# Аннотация

В данной работе были изучены существующие системы контроля рабочего времени сотрудника, основанные на технологиях компьютерного зрения.

Целью данной работы является анализ существующих системы контроля рабочего времени сотрудника, выбор среды разработки и создание системы, позволяющей сократить время, расходуемое руководителями на контроль рабочего времени сотрудников, а также сократить время, расходуемое на поиск уязвимостей в системах безопасности предприятий.

В ходе работы решены следующие задачи:

· Исследованы существующие системы контроля рабочего времени сотрудника.

· Описаны требования к решению задачи

· Разработана структура программного решения

· Разработано программное решение к поставленной задаче

Данная работа содержит:

· 36 страниц

· 6 изображений

Оглавление

[Аннотация 1](#_Toc484956812)

[Введение 3](#_Toc484956813)

[1.Анализ и сравнение существующих систем с разрабатываемой. 5](#_Toc484956814)

[1.1 Системы контроля времени 5](#_Toc484956815)

[1.1.1. БОСС Контроль 6](#_Toc484956816)

[1.1.2 PERCo 7](#_Toc484956817)

[1.2 Компьютерное зрение 9](#_Toc484956818)

[1.2.1. Беспилотные автомобили 9](#_Toc484956819)

[1.2.2. Персонализация 11](#_Toc484956820)

[**1.2.3. Интерфейсы** 12](#_Toc484956821)

[2.Структура и функциональные возможности системы 14](#_Toc484956822)

[2.1. Описание функциональности готовой системы 14](#_Toc484956823)

[2.1.1. Автоматический контроль на основе технологий компьютерного зрения. 14](#_Toc484956824)

[2.1.2. Потоки видеоинформации с одноплатных компьютеров 15](#_Toc484956825)

[2.1.3. Создание отчетов 15](#_Toc484956826)

[2.1.4. Возможности настройки элементов системы 16](#_Toc484956827)

[2.1.5. Ввод и удаление данных о сотруднике 16](#_Toc484956828)

[2.3.1 Описание одноплатного компьютера 17](#_Toc484956829)

[3.Реализация системы 18](#_Toc484956830)

[3.1.Выбор библиотеки 18](#_Toc484956831)

[3.1.1 OpenCV 18](#_Toc484956832)

[3.1.2.AForge.NET 20](#_Toc484956833)

[3.1.3 VXL 21](#_Toc484956834)

[3.1.4 LTI 22](#_Toc484956835)

[3.1.5. Вывод по выбору библиотеки 22](#_Toc484956836)

[3.2. Выбор языка программирования 23](#_Toc484956837)

[3.2.1. Java 23](#_Toc484956838)

[3.2.2. Python 24](#_Toc484956839)

[3.2.3. С++ 24](#_Toc484956840)

[3.2.4. Выводы 25](#_Toc484956841)

[3.3 Выбор алгоритма распознавания лиц 25](#_Toc484956842)

[3.3.1 Алгоритм EigenFaces [2] 25](#_Toc484956843)

[3.3.2 Алгоритм FisherFaces [3] 26](#_Toc484956844)

[3.3.3 Описание LPB Histogram 26](#_Toc484956845)

[3.3.4 Вывод 30](#_Toc484956846)

[Заключение 30](#_Toc484956847)

[Библиографический список 31](#_Toc484956848)

[Приложение 31](#_Toc484956849)

# Введение

Распознавание лиц – это простая задача для человека. Эксперименты в статье «Newborns' face recognition: Role of inner and outer facial features. Child development» [1] показали, что даже дети одного- трех дней от роду способны различать знакомые лица. Так почему это может вызывать трудности для компьютера? На самом деле мы мало знаем о том, как человек распознает лица. Используются ли внутренние признаки (глаза, нос, рот) или внешние (форма головы, линии роста волос) для успешного распознавания лиц? Как анализировать изображение и как мозг их кодирует? Как было показано Дэвидом Хьюбелом и Торстеном Визелом наш мозг имеет специализированные нервные клетки, реагирующие на конкретные местные особенности сцены, такие как линии, края, углы и движения. Так как мы не видим мир, как разрозненные куски, наша зрительная кора должна как-то объединить различные источники информации в шаблоны, которые можно использовать. С помощью автоматического распознавания лиц возможно извлекать содержательные характеристик изображения и составлять из них полезную интерпритацию, которую возможно как-либо классификацировать.

На сегодняшний день технологии контроля рабочего времени сотрудников в основном работают либо на основе пропускных пунктов на предприятии, либо на основе определенного программного обеспечения, установленного на компьютерах сотрудников. Первые системы не могут эффективно отображать человеческую деятельность внутри самого предприятия. Вторые же системы хоть и могут более точно отобразить чем занят человек во время работы, имеют жесткую привязку к компьютеру, за которым должен работать человек.

Но для предприятий, в которых люди работают не за компьютером, подобная система не выгодна и не информативна. Так что для таких предприятий является актуальным контроль рабочего времени сотрудников. В основном он осуществляется людьми с помощью проверок. Автоматизированная система контроля, основанная на технологиях компьютерного зрения, сможет сэкономить время руководителей, а также предоставить рабочему возможность доказать свое присутствие на рабочем месте в определенные промежутки времени.

Целью данной работы является создание системы, которая позволит учитывать рабочее время сотрудников точнее, чем у известных систем по средством анализа, с помощью компьютерного зрения, видео с камер, расположенных у рабочих мест сотрудников. Это сократит время на получение данных о сотрудниках, а также позволит выявлять различные нарушения, что поспособствует уменьшению расходов.

# 1.Анализ и сравнение существующих систем с разрабатываемой.

Т. к. в системах контроля времени ранее не использовалось компьютерное зрение, было решено проанализировать системы контроля времени и системы, использующие компьютерное зрение по отдельности.

## 1.1 Системы контроля времени

Как было указано в введение, существует два типа систем для учета рабочего времени сотрудника. Первый тип, это системы, которые считают время по проходу используя, например, бесконтактные карты. Представителей такой системы множество, для примера были выбраны компании БОСС Контроль и PERCo. Разрабатываемая система также относится к первому типу.

Второй тип систем, это системы устанавливающиеся на компьютеры сотрудников. Такие системы изначально нам не подходят.

### 1.1.1. БОСС Контроль

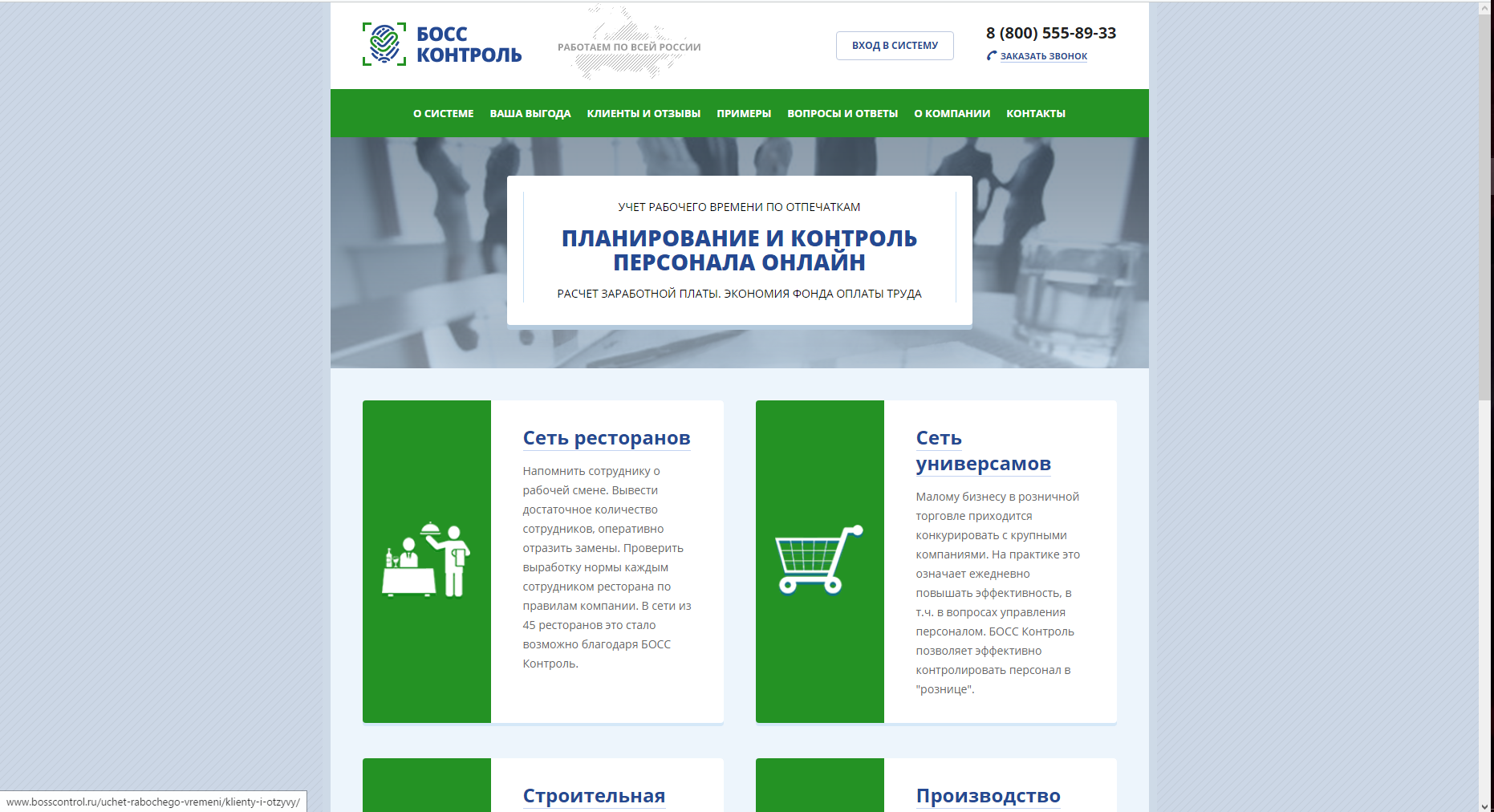


Рис. 1.1. Сайт компании БОСС Контроль

Информация с сайта[2]:

