

Министерство образования и науки Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ  
Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра дискретной математики  
и информационных технологий

**Разработка системы распознавания лиц с использованием  
библиотеки компьютерного зрения OpenCV**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студента 5 курса 521 группы  
направления 09.03.01 «Информатика и вычислительная техника»  
факультета компьютерных наук и информационных технологий  
Шапошникова Кирилла Сергеевича

Научный руководитель  
ассистент каф. ДмиИТ

\_\_\_\_\_

дата, подпись

М.В. Белоконь

Заведующий кафедрой  
к. ф.-м.н., доцент

\_\_\_\_\_

дата, подпись

Л.Б. Тяпаев

Саратов 2017

## ВВЕДЕНИЕ

Системы обнаружения и распознавания лиц применяются для идентификации людей в биометрических системах безопасности или как дополнительный сервис, например, в социальных сетях. Такие системы обеспечивают высокую точность распознавания, что позволяет повысить уровень безопасности какого-либо устройства или сервиса, а также предоставить пользователям удобные возможности по поиску запечатленных на фото людей.

Обычно системы распознавания лиц используются крупными компаниями в качестве внутренних систем безопасности или в разрабатываемой ими продукции. В небольших же учреждениях, будь то учебные заведения или мелкие предприятия, такие системы применяются редко, вместо них обычно используются RFID-ключи. Хотя можно было бы существенно увеличить безопасность и удобство сотрудников, отказавшись от таких ключей в пользу систем распознавания лиц, что позволит исключить часто возникающие проблемы в виде потерянных или испортившихся ключей.

Актуальность работы заключается в разработке системы распознавания лиц, представляющей из себя веб-сервис, который можно интегрировать с внешними системами и использовать в составе системы безопасности.

В данной работе ставится цель разработать такую систему. Для достижения поставленной цели необходимо решить следующие задачи:

- изучить возможность применения библиотеки OpenCV для распознавания лиц, по литературным источникам;
- изучить технологию Spring Boot для разработки Java приложений;
- изучить технологию Spring Security для организации разграничения доступа в приложении;
- рассмотреть возможность использования MongoDB для хранения служебной информации системы;
- изучить методы разработки Java приложений с использованием Spring Beans;

- основываясь на выше перечисленных технологиях, реализовать систему распознавания лиц, представленную в виде веб-сервиса.

Структура выпускной квалификационной работы включает в себя следующие разделы: введение, обозначения и сокращения, семи глав, заключения и списка использованных источников.

В первой главе дается описание метода Виолы-Джонса, используемого для детектирования лиц в кадре видеопотока.

Во второй главе описывается алгоритм Fisherface с помощью которого происходит распознавание лица.

В третьей главе обосновывается выбор технологий для реализации веб-сервиса.

В четвертой главе описывается применение фреймворка Spring Boot и настройка проекта.

В пятой главе приводится описание способа подключения и использования библиотеки OpenCV в разрабатываемом веб-сервисе.

В шестой главе дано описание структуры базы данных.

В седьмой главе дается подробное описание принципов работы разрабатываемой системы.

## Основное содержание работы

**Метод Виолы-Джонса.** Алгоритм строится на следующих основных принципах:

- подаваемые на вход алгоритма изображения представляются в интегральной форме;
- для поиска непосредственно самого лица применяются признаки Хаара;
- для выбора подходящих признаков используется ускорение;
- имеется классификатор на вход которого подаются признаки и он решает истина это или ложь;
- для быстрого отбрасывания участков, где нет лиц, используются каскады признаков.

Интегральное представление – это матрица, размерность которой совпадает с размером входного изображения. Каждый элемент матрицы представляет собой сумму интенсивностей всех пикселей, находящихся левее и выше текущего элемента, рассчитываются элементы по формуле:

$$L(x, y) = \sum_{i=0, j=0}^{i \leq x, j \leq y} I(i, j) \quad (1)$$

где  $I(i, j)$  – яркость пикселя входящего изображения.

