

Разработка программного сервиса детекции движения глаз на основе методов компьютерного зрения

Development of a software service for eye movement detection based on computer vision methods

УДК 004.93

Получено: 19.01.2026

Одобрено: 22.02.2026

Опубликовано: 25.03.2026

Буценко Е.В.

Канд. экон. наук, доцент кафедры бизнес-информатики, ФГБОУ ВО «Уральский государственный экономический университет», г. Екатеринбург
e-mail: evl@usue.ru

Butsenko E.V.

Candidate of Economic Sciences, Associate Professor, Department of Business Informatics, Ural State University of Economics, Yekaterinburg
e-mail: evl@usue.ru

Аннотация

В статье рассматривается проблема создания программного сервиса для распознавания положения и движения глаз пользователя в режиме реального времени. Актуальность исследования обусловлена растущим спросом на бесконтактные интерфейсы человеко-машинного взаимодействия и интеллектуальные системы анализа визуального внимания. Целью работы является разработка и реализация приложения на языке Java с использованием библиотеки OpenCV, выполняющего детекцию лица и области глаз. Проведен анализ существующих методов детекции объектов, обоснован выбор классического алгоритма Виолы-Джонса, обеспечивающего баланс между производительностью и точностью в условиях ограниченных вычислительных ресурсов. Предложена архитектура программного сервиса, описаны алгоритмы предварительной обработки видеопотока и каскадной классификации. В результате тестирования подтверждена работоспособность разработанного решения в реальном времени. Статья содержит листинг основных модулей программы и анализ полученных результатов.

Ключевые слова: компьютерное зрение, распознавание лиц, детекция глаз, алгоритм Виолы-Джонса, OpenCV, Java, видеопоток, реальное время, каскадный классификатор.

Abstract

The article addresses the problem of creating a software service for real-time user eye position and movement detection. The relevance of the research is driven by the growing demand for contactless human-computer interaction interfaces and intelligent visual attention analysis systems. The aim of the study is to develop and implement a Java-based application using the OpenCV library to perform face and eye region detection. An analysis of existing object detection methods is conducted, justifying the choice of the classical Viola-Jones algorithm, which provides a balance between performance and accuracy under limited computational resources. The architecture of the software service is proposed, and algorithms for video stream preprocessing and cascade classification are described. Testing confirmed the real-time functionality of the developed solution. The article includes a listing of key program modules and an analysis of the obtained results.

Keywords: computer vision, face detection, eye detection, Viola-Jones algorithm, OpenCV, Java, video stream, real-time, cascade classifier.

Введение

Современный этап развития информационных технологий характеризуется активной интеграцией методов искусственного интеллекта в повседневные и промышленные системы. Компьютерное зрение занимает важное место в этом процессе, предоставляя инструменты для автоматического анализа визуальной информации. Одним из динамично развивающихся направлений является видеоаналитика поведения пользователя, в частности, отслеживание направления взгляда и движения глаз. Технологии трекинга глаз находят применение в биометрической аутентификации, адаптивных пользовательских интерфейсах, системах оценки внимания оператора, маркетинговых исследованиях, где, например, с помощью айтрекинга анализируют визуальное внимание потребителей к товарам на виртуальных полках интернет-магазинов и прогнозируют вероятность выбора продукта [1; 2].

Несмотря на высокую точность, достигаемую современными методами глубокого обучения, их применение в условиях ограниченных аппаратных ресурсов или необходимости обеспечения высокой скорости обработки не всегда оправдано. Классические методы компьютерного зрения, предложенные еще в начале 2000-х годов, сохраняют актуальность благодаря вычислительной эффективности и простоте реализации [3]. В связи с этим, задача разработки легковесного программного сервиса детекции глаз, способного функционировать в режиме реального времени на персональных компьютерах, является актуальной.

Настоящее исследование направлено на создание такого сервиса с использованием языка Java и библиотеки OpenCV. Основное внимание уделяется выбору оптимального метода детекции, проектированию архитектуры приложения и его практической реализации.