«БОСС Контроль — первая система учёта рабочего времени с центром обработки данных в "облаке". Система состоит из двух частей - терминалов БОСС Контроль и сервиса обработки данных. Терминалы учета рабочего времени сканируют отпечатки пальцев сотрудников и фиксируют время прихода/ухода с работы. Сервис БОСС Контроль получает данные с терминалов и формирует необходимую отчётность для руководства, бухгалтерии и кадров. Решение работает сразу «из коробки». Для начала работы требуется только повесить терминал БОСС Контроль на стену и включить в розетку. Все управление происходит из веб-браузера с любого компьютера или планшета с подключением к интернет.

* Сотрудник приходит на работу, подходит к терминалу БОСС Контроль и прикладывает палец
* БОСС Контроль распознаёт сотрудника, записывает точное время прихода и делает фотографию
* Через встроенный канал сотовой связи информация из терминала направляется в центр обработки данных БОСС Контроль
* Данные оперативно обрабатываются, а руководство, отдел кадров и бухгалтерия получают доступ к информации в режиме online»

Разрабатываемая система лучше, т. к. позволяет проводить контроль прямо на рабочем месте, а не только при входе.

### 1.1.2 PERCo



Риc. 1.2. Сайт компании PERCo

Информация с сайта[3]:

«Контроллер регистрации PERCo-CR01 предназначен для организации учета рабочего времени и контроля трудовой дисциплины.  
LICON применяется в тех случаях, когда целесообразно контролировать время прихода/ухода сотрудников не на проходных, где установлены турникеты, а в специально отведенных для этого местах. Это может быть учреждение без турникета на входе, либо, наоборот, большое предприятие, когда место работы сотрудника территориально удалено от проходной. В этом случае терминалы LICON устанавливаются в цехах и других помещениях, где расположены рабочие места.  
Контроллер регистрации CR01 работает с бесконтактными картами доступа формата EM-Marin и HID (HID ProxCard II, ISO prox II; EM-Marin IL-05ELR, ЕМ4100).

LICON может работать как самостоятельная система или как часть комплексной системы безопасности PERCo-S-20.

Интерфейс связи с ПК и другими контроллерами системы – Ethernet (обеспечивается поддержка стека протоколов TCP/IP (ARP, IP, ICMP, TCP, UDP, DHCP)).

Контроллер регистрации PERCo-CR01 LICON имеет два встроенных считывателя бесконтактных карт доступа формата EMM/HID для регистрации прихода/ухода, места для предъявления карт помечены пиктограммами.

При поднесении карт доступа к встроенным считывателям в энергонезависимой памяти контроллера запоминаются события «Вход» или «Выход». При наличии связи с сервером системы безопасности PERCo-S-20 события передаются в базу данных и далее, наравне с событиями от других контроллеров системы, используются для построения требуемых отчетов.

Контроллер CR01 имеет ЖКИ, на котором отображается:

* информация о том, какой из считывателей регистрирует «вход», а какой «выход» (соответствие левого и правого считывателя событиям Вход и Выход задается программно)
* фамилия владельца карты (при наличии связи с сервером системы)
* информация о нарушениях – повторное предъявление карты на вход или выход, нарушение времени, зональности (нарушения сопровождаются звуковой индикацией)»

Преимуществом разрабатываемой системы по сравнению с данной является возможность записи происходящего в помещении, отсутствие требования дополнительных предметов для идентификации.

## 1.2 Компьютерное зрение

Компьютерное зрение так же востребовано, вот несколько примеров использования.

### 1.2.1. Беспилотные автомобили

Самые крупные стенды с компьютерным зрением принадлежат автомобильной промышленности. В конце концов, технологии беспилотных и полуавтономных автомобилей работают, во многом, благодаря компьютерному зрению.

Продукты компании NVIDIA, которая уже сделала большие шаги в области глубинного обучения, используются во многих беспилотных автомобилях. Например, суперкомпьютер NVIDIA Drive PX 2[4] уже служит базовой платформой для беспилотников Tesla, Volvo, Audi, BMW и Mercedes-Benz.

Технология искусственного восприятия DriveNet от NVIDIA представляет собой самообучаемое компьютерное зрение, работающее на основе нейронных сетей. С ее помощью лидары, радары, камеры и ультразвуковые датчики способны распознавать[5] окружение, дорожную разметку, транспорт и многое другое.

NVIDIA и Audi планируют в 2020 году выпустить на трассу первую полностью автономную машину, которая будет использовать новый суперкомпьютер Xavier AI.

Рис 1.3.Генеральный директор NVIDIA Джен-Сан Хуанг представляет ИИ Co Pilot. Фото: NVIDIA

NVIDIA также представила искусственный интеллект Co Pilot, который способен распознавать лица, движения губ, направление взгляда и язык — из-за чего водить автомобиль станет удобнее. Так, например, благодаря распознаванию движения губ компьютер будет лучше понимать произносимые слова, а способность улавливать направление взгляда (анализ глаз, лица и положения головы) поможет не только определить, не заснул ли водитель, но и спасет в трудных ситуациях, когда человек не замечает опасность — например, приближающегося сзади по центру дороги мотоциклиста. Что касается считывания движения губ, то сейчас сети, использующие глубинное обучение, способны распознавать[6] речь с точностью до 95%, в то время как человек распознает ее с точностью в 3%. Сейчас эта невероятная способность используется для улучшения распознавания речи в автомобиле, а именно в шумных ситуациях.

### 1.2.2. Персонализация

В будущем благодаря технологии распознавания лиц индивидуальные настройки автомобиля станут значительно лучше.

Рис. 1.4. Chrysler

Так, в концепт-каре Chrysler Portal[7] есть специальные камеры, расположенные за рулем. С помощью компьютерного зрения автомобиль может мгновенно определять личность водителя еще до того, как тот сядет в машину, и загружать его любимые музыкальные композиции, настроить кресло в нужное положение, отрегулировать температуру и так далее. Машина умеет распознавать не только водителя, но и пассажиров, и точно так же автоматически регулировать сиденья и температуру и даже раскрывать шумоподавляющие «коконы», в которых можно послушать любимую музыку. Такие способности не только понравятся владельцам автомобилей, но и смогут сильно повлиять на райдшеринговые сервисы вроде Uber и Lyft.

### 1.2.3. Интерфейсы

Технологии отслеживания движения глаз с помощью компьютерного зрения используется не только в игровых ноутбуках, но и в обычных, и корпоративных компьютерах, для того чтобы ими могли управлять люди, которые не могут воспользоваться руками. [Tobii Dynavox PCEye Mini](https://www.tobiidynavox.com/en-US/?MarketPopupClicked=true" \t "_blank) представляет собой устройство размером с шариковую ручку, которое станет идеальным и незаметным аксессуаром для планшетов и ноутбуков. Также эта технология отслеживания движения глаз используется в новых игровых и обычных ноутбуках Asus и смартфонах Huawei.

Тем временем продолжает развиваться жестовое управление (технология компьютерного зрения, которое может распознавать особые движения руками). Теперь оно будет использоваться в будущих автомобилях BMW и Volkswagen.

Рисунок 5.BMW

Новый интерфейс [HoloActive Touch](http://www.bmwblog.com/2016/12/15/bmw-introduce-holoactive-touch-2017-ces/" \t "_blank) позволяет пользователям управлять виртуальными 3D-экранами и нажимать кнопки в пространстве. Можно сказать, что он представляет собой простую версию самого настоящего голографического [интерфейса Железного человека](https://vimeo.com/59374379) (он даже точно так же реагирует легкой вибрацией на нажатие элементов). Благодаря таким технологиям, как [ManoMotion](http://manomotion.com/" \t "_blank), можно будет легко добавить жестовое управление практически в любое устройство. Причем для получения контроля над виртуальным 3D-объектом с помощью жестов ManoMotion использует обычную 2D-камеру, так что вам не понадобится никакое дополнительное оборудование.