Это означает, что каждый элемент результирующей матрицы содержит сумму пикселей в прямоугольнике от  $(0, 0)$  до  $(x, y)$ . Расчет матрицы занимает линейное время, пропорциональное числу пикселей входного изображения, поэтому для расчета можно воспользоваться рекуррентной формулой:

$$L(x, y) = I(x, y) - L(x-1, y-1) + L(x, y-1) + L(x-1, y) \quad (2)$$

В методе Виолы-Джонса применяются прямоугольные признаки, которые называются признаками Хаара.

Реализация метода в библиотеке OpenCV на ряду со стандартными признаками использует дополнительные признаки. Значением признака для исследуемой области изображения является число

$$F = W - B \quad (3)$$

где  $W$  – сумма значений пикселей, находящихся в белых областях признака,

$B$  – сумма значений пикселей из темных частей признака, для их вычисления используется интегральное представление изображения [1].

Поскольку обучение классификаторов крайне трудоемкий процесс, то для его ускорения применяется специальная технологий бустинга (ускорения), которая из слабых классификаторов позволяет собрать сильный классификатор. В методе Виолы-Джонса используется AdaBoost. Формально алгоритм описывается следующим образом. Даны классификаторы  $k_t$  ( $t = \{1, 2, \dots, T\}$ ) ассоциированные с весом  $a_t$ , и вектор признаков  $X = (x_1, x_2, \dots, x_i, \dots, x_M)$ , где  $i$  – номер признака, помечаемый метками  $y_i = \{-1, +1\}$ :

1.  $D_1(i) = \frac{1}{m}$ , для  $i = 1, \dots, m$ ;

2. Для  $t = 1, \dots, T$ :

- 2.1. Найти классификатор  $h_t$  минимизирующий  $D_t(i)$  весовую ошибку

$$k_t = \arg \min_{k_j \in K} \varepsilon_j, \text{ где } \varepsilon_j = \sum_{i=1}^M D_t(i) I[y_i \neq k_j(x_i)]. \text{ Пока } \varepsilon_j < 0.5,$$

иначе выход;

- 2.2. установить для  $a_t$  вес  $a_t = \frac{1}{2} \log \left[ \frac{1 - \varepsilon_t}{\varepsilon_t} \right]$ , где  $\varepsilon_t$  – минимальная ошибка на предыдущем шаге;

- 2.3. Обновить весовые коэффициенты для признаков

$$D_{t+1} = \frac{D_t(i) \exp(-a_t y_i k_t(x_i))}{Z_t}, \text{ где } Z_t \text{ нормализует по всем признакам.}$$

По окончании алгоритма строится результирующий классификатор по

формуле  $K(X) = \text{sign} \left( \sum_{i=1}^T a_i k_i(X) \right)$  [2].

**Алгоритм Fisherface.** Для распознавания лица используется алгоритм Fisherface, который является потомком Eigenface и обеспечивает более высокую точность распознавания при изменениях освещения или выражения лица.

Eigenface в своей основе использует метод главных компонент (PCA). PCA – это один из способов уменьшить размерность данных, потеряв наименьшее количество информации. В PCA используется тренировочная выборка изображений размерности  $N$ , где каждое изображение –  $x_i$ , то есть вся выборка это  $\{x_1, x_2, \dots, x_n\}$ , также считается, что каждое изображение принадлежит одному из  $s$  классов  $\{X_1, X_2, \dots, X_s\}$ . Также существует линейное преобразование, отображающее исходное  $n$ -мерное пространство изображений в  $m$ -мерное пространство, где  $m < n$ . Новые вектора признаков  $y_k = \mathbb{R}^m$  определяются с помощью линейной трансформации:

$$y_k = W^T x_k, k = 1, 2, \dots, N \quad (4)$$

где  $W \in \mathbb{R}^{n \times m}$  матрица с ортонормальными столбцами.