1. Анализ литературы и постановка задачи исследования

Теоретической базой для решения задач детекции объектов на изображениях служат фундаментальные труды в области цифровой обработки изображений и компьютерного зрения [4; 5]. В работе Р. Гонсалеса и Р. Вудса [4] подробно освещены методы предварительной обработки изображений, такие как преобразование цветовых пространств, выравнивание гистограммы и пороговая обработка, которые являются необходимым этапом перед детекцией.

Основным методом, обеспечивающим детекцию объектов в реальном времени, стал алгоритм, предложенный П. Виолой и М. Джонсом [6]. Он основан на использовании признаков Хаара, интегрального представления изображения для быстрого вычисления признаков, метода бустинга для отбора наиболее информативных признаков и каскадной структуры классификаторов. Данный алгоритм позволяет последовательно отсеивать области, не содержащие искомый объект, что обеспечивает высокую скорость работы. Как отмечают Г. Брэдки и А. Кэлер [7], реализация этого алгоритма в библиотеке OpenCV стала стандартом индустрии для задач детекции лиц и простых объектов.

Альтернативой классическим методам являются подходы, основанные на глубоких нейронных сетях, например, на архитектурах сверточных нейронных сетей (CNN) [8; 9]. Такие методы, как YOLO (You Only Look Once) или SSD (Single Shot MultiBox Detector), демонстрируют превосходную точность, однако требуют значительных вычислительных затрат, особенно на этапе обучения и инференса, а также наличия размеченных наборов данных большого объема [9].

Анализ литературы позволяет сформулировать цель исследования: разработать программный сервис для детекции лица и области глаз пользователя в режиме реального времени, обеспечивающий приемлемую точность при минимальных требованиях к вычислительным ресурсам. Для достижения цели необходимо решить следующие задачи: провести сравнительный анализ методов детекции и обосновать выбор технологии; разработать алгоритм обработки видеопотока; спроектировать архитектуру приложения; выполнить программную реализацию на языке Java с использованием библиотеки OpenCV; провести экспериментальное тестирование разработанного решения.

2. Материалы и методы

2.1. Обоснование выбора технологий и методов

Исходя из цели исследования – создание легковесного приложения для работы в реальном времени – в качестве основного метода детекции был выбран алгоритм Виолы-Джонса. Данный выбор обусловлен его высокой скоростью работы, которая достигается за счет каскадной структуры классификаторов, позволяющей быстро отбрасывать фоновые области изображения [6]. В условиях использования веб-камеры стандартного разрешения и необходимости обработки потока кадров без существенных задержек, этот метод обеспечивает необходимый баланс между скоростью и точностью.

Важно отметить, что выбор метода детекции должен осуществляться не только с учётом производительности модели, но и доступных вычислительных ресурсов и конкретных требований поставленной задачи [10]. Для нейромаркетинговых исследований, где требуется не только локализация глаз, но и оценка эмоциональных реакций (например, с помощью электроэнцефалографии), часто применяются гибридные схемы, сочетающие компьютерное зрение и регистрацию физиологических сигналов [11]

В качестве инструментария разработки выбрана открытая библиотека компьютерного зрения OpenCV (версия 4.x) [7; 13]. OpenCV предоставляет готовые, оптимизированные реализации алгоритма Виолы-Джонса (класс CascadeClassifier), а также обширный набор функций для работы с изображениями и видеопотоками (модули imgproc, videoio). Выбор языка программирования Java обоснован его кроссплатформенностью, строгой типизацией и наличием удобных средств для создания графического интерфейса пользователя [12]. Сочетание Java и OpenCV (через JavaCV или нативные обертки) позволяет создавать переносимые приложения с доступом к низкоуровневым функциям компьютерного зрения.

2.2. Алгоритм обработки видеопотока

Разработанный алгоритм работы сервиса представляет собой итеративный цикл обработки кадров, поступающих с веб-камеры. Основные этапы алгоритма включают:

1. **Инициализация:** загрузка библиотеки OpenCV, предварительно обученных XML-файлов каскадов Хаара для детекции лица (haarcascade_frontalface_default.xml) и глаз (haarcascade_eye.xml), открытие видеопотока с камеры (индекс 0).