Устройство [eyeSight’s Singlecue Gen 2](https://singlecue.com/" \t "_blank) использует компьютерное зрение (распознавание жестов, анализ лица, определение действий) и позволяет управлять с помощью жестов телевизором, «умной» системой освещения и холодильниками.

Краудфандинговый проект [Hayo](http://hayo.io/" \t "_blank), пожалуй, является самым интересным новым интерфейсом. Эта технология позволяет создавать виртуальные средства управления по всему дому — просто подняв или опустив руку, вы можете увеличить или уменьшить громкость музыки, или же включить свет на кухне, взмахнув рукой над столешницей. Все это работает благодаря цилиндрическому устройству, использующему компьютерное зрение, а также встроенную камеру и датчики 3D, инфракрасного излучения и движения.

## Вывод

Можно утверждать, что системы с компьютерным зрением востребованы, и хорошо справляются со своими задачами, а системам контроля времени не хватает гибкости «для работы с заводами»?.

# 2.Структура и функциональные возможности системы

## 2.1. Описание функциональности готовой системы

Следует определить требования к функциональности результирующей системы.

Исходя из задач, описанных в п. 1.2., список возможностей, которые должна утилизировать система, выглядит следующим образом:

1. Автоматический контроль рабочего времени сотрудников на основе технологий компьютерного зрения. Поверхностная идентификация сотрудников, с целью выявления несанкционированного доступа на предприятие и рабочее место сотрудника.

2. Предоставление возможности просмотра потоков видеоинформации с рабочих мест сотрудников, для удобства службы безопасности предприятия.

3. Создание отчетов о рабочем времени сотрудников, а также подготовка зарплатных ведомостей исходя из времени, проведенного сотрудниками на работе, при почасовой оплате

4. Возможность настройки элементов системы удаленно. Мониторинг системы одноплатных компьютеров, а также сообщение о необходимости ремонта той или иной части системы.

5. Ввод данных о сотруднике, а также снятие его изображения, для работы автоматическим систем контроля.

### 2.1.1. Автоматический контроль на основе технологий компьютерного зрения.

Данный элемент системы будет реализован в другой работе. Тем не менее в данной работе реализуется система сопровождающая этот элемент.

Для сопровождения данного функционала предусмотрена работы база данных сотрудников на базе сервера, также отвечающего за базу данных рабочего времени, создания отчетов и мониторинга всей системы в целом.

Сам по себе автоматический контроль, является основной функцией системы. С помощью автоматизации контроля времени, будет экономиться время руководящего состава, а также для простых рабочих будет предоставлена возможность апеллировать к претензиям об их отсутствие на рабочем месте с помощью этой системы.

Также должны быть реализованы системы оповещения службы безопасности, если сотрудника долго нет на месте, или на его месте находиться несанкционированный человек.

### 2.1.2. Потоки видеоинформации с одноплатных компьютеров

Помимо функций автоматического компьютерного зрения будут реализованы и обычные функции видеокамеры на местах сотрудников. При запросе с клиентского приложения службы безопасности, сервер будет отправлять запрос на одноплатный компьютер системы контроля доступа, после чего поток видео будет перенаправлен клиенту. Это будет реализовано, для упрощения поиска несанкционированного доступа службе безопасности, после того, как система получит сигнал о его обнаружение.

Несмотря на полезность этого функционала с точки зрения безопасности, наличие подобных функций в программном обеспечение может повлечь за собой изменение контрактов сотрудников, либо отказ от использования подобного ПО, поэтому данный функционал может быть выпущен в качестве дополнительного, но не как основной.

### 2.1.3. Создание отчетов

Отчеты будут создаваться исходя из данных в базе времени работы сотрудников. Данные в этой базе будут храниться на основе реляционных таблиц. Запросы к базе будут генерироваться автоматически исходя из клиентского интерфейса, после чего полученные данные будут представлены пользователю в виде понятного графика.

Все эти действия должен совершаться на клиентской стороне. Сервер будет нужен только для исполнения запроса к базе данных и собственно хранения базы данных.

### 2.1.4. Возможности настройки элементов системы

Поскольку данная система состоит из множества элементов, в том числе и множество одноплатных компьютеров без интерфейсов для работы с ними, поэтому настройка этих одноплатных компьютеров должна выполняться посредством клиент-серверного взаимодействия. Поэтому каждый элемент системы контроля доступа должен помимо отправки отчетов на сервер, слушать канал, на случай появления обновлений.

Также должен быть реализован сам по себе клиентский интерфейс для настройки этих данных.

### 2.1.5. Ввод и удаление данных о сотруднике

Данным функционалом должно обладать клиентское приложение службы безопасности, для того чтобы добавлять новых сотрудников и удалять старых. Со стороны сервера должны работать протоколы, которые будут добавлять сотрудника в базу данных, а также запускать обновление информации на одноплатных компьютерах.

Также должен обладать возможностью изменять записи сотрудников, а также удалять их.

**/\***

На основании анализа, сделанного в введение и в 1-м разделе, можно сформулировать основные требования к разрабатываемой системе:

* + Не требует наличия компьютера на рабочем месте.
  + Высокая точность распознавания
  + ,,,
  + ,,,

Система состоит из 2 частей, клиентской и серверной. В состав клиентской части входит одноплатовый компьютер Raspberry Pi с подключенными к нему камерами.

Функциональные возможности системы:

* + Автоматическая синхронизация информации о сотрудниках с БД на сервере через равные промежутки времени
  + Запись видеопотока
  + Анализирование, через определенные интервалы времени, видеопотока на наличие сотрудника
  + Отправление отчетов, а так же архива изображений, по которым создавался отчет, в конце дня на сервер
  + Возможность в любой момент обновить информацию о сотрудниках

\*/

### 2.3.1 Описание одноплатного компьютера

Оборудование состоит из:

* + Самого одноплатного компьютера Raspberry Pi
  + Блока питания
  + Камер

Программное обеспечение одноплатного компьютера Raspberry Pi состоит из:

* + Операционной системы **Raspbian**
  + Демона, записывающего видеопоток
  + Демона, анализирующего видеопоток
  + Демона, составляющего отчет
  + Добавленного в демона crond расписания синхронизации с сервером

# 3.Реализация системы

Для того, чтобы реализовать систему нужно выбрать язык программирования, алгоритм распознавания, и библиотеку компьютерного зрения. С начала выбирается библиотека, т.к. из-за того, что они реализованы на императивных языках для меня не будет проблемой, при необходимости, освоить язык, если выбранная библиотека не содержит языка которым я владею на должном уровне (C++, Python).

## 3.1.Выбор библиотеки

Существует несколько библиотек компьютерного зрения, так что для начала их надо сравнить и выбрать, какую из них использовать.

### 3.1.1 OpenCV

OpenCV (Open Source Computer Vision Library) — это библиотека компьютерного зрения, которая поставляется с открытым исходным программным кодом. Спектр возможностей данной библиотеки очень широк. В ней собрано большое количество алгоритмов для использования технологий компьютерного зрения. После подключения данной библиотеки к своему проекту пользователь получает доступ к более чем 500 функций, предназначенных для решения разнообразных задач. Помимо алгоритмов для работы с технологиями компьютерного зрения, данная библиотека применяется и для обработки изображений, содержит большое число численных алгоритмов и многое другое. Данная библиотека реализована на языках программирования C/C++. Однако поставляется также и для других языков, таких как Python, Java, Ruby, Matlab, Lua и т.д.. Библиотека может использоваться на различных операционных системах, к числу которых относятся Linux, Mac OS X, iOS, Android и др. Целью разработки данной библиотеки является повышение эффективности вычислений в приложениях реального времени. Язык C, на котором была написана библиотека, является оптимизированным. Библиотека OpenCV способна использовать многоядерные процессоры. В случае, если понадобится автоматическая оптимизация на различных аппаратных платформах Intel, возможно дополнительное приобретение библиотеки IPP, с английского Integrate Performance Primitives. В состав данной библиотеки входят процедуры с низкоуровневой оптимизацией, которые могут применяться для разнообразных алгоритмических областей. В спектр возможностей OpenCV входит автоматическое применение IPP во время выполнения какой-либо программы. Главной целью библиотеки OpenCV является предоставление легкого в использовании интерфейса, который поможет облегчить использование технологий компьютерного зрения в довольно сложных приложениях. Функции, которые поддерживает библиотека, охватывают разнообразные сферы компьютерного зрения, от медицины, безопасности и до стереозрения и робототехники. Все это благодаря тому, что компьютерное зрение и машинное обучение — два неразрывно связанных понятия. Кроме того, библиотека OpenCV содержит библиотеку MLL, с английского Machine Learning Library. Даная библиотека является библиотекой общего назначения и ориентирована на распознавание статических образов и технологию кластеризации. Данная библиотека является очень эффективной для решения задач компьютерного зрения, которое как раз и является основой OpenCV. Однако для решения конкретных задач машинного обучения данная библиотека не приспособлена и является довольно обобщенной.