Если полная матрица разброса определяется как

$$S_T = \sum_{k=1}^N (x_k - \mu)(x_k - \mu)^T \quad (5)$$

где  $k$  – номер изображения из тренировочной выборки,

$N$  – количество изображений в тренировочной выборке,

$\mu \in \mathbb{R}^n$  – среднее изображение из всех примеров,

то после применения линейной трансформации  $W^T$ , разброс трансформированных векторов признаков равен  $W^T S_T W$ . В PCA проекция  $W_{opt}$  выбирается таким образом, чтобы максимизировать определитель полной матрицы разброса проекции изображений из тренировочной выборки, то есть

$$W_{opt} = \underset{W}{arg\ max} |W^T S_T W| = [w_1 w_2 \dots w_m] \quad (6)$$

где  $\{w_i \mid i = 1, 2, \dots, m\}$  – множество  $n$ -мерных собственных векторов из матрицы  $S_T$  соответствующих  $m$  наибольшим собственным значениям. Так как эти собственные вектора имеют ту же размерность, что и исходное изображение, они называются собственными изображениями или собственными лицами.

В свою очередь алгоритм Fisherface был разработан для улучшения результатов распознавания при изменении освещения или выражения лица. В его основе лежит линейный дискриминантный анализ LDA, который выбирает такое линейное пространство, которое максимизирует отношение

межклассового разброса к внутриклассовому. Алгоритм работает с матрицей межклассового разброса, определяемой как

$$S_B = \sum_{i=1}^c N_i (\mu_i - \mu) (\mu_i - \mu)^T \quad (7)$$

а также с матрицей внутриклассового разброса

$$S_W = \sum_{i=1}^c \sum_{x_k \in X_i} (x_k - \mu_i) (x_k - \mu_i)^T \quad (8)$$

где  $\mu_i$  – среднее изображение из класса  $X_i$

$N_i$  – количество примеров в классе  $X_i$ .

При этом полный разброс как раз будет определяться как сумма разброса внутри класса и разброса между классами:

$$S_T = S_W + S_B \quad (9)$$

Если матрица  $S_W$  невырожденная, то проекция  $W_{opt}$  выбирается как матрица с ортонормальными столбцами, которая, как было описано выше, максимизирует отношение определителя матрицы межклассового разброса к определителю матрицы внутриклассового разброса, то есть

$$W_{opt} = \arg \max_W \frac{|W^T S_B W|}{|W^T S_W W|} = [w_1 w_2, \dots, w_m] \quad (10)$$

где  $\{w_i \mid i = 1, 2, \dots, m\}$  – множество обобщенных собственных векторов из  $S_B$  и  $S_W$ , соответствующих  $m$  наибольшим обобщенным собственным значениям  $\{\lambda_i \mid i = 1, 2, \dots, m\}$ , то есть [3]

$$S_B w_i = \lambda_i S_W w_i, \quad i = 1, 2, \dots, m \quad (11)$$

**Определение функционала и выбор технологий.** Была реализована система, обладающая следующим свойствами:

- распознавание лиц людей, попавших в объектив камеры;
- возможность добавления новых камер и известных людей;
- возможность обучения алгоритма путем загрузки файлов с шаблонами через веб-форму;
- наличие двух ролей для пользователей (USER, ADMIN);

- возможность запуска распознавания с камеры и остановка его только пользователями с ролью ADMIN;
- пользователи с ролью USER имеют возможность подключиться к существующей трансляции и просматривать распознанных в данный момент времени людей;
- все распознавания удачные или нет записываются в базу данных для хранения истории;
- при распознавании отправляется HTTP-сообщение посредством REST-сервиса на заранее настроенный адрес, чтобы поддержать интеграцию со сторонними системами.

Для разработки данной системы использовался язык программирования Java. Сборка проекта осуществлялась с помощью Gradle. В качестве платформы применялся фреймворк Spring Boot [4], а также Spring MVC для поддержки трехслойной архитектуры приложения. Для обеспечения безопасности и разграничения ролей пользователей в приложении использовался фреймворк Spring Security [5]. В качестве базы данных был выбран MongoDB из-за отсутствия четко выраженной модели данных в приложении и наличия встроенной файловой системы GridFS, хорошо подошедшей для хранения обучающей выборки для Fisherface. Для работы с базой данных использовался фреймворк Spring Data [6].