2. **Захват кадра:** получение текущего кадра в формате BGR.

3. **Предварительная обработка:**

–преобразование кадра в оттенки серого (цветовое пространство GRAY) для уменьшения объема данных и соответствия требованиям каскадного классификатора (cvtColor).

–выравнивание гистограммы полученного полутонового изображения (equalizeHist) для улучшения контрастности и компенсации неравномерности освещения [4].

–зеркальное отображение кадра (flip) для более интуитивного взаимодействия пользователя.

4. **Детекция лица:** применение каскадного классификатора к обработанному полутоновому изображению. Поиск областей, содержащих лицо, с заданными параметрами масштабирования (scaleFactor) и минимальным размером объекта.

5. **Детекция глаз:** для каждой обнаруженной области лица (Region of Interest, ROI) выполняется поиск глаз. Детекция производится в пределах ROI лица на полутоновом изображении, что существенно сокращает область поиска и повышает точность и скорость.

6. **Визуализация:** на оригинальном цветном кадре отрисовываются прямоугольники вокруг обнаруженного лица (зеленый цвет) и вокруг каждого глаза (синий цвет).

7. **Вывод:** обработанный кадр отображается в графическом окне приложения. Цикл повторяется до закрытия окна пользователем.

3. Реализация программного сервиса

Программная реализация выполнена в интегрированной среде разработки Visual Studio Code с использованием системы сборки Maven для управления зависимостями. Основная

логика приложения инкапсулирована в класс GazeTrackerApp, наследующий от JFrame для создания графического окна.

Основные фрагменты программного кода, реализующего описанный алгоритм, представлены в Листинге 1.

```
// Листинг 1. Основной цикл обработки видеопотока
public void start() {
    VideoCapture cap = new VideoCapture(0); // Открытие камеры
    // ... проверка открытия ...
    Mat frame = new Mat();
    Mat gray = new Mat();
    while (running) {
        if (!cap.read(frame) || frame.empty()) continue;
        // 1. Предварительная обработка
        cvtColor(frame, gray, COLOR_BGR2GRAY);
        equalizeHist(gray, gray);
        // 2. Детекция лица
        RectVector faces = new RectVector();
        faceCascade.detectMultiScale(gray, faces);
        for (long i = 0; i < faces.size(); i++) {
            Rect face = faces.get(i);
            rectangle(frame, face, new Scalar(0, 255, 0, 0), 2); // Рамка лица
            // 3. Детекция глаз в области лица
            Mat faceRoi = new Mat(gray, face);
            RectVector eyes = new RectVector();
            eyeCascade.detectMultiScale(faceRoi, eyes);
            for (long e = 0; e < eyes.size(); e++) {
                Rect er = eyes.get(e);
                // Преобразование координат относительно исходного кадра
                Rect eyeAbs = new Rect(face.x() + er.x(), face.y() + er.y(), er.width(), er.height());
                rectangle(frame, eyeAbs, new Scalar(255, 0, 0, 0), 2); // Рамка глаз
            }
            break; // Обработка только первого лица
        }
        // 4. Отображение результата
        flip(frame, frame, 1); // Зеркальное отображение
        videoLabel.setIcon(new ImageIcon(matToBufferedImage(frame)));
        try { Thread.sleep(16); } catch (InterruptedException ignored) {} // ~60 FPS
    }
    cap.release();
}
```

Функция detectMultiScale инкапсулирует всю сложность алгоритма Виолы-Джонса. Для преобразования внутреннего представления изображения OpenCV (Mat) в BufferedImage, используемый для отображения в Swing-компонентах, реализована вспомогательная функция matToBufferedImage, осуществляющая копирование байтовых массивов данных.

4. Результаты тестирования и обсуждение

Тестирование разработанного программного сервиса проводилось на компьютере под управлением macOS. В ходе испытаний оценивались корректность детекции, стабильность работы и производительность.