### 3.1.2.AForge.NET

AForge.NET является библиотекой с открытым исходным кодом, созданной на языке C#, которая предназначена для разработчиков и исследователей в области компьютерного зрения. Кроме того, в библиотеке есть функционал для разработчиков в области искусственного интеллекта. Спектр возможностей библиотеки довольно широк: обработка изображений, нейронные сети, генетические алгоритмы, нечеткая логика, машинное обучение, робототехника и многое другое. Библиотека включает несколько основных компонентов. AForge.Imaging — библиотека подпрограмм для обработки изображений и фильтров. AForge.Vision — библиотека компьютерного зрения. AForge.Video — набор библиотек для работы с видео информацией. AForge.Neuro — библиотека для выполнения разнообразных действий и операций с нейронными сетями. AForge.Genetic — библиотека подпрограмм для использования генетических алгоритмов для решения различных задач. AForge.Fuzzy — библиотека для работы с нечеткой логикой. AForge.Robotics — библиотека, обеспечивающая поддержку некоторых методов, применяемых в сфере робототехники. AForge.MachineLearning — библиотека для работы с элементами машинного обучения. AForge.NET постоянно улучшается и прогрессирует. По данной библиотеке есть большое количество примеров, демонстрирующих ее работу, а также актуальная html-документация, которая помогает начинающим разработчикам, которые хотят применять данный фреймворк в своих проектах. Кроме того, как и у библиотеки OpenCV существует сообщество, где можно задавать вопросы и делиться своими наработками по функциям и компонентам AForge.NET. Однако количество участников сообщества по данной библиотеке все-таки меньше, чем в сообществах по библиотеке OpenCV. Данный фреймворк уступает по популярности OpenCV. Еще одним недостатком является тот факт, что вся документация по библиотеке написана только на английском языке, поэтому у некоторых разработчиков могут возникнуть трудности с изучением и работой с данной библиотекой.

### 3.1.3 VXL

VXL, от английского the Vision-somethingLibraries, — это набор библиотек, написанных на языке C++, которые предназначены для научных исследований и реализации технологий компьютерного зрения. VXL была написана в ANSI/ISO C++ и предназначена для портативных платформ. Библиотека состоит из нескольких основных составляющих: VNL (числа) — численные алгоритмы и контейнеры, например, матрицы, векторы, оптимизаторы и т.д., VIL (изображения) — загрузка, сохранение и редактирование изображений во многих наиболее распространенных форматах (также существует возможность работы с очень большими изображениями), VGL (геометрия) — геометрия точек, кривых и других элементарных объектов в одно-, двух- и трехмерном пространствах, VSL (входный и выходной потоки), VBL (основные шаблоны), VUL (утилиты) — разный функционал для независимых платформ. Кроме основных библиотек входящих в состав VXL, есть также и дополнительные. Дополнительные библиотеки отвечают за такие понятия, как численные алгоритмы, обработка изображений, системы координат, геометрия камеры, стерео, манипуляции с видео потоком, восстановление структуры при движении камеры, графический дизайн, функции отслеживания, топология, классификаторы, 3d визуализация и многое другое. Особенность библиотеки заключается в том, что каждый ее компонент может использоваться отдельно, не ссылаясь на другие компоненты библиотеки. Таким образом, в приложении можно использовать только те библиотеки, которые действительно необходимы. VXL используется по всему миру. Библиотека применяется в сфере обучения и промышленности, некоторые ведущие мировые эксперты в сфере компьютерного зрения пользуются данной библиотекой. Существует документация по VXL с описанием каждого класса и функций. Однако недостатком является тот факт, что аналогично AForge.NET вся документация написана на английском языке.

### 3.1.4 LTI

LTI или LTI-lib является объектно- ориентированной библиотекой алгоритмов и структур данных, часто применяется при обработке изображений и в сфере компьютерного зрения. LTI-lib была разработана в техническом университете как часть научно-исследовательских проектов в области компьютерного зрения с технологиями робототехники, распознавания объектов, голоса и жестов. Основной целью разработки данной библиотеки является создание объектно- ориентированной библиотеки на языке C++, что во многом упрощало бы использование кода и его обслуживание, но при этом были бы обеспечены быстрые алгоритмы, которые можно было бы использовать в реальных приложениях.

### 3.1.5. Вывод по выбору библиотеки

Исходя из достоинств и недостатков вышеперечисленных библиотек, решено было выбрать OpenCV, т. к. она предоставляет наибольшее количество возможностей, при этом имея не сложный интерфейс, имеет большое количество материала для обучения (в том числе и на русском языке).

## 3.2. Выбор языка программирования

Библиотека OpenCV может поставляется для следующих языков:

* + С++
  + Java
  + Python
  + С# (если использовать «обертку» EmguCV)

### 3.2.1. Java

Java — сильно типизированный объектно-ориентированный язык программирования. Программы на Java транслируются в байт-код Java, выполняемый виртуальной машиной Java (JVM) — программой, обрабатывающей байтовый код и передающей инструкции оборудованию как

интерпретатор. Достоинством подобного способа выполнения программ является полная независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии Java является гибкая система безопасности, в рамках которой исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером), вызывают немедленное прерывание. (Википедия. 05.06.17)

Основным недостатком по сравнению с конкурентами является сильное снижение производительности программ, написанных на Java. Это объясняется тем, что программам требуется лишняя итерация, то есть транслирование в байт код, перед выполнением.

### 3.2.2. Python

Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций. (Википедия 05.06.17)

Основным недостатком можно назвать скорость исполнения программ, она значительно уступает С++.

### 3.2.3. С++

C++ — компилируемый, статически типизированный язык программирования общего назначения.

Поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование. Язык имеет богатую стандартную библиотеку, которая включает в себя распространённые контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности и другие возможности. C++ сочетает свойства как высокоуровневых, так и низкоуровневых языков. В сравнении с его предшественником — языком C, — наибольшее внимание уделено поддержке объектно-ориентированного и обобщённого программирования. (Википедия)

Сильной стороной разработки на C++ является высокая производительность и низкое потребление памяти.

Несмотря на то, что разработка на нем виде сложнее, чем на других языках, его сильные стороны намного важнее для разрабатываемой системы.

### 3.2.4. Выводы

Исходя из вышеперечисленных особенностей языков, для разработки системы был выбран язык C++, который позволяет реализовать самые эффективные решения.

## 3.3 Выбор алгоритма распознавания лиц

Проанализируем 3 алгоритма, реализованных в OpenCV.

### 3.3.1 Алгоритм EigenFaces [18]

Работа алгоритма основана на методе главных компонент. Предположим, что имеется база данных лиц, где изображения имеют размер N x N пикселей. Каждое изображение из базы данных представляют точкой в пространстве размерностью N x N. Основная идея алгоритма состоит в том, чтобы найти такой базис меньшей размерности, после проекции в который максимально сохраняется информация по осям с большой дисперсией и теряется информацию по осям с маленькой дисперсией. Это нужно для того, чтобы оставить только ту информацию, которая бы характеризовала различия лиц и удалить ненужную информацию, которая может помешать правильно идентифицировать человека.

Процедура идентификации выполняется в новом базисе с использованием Евклидовой метрики.

Основными недостатками алгоритма EigenFaces является отсутствие устойчивости к изменению условий освещенности и отсутствие инвариантности к аффинным преобразованиям.

### 3.3.2 Алгоритм FisherFaces [19]

Алгоритм предполагает наличие множества фотографий при разных условиях освещенности у каждой персоны в базе данных. В алгоритме, как и в EigenFaces, предполагается поиск базиса, но такого, который позволил бы максимизировать дисперсию между множествами изображений лиц и одновременно минимизировать дисперсию внутри каждого множества.

За счет множества фотографий каждой персоны алгоритм получается устойчивым к изменениям условий освещенности, но сохраняет недостаток алгоритма EigenFaces в отсутствии инвариантности к аффинным преобразованиям.

### 3.3.3 Описание LPB Histogram

Eigenfaces и Fisherfaces используют целостный подход к распознаванию. В них представляются данные как вектор в многомерном пространстве изображения. Мы все понимаем,что использовать многомерное пространство плохо, так что лучше определять маломерное пространство в котором (возможно) будет полезная информация. Подход Eigenfaces максимизирует общий разброс, что может привести к проблемам, если дисперсия генерируется внешним источником, поскольку компоненты с максимальной дисперсией по всем классам необязательно полезны для классификации {1}.Таким образом, чтобы сохранить некоторую дискриминационную информацию, мы применили линейный дискриминантный анализ и оптимизировали, как описано в методе Fisherfaces. Метод Fisherfaces отлично работает ... по крайней мере, для ограниченного сценария, который мы предположили в нашей модели.