**Применение Spring Boot и настройка проекта.** Для подключения Spring Boot к проекту используется плагин `org.springframework.boot` для системы сборки Gradle. Также для получения дополнительных возможностей, увеличивающих удобство разработки. В качестве точки входа в приложение реализован класс `Application`, который является наследником `SpringBootServletInitializer` для поддержки развертывания приложения в полноценных контейнерах-сервлетов.

**Подключение OpenCV к проекту.** Для сборки OpenCV использовалась Visual Studio 2017 Community. Пересборка требуется по причине отсутствия в стандартной поставке библиотеки алгоритмов распознавания лиц и,

соответственно, API для вызова этого функционала из Java. Также был разработан специальный класс, для правильной загрузки нативной библиотеки OpenCV при использовании dev-tools из Spring Boot.

**Настройка базы данных.** База данных на основе MongoDB содержит следующие коллекции.

- cameras – содержит зарегистрированные в системе камеры;
- faces – используется для сопоставления людей, зарегистрированных в системе и принадлежащих им фотографий;
- history – хранит историю всех распознаваний в системе;
- humans – список людей, зарегистрированных в системе;
- users – хранит учетные данные пользователей, имеющих право пользоваться системой;
- faces\_to\_training.chunks – генерируется автоматически MongoDB для хранения файлов в GridFS;
- faces\_to\_training.files – генерируется автоматически MongoDB для хранения файлов в GridFS;

**Описание системы.** При входе в систему пользователь попадает на страницу */login*, показана на рисунке 1, где нужно ввести имя пользователя и пароль

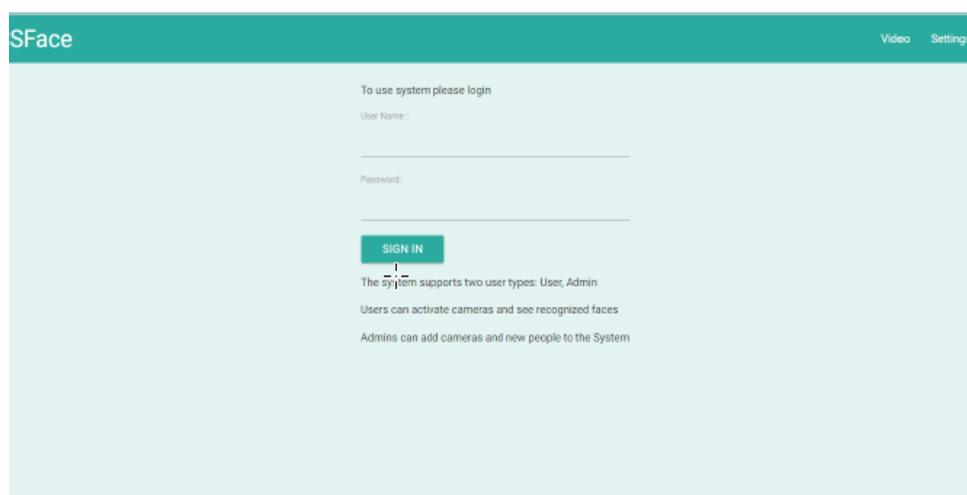


Рисунок 1 – Изображение формы входа пользователя в систему.

После входа в систему пользователь перенаправляется на страницу */video* – эта страница считается домашней. Затем необходимо обучить алгоритм, для этого

нужно перейти в соответствующий раздел страницы */settings*, как показано на рисунке 2, и загрузить архив с фотографиями.

