8459-0542-0.

2. Google ML Kit documentation. URL: https://developers.google.com/ml-kit (date of request: 10.04.2026).

3. Flutter camera plugin documentation. URL: https://pub.dev/packages/camera (date of request: 10.04.2026).

4. TensorFlow Lite documentation. URL: https://www.tensorflow.org/lite (дата обращения: 10.04.2026).

5. Google Cloud Vision API documentation. URL: https://cloud.google.com/vision (дата обращения: 10.04.2026).

6. Biessek A. Flutter for Beginners. 2nd ed. Packt Publishing, 2023. 370 p. ISBN 978-1800565999.