К несчастью, настоящая жизнь не идеальна. Вы просто не можете гарантировать идеальные настройки освещения в ваших изображениях или 10 разных изображений человека. Так что, если для каждого человека есть только одно изображение? Наши оценки ковариации для подпространства могут быть ужасно неправильными, так же как и распознавание. Помните, что метод Eigenfaces имел коэффициент распознавания 96% на базе AT & T Facedatabase? Сколько изображений нам действительно нужно, чтобы получить такие полезные оценки? Ниже приведены скорости распознавания Rank-1 метода Eigenfaces и Fisherfaces в базе данных AT & T Faced, которая представляет собой довольно легкую базу данных изображений:

**image**

Поэтому, чтобы получить хорошие показатели распознавания, для каждого человека вам понадобится как минимум 8 (+ - 1) изображений, а метод Fisherfaces действительно не поможет. Вышеуказанный эксперимент представляет собой 10-кратный перекрестный подтвержденный результат, выполненный с каркасом facerec по адресу: https://github.com/bytefish/facerec. В [106] есть детальный анализ обоих методов на небольших наборах учебных данных.

Поэтому некоторые исследования были сосредоточены на извлечении локальных особенностей из изображений. Идея состоит в том, чтобы не рассматривать весь образ как высокомерный вектор, а описывать только локальные особенности объекта. Функции, которые вы извлекаете таким образом, будут иметь неявное значение. Прекрасная идея! Но вы скоро заметите, что изображение, которое мы даем, не только страдает от колебаний освещенности. Подумайте о вещах, таких как масштабирование, перевод или ротация изображений - ваше местное описание должно быть по крайней мере немного устойчивым к этим вещам. Как и SIFT, методология локальных двоичных паттернов имеет свои корни в анализе 2D-текстуры. Основная идея локальных двоичных паттернов состоит в том, чтобы суммировать локальную структуру изображения, сравнивая каждый пиксель с его окрестностью.. Если интенсивность центрального пикселя больше равна его соседству, то обозначают его с 1 и 0, если нет. Вы получите двоичный номер для каждого пикселя, как и

1. Таким образом, с 8 окружающими пикселями вы получите 2 ^ 8 возможных комбинаций, называемых локальными двоичными паттернами или иногда называемыми кодами LBP. Первый оператор LBP, описанный в литературе, фактически использовал фиксированную область 3 x 3 так же, как это:

**Image**

**Детальное описание алгоритма LBPH**

Более формальное описание оператора LBP может быть дано как:

С (xc,yc) в качестве центрального пикселя с интенсивностью ic; и ip в качестве интенсивности соседнего пикселя. S - знаковая функция, определенная как:

Это описание позволяет отображать очень мелкие детали в изображениях. Фактически авторы смогли конкурировать с современными результатами для классификации текстур. Вскоре после публикации оператора было отмечено, что фиксированная область не может кодировать детали, отличающиеся по маштабу. Таким образом, оператор был расширен, чтобы использовать переменную окрестность в [3]. Идея состоит в том, чтобы выровнять произвольное число соседей по кругу с переменным радиусом, что позволяет захватывать следующие окрестности:

**image**

Для данной точки (xc,yc) положение соседа (xp,yp), p∈P может быть вычислено:

Где R - радиус окружности, а P - количество точек выборки.

Оператор является расширением исходных кодов LBP, поэтому его иногда называют Extended LBP (также называемым Circular LBP). Если координата точек на круге не соответствует координатам изображения, точка интерполируется. Существует множество умных схем интерполяции, реализация OpenCV выполняет билинейную интерполяцию:

По определению оператор LBP устойчив к преобразованиям в градации серого. Мы легко можем проверить это, посмотрев на изображение LBP искусственно модифицированного изображения (так что вы видите, как выглядит изображение LBP!):

**image**

Так что осталось сделать, как включить пространственную информацию в модель распознавания лиц. Представление, предложенное Ahonen et. Al [3] состоит в том, чтобы разделить изображение LBP на m локальных областей и извлечь гистограмму из каждого. Затем пространственно расширенный вектор признаков получается путем конкатенации локальных гистограмм (не слияния их). Эти гистограммы называются локальными бинарными гистограммами.

### 3.3.4 Вывод

Исходя из анализа алгоритмов, решено было использовать метод LBP гистограмм, т. к. он показывает лучшие, по сравнению с рассматриваемыми, результаты в условиях плохой освещенности и малого набора тестовых изображений.

# Заключение

При выполнении выпускной квалификационной работы на соискание академической степени бакалавра по направлению 09.03.01 «Информатика и вычислительная техника» на тему «Разработка клиентской части процедуры идентификации пользователей на базе системы Raspberry Pi» был проведен анализ существующих систем контроля рабочего времени и идентификации пользователей, была определенна актуальность новой системы.

Для создаваемой системы были описаны требования к функциональности и средствам реализации. На основе этих требований была построена модель выбрана библиотека (OpenCV), а также выбраны язык реализации (C++) и метод.

Разработано программное обеспечение, представляющее собой локальный сервер, с соответствующей логикой для доступа к базам данных, а также два клиентских приложения, для получения данных из базы данных и добавления новых.