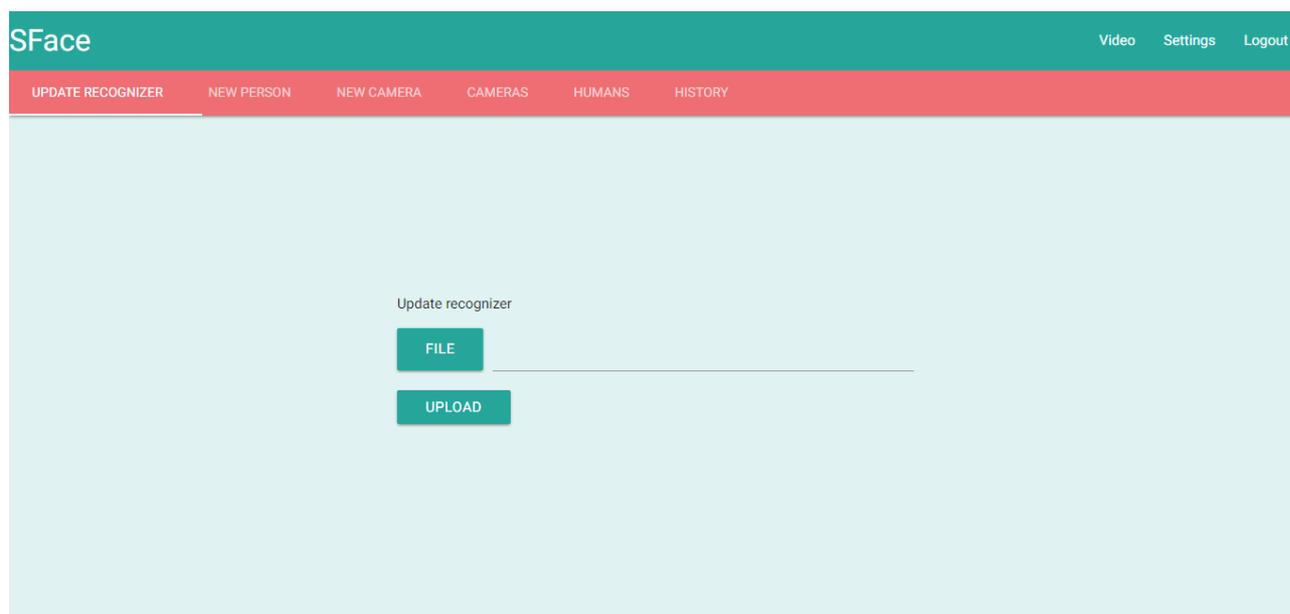


Рисунок 2 – Форма загрузки архива с фотографиями для обучения системы.

Для запуска распознавания с камеры нужно перейти в раздел *cameras*, выбрать камеру и нажать кнопку *Start*, вид раздела представлен на рисунке 3.