Установлено, что программа успешно подключается к веб-камере и обрабатывает видеопоток в реальном времени без ощутимых задержек при стандартном освещении. Визуализация результатов в виде рамок вокруг лица и глаз происходит корректно (пример

выходного изображения приведен на рис. 1). Система продемонстрировала устойчивость к небольшим поворотам головы и изменению мимики.

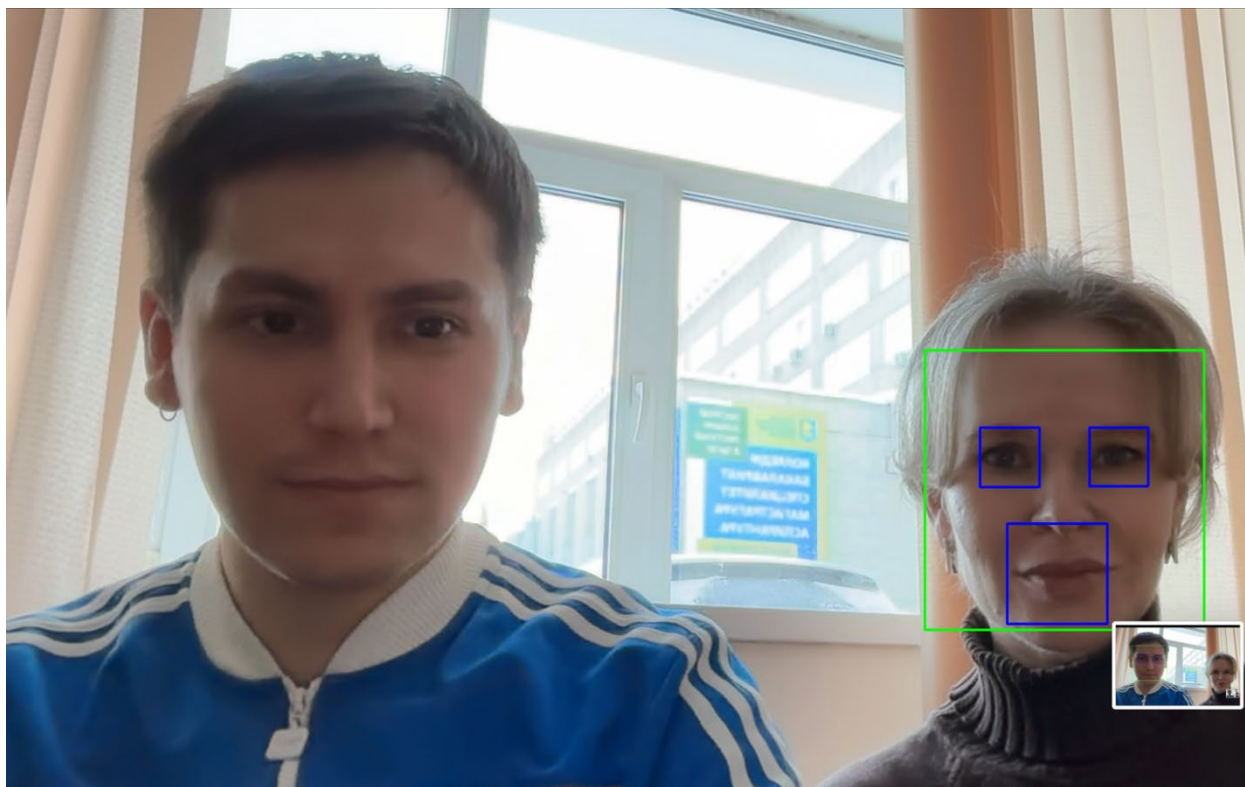


Рис. 1. Результат работы программы

В ходе анализа работы были выявлены ограничения, характерные для выбранного метода:

- **Чувствительность к освещению:** при сильном боковом освещении или в условиях низкой освещенности качество детекции глаз снижается, что связано с падением контрастности в области ROI.
- **Ложные срабатывания:** в редких случаях классификатор глаз может ошибочно детектировать брови, тени или другие темные области на лице как глаза.
- **Ориентация лица:** метод оптимизирован для детекции анфас. При сильных поворотах головы детекция становится нестабильной.