# Библиографический список

1. Chiara Turati, Viola Macchi Cassia, Francesca Simion, and Irene Leo. Newborns' face recognition: Role of inner and outer facial features. Child development, 77(2):297–311, 2006.
2. БОСС Контроль//Главная URL:<http://www.bosscontrol.ru/uchet-rabochego-vremeni/> (дата обращения: 16.10.2016).
3. PERCo//Терминал учета рабочего времени PERCo-CR01 в составе системы безопасности URL:<https://www.perco.ru/products/kontroller-registratsii-cr01.php> (дата обращения: 16.10.2016).
4. NVIDIA//Self-Driving Vehicles Development Platform URL:<http://www.nvidia.com/object/drive-px.html> (дата обращения: 9.11.2016).
5. NVIDIA//NVIDIA AI Car Demonstration URL:<https://www.youtube.com/watch?v=-96BEoXJMs0> (дата обращения: 8.01.2017).
6. LipNet AI takes lip reading into the future URL:<http://www.cs.ox.ac.uk/news/1217-full.html> (дата обращения: 9.11.2016).
7. Driving Gets Personal with the Panasonic-Chrysler Portal at #CES2017 URL:<https://channel.panasonic.com/contents/19735/> (дата обращения: 1.02.2017).
8. 18Matthew Turk and Alex Pentland Eigenfaces for Recognition, 1991.
9. 19Peter N. Belhumeur, Joao P. Hespanha, and David J. Kriegman Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection, 1997.
10. **{1}**
11. Philipp Wagner. Principal Component Analysis and Linear Discriminant Analysis with GNU Octave. URL:[**http://www.bytefish.de/wiki/pca\_lda\_with\_gnu\_octave**](http://www.bytefish.de/wiki/pca_lda_with_gnu_octave) 14.03.17
12. **[1]**
13. Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. URL:<http://dx.doi.org/10.1109/TPAMI.2012.120> IEEE Trans. Pattern Anal. Mach. Intell., 34(11):2274–2282, nov 2012.
14. **[2]**
15. Motilal Agrawal, Kurt Konolige, and Morten Rufus Blas. Censure: Center surround extremas for realtime feature detection and matching. In Computer Vision–ECCV 2008, pages 102–115. Springer, 2008.
16. **[3]**
17. Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face recognition with local binary patterns. In Computer vision-eccv 2004, pages 469–481. Springer, 2004.
18. **[4]**
19. Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. Freak: Fast retina keypoint. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 510–517. Ieee, 2012.
20. **[5]**
21. Pablo F Alcantarilla, Jesús Nuevo, and Adrien Bartoli. Fast explicit diffusion for accelerated features in nonlinear scale spaces. Trans. Pattern Anal. Machine Intell, 34(7):1281–1298, 2011.
22. **[6]**
23. Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J Davison. Kaze features. In Computer Vision–ECCV 2012, pages 214–227. Springer, 2012.
24. **[7]**
25. Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Visual tracking with online multiple instance learning. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 983–990. IEEE, 2009.
26. **[8]**
27. Dana H Ballard. Generalizing the hough transform to detect arbitrary shapes. Pattern recognition, 13(2):111–122, 1981.
28. **[9]**
29. Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. Computer Vision–ECCV 2006, pages 404–417, 2006.
30. **[10]**
31. Christian Beecks, Merih Seran Uysal, and Thomas Seidl. Signature quadratic form distance. In CIVR, pages 438–445. ACM, 2010.
32. **[11]**
33. Peter N. Belhumeur, João P Hespanha, and David Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 19(7):711–720, 1997.
34. **[12]**
35. Alexandre Benoit, Alice Caplier, Barthélémy Durette, and Jeanny Hérault. Using human visual system modeling for bio-inspired low level image processing. Computer vision and Image understanding, 114(7):758–773, 2010.
36. **[13]**
37. Stan Birchfield and Carlo Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 20(4):401–406, 1998.
38. **[14]**
39. Gunilla Borgefors. Distance transformations in digital images. Computer vision, graphics, and image processing, 34(3):344–371, 1986.
40. **[15]**
41. Léon Bottou. Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT'2010, pages 177–186. Springer, 2010.
42. **[16]**
43. Jean-Yves Bouguet. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. Intel Corporation, 5, 2001.
44. **[17]**
45. Jean-Yves Bouguet. Camera calibration tool box for matlab [eb/ol], 2004.
46. **[18]**
47. GR Bradski and J Davis. Motion segmentation and pose recognition with motion history gradients. In Applications of Computer Vision, 2000, Fifth IEEE Workshop on., pages 238–244. IEEE, 2000.
48. **[19]**
49. Gary R Bradski. Computer vision face tracking for use in a perceptual user interface. 1998.
50. **[20]**
51. Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. Classification and regression trees. CRC press, 1984.
52. **[21]**
53. Matthew Brown and David G Lowe. Automatic panoramic image stitching using invariant features. International journal of computer vision, 74(1):59–73, 2007.
54. **[22]**
55. Thomas Brox, Andres Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In Computer Vision-ECCV 2004, pages 25–36. Springer, 2004.
56. **[23]**
57. Roberto Brunelli and Tomaso Poggio. Face recognition through geometrical features. In Computer Vision—ECCV'92, pages 792–800. Springer, 1992.
58. **[24]**
59. Peter J Burt and Edward H Adelson. A multiresolution spline with application to image mosaics. ACM Transactions on Graphics (TOG), 2(4):217–236, 1983.
60. **[25]**
61. Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In Computer Vision–ECCV 2010, pages 778–792. Springer, 2010.
62. **[26]**
63. John Canny. A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, (6):679–698, 1986.
64. **[27]**
65. Antonin Chambolle, Vicent Caselles, Daniel Cremers, Matteo Novaga, and Thomas Pock. An introduction to total variation for image analysis. Theoretical foundations and numerical methods for sparse recovery, 9:263–340, 2010.
66. **[28]**
67. Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST), 2(3):27, 2011.
68. **[29]**
69. Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In IEEE CVPR, 2014.
70. **[30]**
71. Dongliang Cheng, Brian Price, Scott Cohen, and Michael S Brown. Effective learning-based illuminant estimation using simple features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1000–1008, 2015.
72. **[31]**
73. Hojin Cho, Hyunjoon Lee, Henry Kang, and Seungyong Lee. Bilateral texture filtering. ACM Transactions on Graphics, 33(4):128:1–128:8, July 2014.
74. **[32]**
75. Pengyu Cong, Zhiwei Xiong, Yueyi Zhang, Shenghui Zhao, and Feng Wu. Accurate dynamic 3d sensing with fourier-assisted phase shifting. IEEE Journal of Selected Topics in Signal Processing, 9(3):396–408, 2015.
76. **[33]**
77. Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 886–893. IEEE, 2005.
78. **[34]**
79. M. Danelljan, F.S. Khan, M. Felsberg, and J. van de Weijer. Adaptive color attributes for real-time visual tracking. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 1090–1097, June 2014.
80. **[35]**
81. James W Davis and Aaron F Bobick. The representation and recognition of human movement using temporal templates. In Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on, pages 928–934. IEEE, 1997.
82. **[36]**
83. Brice Chaix De Lavarène, David Alleysson, Barthélémy Durette, and Jeanny Hérault. Efficient demosaicing through recursive filtering. In Image Processing, 2007. ICIP 2007. IEEE International Conference on, volume 2, pages II–189. IEEE, 2007.
84. **[37]**
85. Paul E Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In ACM SIGGRAPH 2008 classes, page 31. ACM, 2008.
86. **[38]**
87. Piotr Dollár and C Lawrence Zitnick. Structured forests for fast edge detection. In Computer Vision (ICCV), 2013 IEEE International Conference on, pages 1841–1848. IEEE, 2013.
88. **[39]**
89. Frédéric Drago, Karol Myszkowski, Thomas Annen, and Norishige Chiba. Adaptive logarithmic mapping for displaying high contrast scenes. In Computer Graphics Forum, volume 22, pages 419–426. Wiley Online Library, 2003.
90. **[40]**
91. Bertram Drost and Slobodan Ilic. 3d object detection and localization using multimodal point pair features. In 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on, pages 9–16. IEEE, 2012.
92. **[41]**
93. Richard O Duda, Peter E Hart, and David G Stork. Pattern classification. John Wiley & Sons, 2012.
94. **[42]**
95. Frédo Durand and Julie Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. In ACM Transactions on Graphics (TOG), volume 21, pages 257–266. ACM, 2002.
96. **[43]**
97. Georgios D Evangelidis and Emmanouil Z Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 30(10):1858–1865, 2008.
98. **[44]**
99. Zeev Farbman, Raanan Fattal, Dani Lischinski, and Richard Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. In ACM Transactions on Graphics (TOG), volume 27, page 67. ACM, 2008.
100. **[45]**
101. Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In Image Analysis, pages 363–370. Springer, 2003.
102. **[46]**
103. Sina Farsiu, Dirk Robinson, Michael Elad, and Peyman Milanfar. Fast and robust super-resolution. In Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on, volume 2, pages II–291. IEEE, 2003.
104. **[47]**
105. Raanan Fattal, Dani Lischinski, and Michael Werman. Gradient domain high dynamic range compression. In ACM Transactions on Graphics (TOG), volume 21, pages 249–256. ACM, 2002.
106. **[48]**
107. Pedro Felzenszwalb and Daniel Huttenlocher. Distance transforms of sampled functions. Technical report, Cornell University, 2004.
108. **[49]**
109. Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. volume 59, pages 167–181. Springer, 2004.
110. **[50]**
111. Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient belief propagation for early vision. International journal of computer vision, 70(1):41–54, 2006.
112. **[51]**
113. Pedro F Felzenszwalb, Ross B Girshick, and David McAllester. Cascade object detection with deformable part models. In Computer vision and pattern recognition (CVPR), 2010 IEEE conference on, pages 2241–2248. IEEE, 2010.
114. **[52]**
115. Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 32(9):1627–1645, 2010.
116. **[53]**
117. Ronald A Fisher. The use of multiple measurements in taxonomic problems. Annals of eugenics, 7(2):179–188, 1936.