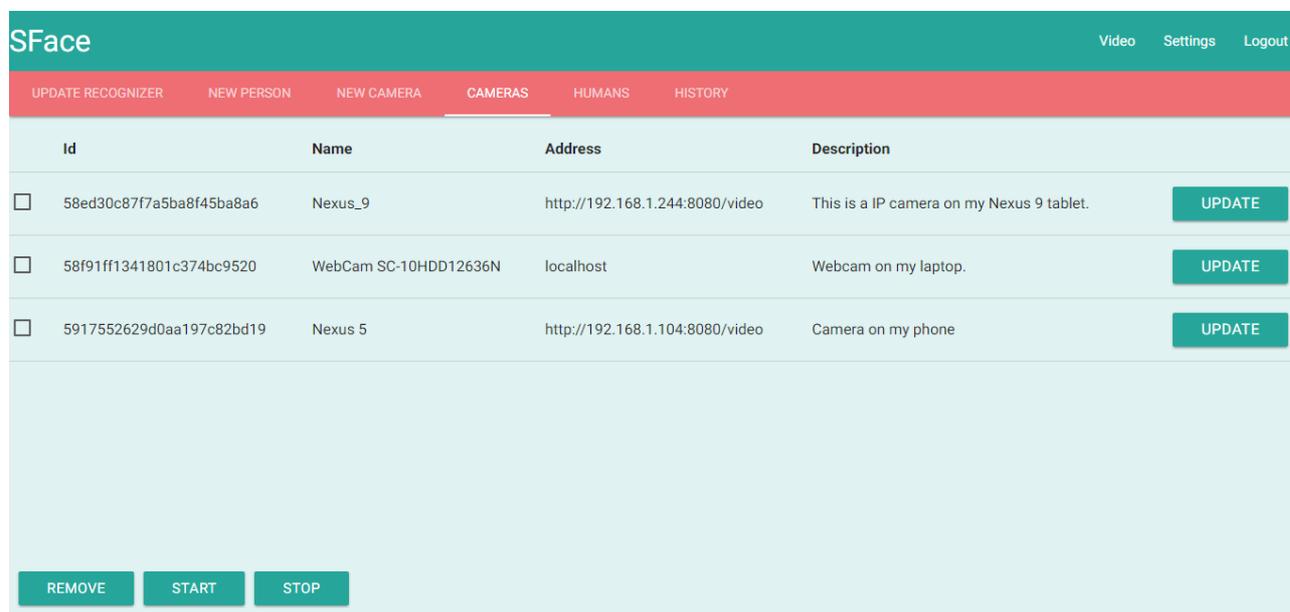


Рисунок 3 – Список добавленных в систему камер.

Как было описано выше, при любом распознавании будь оно удачное или нет, добавляется запись в историю, которую можно увидеть на рисунке 4 и отправляется HTTP-запрос на заранее указанный для конкретной камеры адрес.

SFace						
UPDATE RECOGNIZER   NEW PERSON   NEW CAMERA   CAMERAS   HUMANS   HISTORY						
Date	Human Id	First Name	Last Name	Camera Id	Camera	
Sat, 13 May 2017 19:49:42 GMT	111111	Kirill	Shaposhnikov	5917552629d0aa197c82bd19	Nexus 5	
Sat, 13 May 2017 19:49:49 GMT	111111	Kirill	Shaposhnikov	5917552629d0aa197c82bd19	Nexus 5	
Sat, 13 May 2017 19:49:56 GMT	111111	Kirill	Shaposhnikov	5917552629d0aa197c82bd19	Nexus 5	
Sat, 13 May 2017 19:50:03 GMT	111111	Kirill	Shaposhnikov	5917552629d0aa197c82bd19	Nexus 5	
Sat, 13 May 2017 19:50:10 GMT	111111	Kirill	Shaposhnikov	5917552629d0aa197c82bd19	Nexus 5	
Sat, 13 May 2017 19:50:17 GMT	-1	Unknown	Human	5917552629d0aa197c82bd19	Nexus 5	
Sat, 13 May 2017 19:50:25 GMT	-1	Unknown	Human	5917552629d0aa197c82bd19	Nexus 5	
Thu, 18 May 2017 19:44:10 GMT	111111	Kirill	Shaposhnikov	5917552629d0aa197c82bd19	Nexus 5	
Thu, 18 May 2017 19:44:19 GMT	111111	Kirill	Shaposhnikov	5917552629d0aa197c82bd19	Nexus 5	
Thu, 18 May 2017 19:44:26 GMT	111111	Kirill	Shaposhnikov	5917552629d0aa197c82bd19	Nexus 5	

Рисунок 4 – Общий вид раздела истории.

Чтобы запустить трансляцию с камеры нужно перейти на страницу */video* и выбрать активную камеру, после чего запустится трансляция и помимо видео с захваченными лицами будут отображаться распознанные лица, как показано на рисунке 5.