Несмотря на указанные ограничения, поставленная цель была достигнута: разработанный сервис обеспечивает базовый функционал детекции движения глаз, достаточный для решения ряда прикладных задач, таких как контроль нахождения пользователя перед экраном или реализация простых интерфейсов, управляемых взглядом.

Заключение

В результате выполненного исследования был разработан и реализован программный сервис для распознавания положения лица и глаз пользователя в режиме реального времени. Основные научные и практические результаты работы заключаются в следующем:

1. Проведен анализ современных методов детекции объектов, на основе которого для реализации в условиях ограниченных вычислительных ресурсов обоснован выбор алгоритма Виолы-Джонса.
2. Предложена архитектура и разработан алгоритм обработки видеопотока, включающий этапы предварительной обработки изображения и каскадной детекции.
3. Выполнена практическая реализация сервиса на языке Java с использованием библиотеки OpenCV, подтвердившая эффективность предложенных решений.

4. В ходе экспериментального тестирования выявлены как достоинства (высокая скорость работы), так и ограничения (чувствительность к освещению) разработанного подхода.

Дальнейшее развитие работы может быть направлено на повышение устойчивости детекции путем комбинирования классических методов с элементами машинного обучения для коррекции освещения, а также на внедрение модуля анализа направления взгляда (gaze estimation) на основе геометрических параметров расположения зрачка.

Литература

1. Duchowski A.T. Eye Tracking Methodology: Theory and Practice / A.T. Duchowski. – 3rd ed. – Cham: Springer, 2017. – 366 p.
2. Калькова Н.Н. Потребительский выбор товаров в интернете и прогнозирование ассортимента методом ABC-анализа на основе метрик визуального нейромаркетинга / Н.Н. Калькова // Управленец. – 2025. – Т. 16, № 1. – С. 92-105. – DOI: 10.29141/2218-5003-2025-16-1-7.
3. Szeliski R. Computer Vision: Algorithms and Applications / R. Szeliski. – 2nd ed. – London: Springer, 2022. – 979 p.
4. Gonzalez R.C. Digital Image Processing / R. C. Gonzalez, R. E. Woods. – 4th ed. – New York: Pearson, 2018. – 1168 p.
5. Форсайт Д. Компьютерное зрение. Современный подход / Д. Форсайт, Ж. Понс. – М.: Вильямс, 2004. – 928 с.
6. Viola, P. Rapid Object Detection using a Boosted Cascade of Simple Features / P. Viola, M. Jones // Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). – Kauai, 2001. – Vol. 1. – P. I-511–I-518.
7. Bradski G. Learning OpenCV 4: Computer Vision with OpenCV Library / G. Bradski, A. Kaehler. – Sebastopol: O'Reilly Media, 2019. – 976 p.
8. Goodfellow I. Deep Learning / I. Goodfellow, Y. Bengio, A. Courville. – Cambridge: MIT Press, 2016. – 800 p.
9. Redmon J. You Only Look Once: Unified, Real-Time Object Detection / J. Redmon, S. Divvala, R. Girshick, A. Farhadi // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – Las Vegas, 2016. – P. 779–788.
10. Мухитдинова М.Х. Сравнительный анализ методологий и технологий машинного обучения / М. Х. Мухитдинова // Цифровые модели и решения. – 2025. – Т. 4, № 1. – С. 78-85. – DOI: 10.29141/2949-477X-2025-4-1-6.
11. Калькова Н.Н. Влияние гендерного фактора на визуальное восприятие цвета упаковки потребителями на основе алгоритмов нейромаркетинга / Н.Н. Калькова // Управленец. – 2024. – Т. 15, № 2. – С. 108-123. – DOI: 10.29141/2218-5003-2024-15-2-8.
12. Oracle Corporation. The Java™ Tutorials [Электронный ресурс]. – Режим доступа: <https://docs.oracle.com/javase/tutorial/> (дата обращения: 28.02.2026).
13. OpenCV team. OpenCV: Cascade Classification [Электронный ресурс]. – Режим доступа: https://docs.opencv.org/master/db/d28/tutorial_cascade_classifier.html (дата обращения: 28.02.2026).