118. **[54]**
119. Andrew W Fitzgibbon and Robert B Fisher. A buyer's guide to conic fitting. In Proceedings of the 6th British conference on Machine vision (Vol. 2), pages 513–522. BMVA Press, 1995.
120. **[55]**
121. Per-Erik Forssén. Maximally stable colour regions for recognition and matching. In Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, pages 1–8. IEEE, 2007.
122. **[56]**
123. Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. 1998.
124. **[57]**
125. Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 25(8):930–943, 2003.
126. **[58]**
127. Eduardo SL Gastal and Manuel M Oliveira. Domain transform for edge-aware image and video processing. In ACM Transactions on Graphics (TOG), volume 30, page 69. ACM, 2011.
128. **[59]**
129. Eduardo SL Gastal and Manuel M Oliveira. Domain transform for edge-aware image and video processing. In ACM Transactions on Graphics (TOG), volume 30, page 69. ACM, 2011.
130. **[60]**
131. Eduardo SL Gastal and Manuel M Oliveira. Adaptive manifolds for real-time high-dimensional filtering. ACM Transactions on Graphics (TOG), 31(4):33, 2012.
132. **[61]**
133. Pascal Getreuer. Malvar-he-cutler linear image demosaicking. Image Processing on Line, 2011.
134. **[62]**
135. Andrew B Godbehere, Akihiro Matsukawa, and Ken Goldberg. Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation. In American Control Conference (ACC), 2012, pages 4305–4312. IEEE, 2012.
136. **[63]**
137. Helmut Grabner, Michael Grabner, and Horst Bischof. Real-time tracking via on-line boosting. In BMVC, volume 1, page 6, 2006.
138. **[64]**
139. Matthias Grundmann, Vivek Kwatra, and Irfan Essa. Auto-directed video stabilization with robust l1 optimal camera paths. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pages 225–232. IEEE, 2011.
140. **[65]**
141. N Guil, José María Gonzalez-Linares, and Emilio L Zapata. Bidimensional shape detection using an invariant approach. Pattern Recognition, 32(6):1025–1038, 1999.
142. **[66]**
143. Richard Hartley and Andrew Zisserman. Multiple view geometry in computer vision. Cambridge university press, 2003.
144. **[67]**
145. Richard I Hartley. Theory and practice of projective rectification. International Journal of Computer Vision, 35(2):115–127, 1999.
146. **[68]**
147. Kaiming He and Jian Sun. Statistics of patch offsets for image completion. In Computer Vision–ECCV 2012, pages 16–29. Springer, 2012.
148. **[69]**
149. Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. In Computer Vision–ECCV 2010, pages 1–14. Springer, 2010.
150. **[70]**
151. David Held, Sebastian Thrun, and Silvio Savarese. Learning to track at 100 fps with deep regression networks. In European Conference Computer Vision (ECCV), 2016.
152. **[71]**
153. J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In proceedings of the European Conference on Computer Vision, 2012.
154. **[72]**
155. Kyriakos Herakleous and Charalambos Poullis. 3DUNDERWORLD-SLS: An Open-Source Structured-Light Scanning System for Rapid Geometry Acquisition. arXiv preprint arXiv:1406.6595, 2014.
156. **[73]**
157. Joel Hesch, Stergios Roumeliotis, and others. A direct least-squares (dls) method for pnp. In Computer Vision (ICCV), 2011 IEEE International Conference on, pages 383–390. IEEE, 2011.
158. **[74]**
159. S. Hinterstoisser, S. Benhimane, , V. Lepetit, and N. Navab. Simultaneous recognition and homography extraction of local patches with a simple linear classifier. 2008.
160. **[75]**
161. Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 30(2):328–341, 2008.
162. **[76]**
163. Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. In Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, pages 1–8. IEEE, 2007.
164. **[77]**
165. Ming-Kuei Hu. Visual pattern recognition by moment invariants. Information Theory, IRE Transactions on, 8(2):179–187, 1962.
166. **[78]**
167. Herault Jeanny. Vision: Images, Signals and Neural Networks-Models of Neural Processing in Visual Perception. World Scientific, 2010.
168. **[79]**
169. Pakorn KaewTraKulPong and Richard Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In Video-Based Surveillance Systems, pages 135–144. Springer, 2002.
170. **[80]**
171. Pakorn KaewTraKulPong and Richard Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In Video-Based Surveillance Systems, pages 135–144. Springer, 2002.
172. **[81]**
173. Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Forward-backward error: Automatic detection of tracking failures. In Pattern Recognition (ICPR), 2010 20th International Conference on, pages 2756–2759. IEEE, 2010.
174. **[82]**
175. Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 34(7):1409–1422, 2012.
176. **[83]**
177. Takeo Kanade. Picture processing system by computer complex and recognition of human faces. 1974.
178. **[84]**
179. Victor Klee and Michael C Laskowski. Finding the smallest triangles containing a given convex polygon. Journal of Algorithms, 6(3):359–375, 1985.
180. **[85]**
181. Till Kroeger, Radu Timofte, Dengxin Dai, and Luc Van Gool. Fast optical flow using dense inverse search. In Proceedings of the European Conference on Computer Vision (ECCV), 2016.
182. **[86]**
183. Martin Krulis, Jakub Lokoc, and Tomas Skopal. Efficient extraction of clustering-based feature signatures using GPU architectures. Multimedia Tools Appl., 75(13):8071–8103, 2016.
184. **[87]**
185. Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: image and video synthesis using graph cuts. In ACM Transactions on Graphics (ToG), volume 22, pages 277–286. ACM, 2003.
186. **[88]**
187. Kuang-Chih Lee, Jeffrey Ho, and David Kriegman. Acquiring linear subspaces for face recognition under variable lighting. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 27(5):684–698, 2005.
188. **[89]**
189. Jin Han Lee, Sehyung Lee, Guoxuan Zhang, Jongwoo Lim, Wan Kyun Chung, and Il Hong Suh. Outdoor place recognition in urban environments using straight lines. In 2014 IEEE International Conference on Robotics and Automation (ICRA), pages 5550–5557. IEEE, 2014.
190. **[90]**
191. Hai Lei, Xin-yu Chang, Fei Wang, Xiao-Tang Hu, and Xiao-Dong Hu. A novel algorithm based on histogram processing of reliability for two-dimensional phase unwrapping. Optik-International Journal for Light and Electron Optics, 126(18):1640–1644, 2015.
192. **[91]**
193. Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. International journal of computer vision, 81(2):155–166, 2009.
194. **[92]**
195. Stefan Leutenegger, Margarita Chli, and Roland Yves Siegwart. Brisk: Binary robust invariant scalable keypoints. In Computer Vision (ICCV), 2011 IEEE International Conference on, pages 2548–2555. IEEE, 2011.
196. **[93]**
197. Zhengqin Li and Jiansheng Chen. Superpixel segmentation using linear spectral clustering. June 2015.
198. **[94]**
199. Liyuan Li, Weimin Huang, Irene YH Gu, and Qi Tian. Foreground object detection from videos containing complex background. In Proceedings of the eleventh ACM international conference on Multimedia, pages 2–10. ACM, 2003.
200. **[95]**
201. Xi Li, Weiming Hu, Chunhua Shen, Zhongfei Zhang, Anthony Dick, and Anton Van Den Hengel. A survey of appearance models in visual object tracking. ACM Transactions on Intelligent Systems and Technology (TIST), 4(4):58, 2013.
202. **[96]**
203. Shengcai Liao, Xiangxin Zhu, Zhen Lei, Lun Zhang, and Stan Z Li. Learning multi-scale block local binary patterns for face recognition. In Advances in Biometrics, pages 828–837. Springer, 2007.
204. **[97]**
205. Rainer Lienhart and Jochen Maydt. An extended set of haar-like features for rapid object detection. In Image Processing. 2002. Proceedings. 2002 International Conference on, volume 1, pages I–900. IEEE, 2002.
206. **[98]**
207. Joseph J Lim, C Lawrence Zitnick, and Piotr Dollár. Sketch tokens: A learned mid-level representation for contour and object detection. In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pages 3158–3165. IEEE, 2013.
208. **[99]**
209. H. Louhichi, T. Fournel, J. M. Lavest, and H. Ben Aissia. Self-calibration of scheimpflug cameras: an easy protocol. Meas. Sci. Technol., 18(8):2616–2622, 2007.
210. **[100]**
211. Kok-Lim Low. Linear least-squares optimization for point-to-plane icp surface registration. Chapel Hill, University of North Carolina, 2004.
212. **[101]**
213. David G Lowe. Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2):91–110, 2004.
214. **[102]**
215. Cewu Lu, Li Xu, and Jiaya Jia. Contrast preserving decolorization. In Computational Photography (ICCP), 2012 IEEE International Conference on, pages 1–7. IEEE, 2012.
216. **[103]**
217. "Elmar Mair, Gregory D. Hager, Darius Burschka, Michael Suppa, and Gerhard Hirzinger". <a href=""http://www6.in.tum.de/Main/ResearchAgast"">Adaptive and generic corner detection based on the accelerated segment test". In "European Conference on Computer Vision (ECCV'10)", "September" "2010".
218. **[104]**
219. Ezio Malis, Manuel Vargas, and others. Deeper understanding of the homography decomposition for vision-based control. 2007.
220. **[105]**
221. Rafal Mantiuk, Karol Myszkowski, and Hans-Peter Seidel. A perceptual framework for contrast processing of high dynamic range images. ACM Transactions on Applied Perception (TAP), 3(3):286–308, 2006.
222. **[106]**
223. Aleix M Martínez and Avinash C Kak. Pca versus lda. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 23(2):228–233, 2001.
224. **[107]**
225. Jiri Matas, Charles Galambos, and Josef Kittler. Robust detection of lines using the progressive probabilistic hough transform. Computer Vision and Image Understanding, 78(1):119–137, 2000.
226. **[108]**
227. Yasuyuki Matsushita, Eyal Ofek, Weina Ge, Xiaoou Tang, and Heung-Yeung Shum. Full-frame video stabilization with motion inpainting. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 28(7):1150–1163, 2006.
228. **[109]**
229. Tom Mertens, Jan Kautz, and Frank Van Reeth. Exposure fusion. In Computer Graphics and Applications, 2007. PG'07. 15th Pacific Conference on, pages 382–390. IEEE, 2007.
230. **[110]**