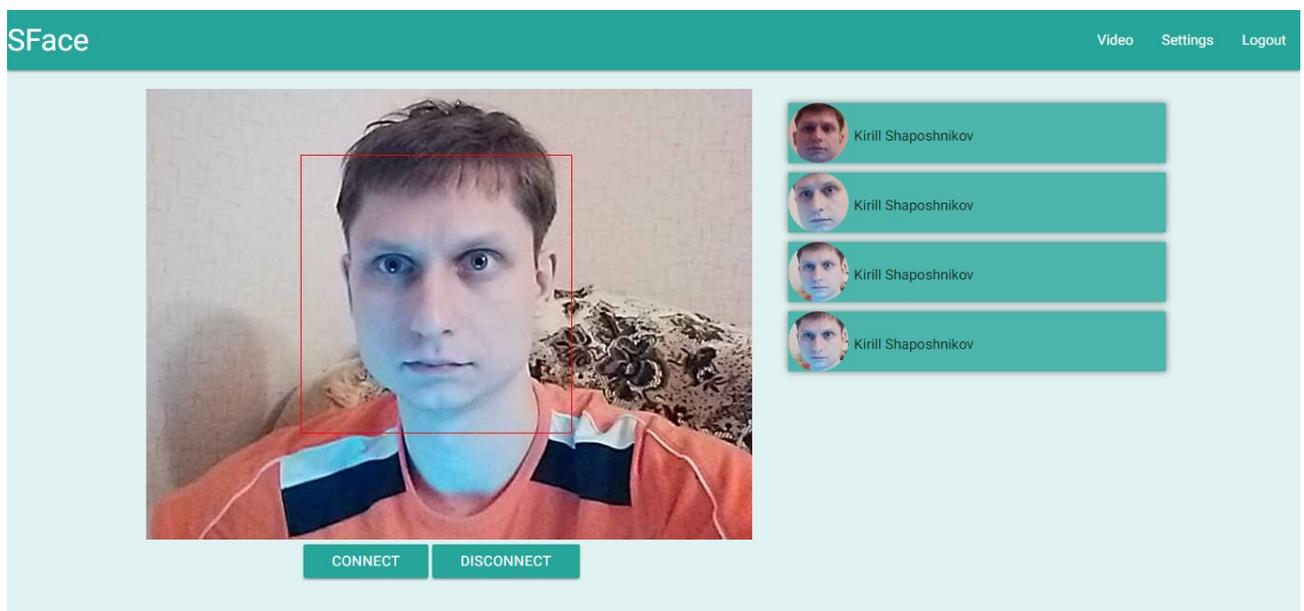


Рисунок 5 – Трансляция изображения с распознанными лицами.

## **ЗАКЛЮЧЕНИЕ**

В результате проделанной работы была изучена возможность применения библиотеки *OpenCV* для распознавания лиц в системах учета и контроля доступа. Кроме того была изучена и применена технология *Spring Boot* в качестве платформы для создания современных веб-приложений, обладающих широкими возможностями к дальнейшему расширению. В качестве системы хранения информации в приложении была рассмотрена и применена документоориентированная база данных *MongoDB*. Также в ходе выполнения работы были изучены методы разработки *Java*-приложений с использованием *Spring Beans*, что позволило уменьшить связность между слоями приложения. Основываясь на выше перечисленном, была разработана система распознавания лиц, обладающая широким возможностями по встраиванию в существующие системы безопасности.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Метод Виолы-Джонса (Viola-Jones) как основа для распознавания лиц [Электронный ресурс] // Хабрахабр [Электронный ресурс] : [сайт]. URL: <https://habrahabr.ru/post/133826/> (дата обращения: 30.05.2017). Загл. с экрана. Яз.: рус.
- 2 Robert E. Shapire. A Brief Introduction to Boosting. IJCAI'99 Proceedings of the 16th international joint conference on Artificial intelligence, Volume 2, August 1999, 1401-1406.
- 3 Peter N. Belhumeur, Joao P. Hespanha, David J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. Journal IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 19 Issue 7, July 1997, 711-720. doi: 10.1109/34.598228
- 4 Spring Boot Reference Guide [Электронный ресурс] : [сайт]. URL: <http://docs.spring.io/spring-boot/docs/current-SNAPSHOT/reference/html/> (дата обращения: 01.05.2017). Загл. с экрана. Яз.: англ.
- 5 Spring Security Reference [Электронный ресурс] : [сайт]. URL: <https://docs.spring.io/spring-security/site/docs/current/reference/htmlsingle/> (дата обращения: 01.05.2017). Загл. с экрана. Яз.: англ.
- 6 Spring Data MongoDB – Reference Documentation [Электронный ресурс] : [сайт]. URL: <http://docs.spring.io/spring-data/data-document/docs/current/reference/html/> (дата обращения: 05.05.2017). Загл. с экрана. Яз.: англ.