231. Kieron Messer, Josef Kittler, James Short, Guillaume Heusch, Fabien Cardinaux, Sebastien Marcel, Yann Rodriguez, Shiguang Shan, Yu Su, Wen Gao, and others. Performance characterisation of face recognition algorithms and their sensitivity to severe illumination changes. In Advances in Biometrics, pages 1–11. Springer, 2005.
232. **[111]**
233. Fernand Meyer. Color image segmentation. In Image Processing and its Applications, 1992., International Conference on, pages 303–306. IET, 1992.
234. **[112]**
235. Laurence Meylan, David Alleysson, and Sabine Süsstrunk. Model of retinal local adaptation for the tone mapping of color filter array images. JOSA A, 24(9):2807–2816, 2007.
236. **[113]**
237. Dongbo Min, Sunghwan Choi, Jiangbo Lu, Bumsub Ham, Kwanghoon Sohn, and Minh N Do. Fast global image smoothing based on weighted least squares. Image Processing, IEEE Transactions on, 23(12):5638–5653, 2014.
238. **[114]**
239. Dennis Mitzel, Thomas Pock, Thomas Schoenemann, and Daniel Cremers. Video super resolution using duality based tv-l 1 optical flow. In Pattern Recognition, pages 432–441. Springer, 2009.
240. **[115]**
241. Sebastian Montabone and Alvaro Soto. Human detection using a mobile platform and novel features derived from a visual saliency mechanism. In Image and Vision Computing, Vol. 28 Issue 3, pages 391–402. Elsevier, 2010.
242. **[116]**
243. Alexander Mordvintsev. [Rof and tv-l1 denoising with primal-dual algorithm](http://znah.net/rof-and-tv-l1-denoising-with-primal-dual-algorithm.html).
244. **[117]**
245. Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In VISAPP (1), pages 331–340, 2009.
246. **[118]**
247. David Nistér and Henrik Stewénius. Linear time maximally stable extremal regions. In Computer Vision–ECCV 2008, pages 183–196. Springer, 2008.
248. **[119]**
249. David Nistér. An efficient solution to the five-point relative pose problem. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 26(6):756–770, 2004.
250. **[120]**
251. Mohammad Norouzi, Ali Punjani, and David J Fleet. Fast search in hamming space with multi-index hashing. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 3108–3115. IEEE, 2012.
252. **[121]**
253. Joseph O'Rourke, Alok Aggarwal, Sanjeev Maddila, and Michael Baldwin. An optimal algorithm for finding minimal enclosing triangles. Journal of Algorithms, 7(2):258–269, 1986.
254. **[122]**
255. Philippe Paillou. Detecting step edges in noisy sar images: a new linear operator. IEEE transactions on geoscience and remote sensing, 35(1):191–196, 1997.
256. **[123]**
257. Adrian Penate-Sanchez, Juan Andrade-Cetto, and Francesc Moreno-Noguer. Exhaustive linearization for robust camera pose and focal length estimation. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 35(10):2387–2400, 2013.
258. **[124]**
259. Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In ACM Transactions on Graphics (TOG), volume 22, pages 313–318. ACM, 2003.
260. **[125]**
261. Irina Perfilieva and Pavel Vlasánek. Image reconstruction by means of f-transform. Knowledge-Based Systems, 70:55–63, 2014.
262. **[126]**
263. Irina Perfilieva. Fuzzy transforms: Theory and applications. Fuzzy sets and systems, 157(8):993–1023, 2006.
264. **[127]**
265. Jan Puzicha, Thomas Hofmann, and Joachim M Buhmann. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on, pages 267–272. IEEE, 1997.
266. **[128]**
267. Sarunas J Raudys and Anil K. Jain. Small sample size effects in statistical pattern recognition: Recommendations for practitioners. IEEE Transactions on pattern analysis and machine intelligence, 13(3):252–264, 1991.
268. **[129]**
269. Erik Reinhard and Kate Devlin. Dynamic range reduction inspired by photoreceptor physiology. Visualization and Computer Graphics, IEEE Transactions on, 11(1):13–24, 2005.
270. **[130]**
271. Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, pages 1164–1172, 2015.
272. **[131]**
273. Martin Riedmiller and Heinrich Braun. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In Neural Networks, 1993., IEEE International Conference on, pages 586–591. IEEE, 1993.
274. **[132]**
275. Mark A Robertson, Sean Borman, and Robert L Stevenson. Dynamic range improvement through multiple exposures. In Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on, volume 3, pages 159–163. IEEE, 1999.
276. **[133]**
277. Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In Computer Vision–ECCV 2006, pages 430–443. Springer, 2006.
278. **[134]**
279. Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: an efficient alternative to sift or surf. In Computer Vision (ICCV), 2011 IEEE International Conference on, pages 2564–2571. IEEE, 2011.
280. **[135]**
281. Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover''s distance as a metric for image retrieval. 1998.
282. **[136]**
283. Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. International Journal of Computer Vision, 40(2):99–121, 2000.
284. **[137]**
285. Samuele Salti, Andrea Cavallaro, and Luigi Di Stefano. Adaptive appearance modeling for video tracking: survey and evaluation. Image Processing, IEEE Transactions on, 21(10):4334–4348, 2012.
286. **[138]**
287. Joaquim Salvi, Jordi Pagés, and Joan Batlle. Pattern codification strategies in structured light systems. Pattern Recognition, 37(4):827–849, April 2004.
288. **[139]**
289. Javier Sánchez Pérez, Enric Meinhardt-Llopis, and Gabriele Facciolo. Tv-l1 optical flow estimation. 2012.
290. **[140]**
291. Jianbo Shi and Carlo Tomasi. Good features to track. In Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on, pages 593–600. IEEE, 1994.
292. **[141]**
293. K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014.
294. **[142]**
295. Jack Sklansky. Finding the convex hull of a simple polygon. Pattern Recognition Letters, 1(2):79–83, 1982.
296. **[143]**
297. Gregory G Slabaugh. Computing euler angles from a rotation matrix. Retrieved on August, 6:2000, 1999.
298. **[144]**
299. Henrik Stewenius. [Calibrated fivepoint solver](http://www.vis.uky.edu/~stewe/FIVEPOINT/).
300. **[145]**
301. S.T. Strat, A. Benoit, and P. Lambert. Retina enhanced bag of words descriptors for video classification. In Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European, pages 1307–1311, Sept 2014.
302. **[146]**
303. Satoshi Suzuki and others. Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing, 30(1):32–46, 1985.
304. **[147]**
305. Richard Szeliski. Image alignment and stitching: A tutorial. Foundations and Trends textregistered in Computer Graphics and Vision, 2(1):1–104, 2006.
306. **[148]**
307. V. Lepetit T. Trzcinski, M. Christoudias and P. Fua. Boosting Binary Keypoint Descriptors. In Computer Vision and Pattern Recognition, 2013.
308. **[149]**
309. M. Christoudias T. Trzcinski and V. Lepetit. Learning Image Descriptors with Boosting. submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2013.
310. **[150]**
311. Michael Tao, Jiamin Bai, Pushmeet Kohli, and Sylvain Paris. Simpleflow: A non-iterative, sublinear optical flow algorithm. In Computer Graphics Forum, volume 31, pages 345–353. Wiley Online Library, 2012.
312. **[151]**
313. C-H Teh and Roland T. Chin. On the detection of dominant points on digital curves. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 11(8):859–872, 1989.
314. **[152]**
315. Alexandru Telea. An image inpainting technique based on the fast marching method. Journal of graphics tools, 9(1):23–34, 2004.
316. **[153]**
317. E. Tola, V. Lepetit, and P. Fua. DAISY: An Efficient Dense Descriptor Applied to Wide Baseline Stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(5):815–830, May 2010.
318. **[154]**
319. Federico Tombari and Luigi Di Stefano. Interest points via maximal self-dissimilarities. In Asian Conference on Computer Vision – ACCV 2014, 2014.
320. **[155]**
321. Hastie Trevor, Tibshirani Robert, and Friedman Jerome. The elements of statistical learning: data mining, inference and prediction. New York: Springer-Verlag, 1(8):371–406, 2001.
322. **[156]**
323. Chiara Turati, Viola Macchi Cassia, Francesca Simion, and Irene Leo. Newborns' face recognition: Role of inner and outer facial features. Child development, 77(2):297–311, 2006.
324. **[157]**
325. Matthew Turk and Alex Pentland. Eigenfaces for recognition. Journal of cognitive neuroscience, 3(1):71–86, 1991.
326. **[158]**
327. Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. International journal of computer vision, 104(2):154–171, 2013.
328. **[159]**
329. Matthew Uyttendaele, Ashley Eden, and R Skeliski. Eliminating ghosting and exposure artifacts in image mosaics. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, volume 2, pages II–509. IEEE, 2001.
330. **[160]**
331. Antoine Vacavant, Thierry Chateau, Alexis Wilhelm, and Laurent Lequièvre. A benchmark dataset for outdoor foreground/background extraction. In Computer Vision-ACCV 2012 Workshops, pages 291–300. Springer, 2013.
332. **[161]**
333. Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin de Capitani, and Luc Van Gool. Seeds: Superpixels extracted via energy-driven sampling. In Computer Vision–ECCV 2012, pages 13–26. Springer, 2012.
334. **[162]**
335. Lieven Vandenberghe. [Qr factorization](http://www.seas.ucla.edu/~vandenbe/133A/lectures/qr.pdf).
336. **[163]**
337. Paul Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, volume 1, pages I–511. IEEE, 2001.
338. **[164]**
339. Paul Viola and Michael J. Jones. Robust real-time face detection. International Journal of Computer Vision, 57(2):137–154, 2004.
340. **[165]**
341. R Grompone Von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall. Lsd: A fast line segment detector with a false detection control. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(4):722–732, 2010.
342. **[166]**
343. Rafael Grompone von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall. Lsd: a line segment detector. 2012.
344. **[167]**
345. Bin Wang and Piotr Dudek. A fast self-tuning background subtraction algorithm. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on, pages 401–404. IEEE, 2014.
346. **[168]**
347. Shenlong Wang, Sean Ryan Fanello, Christoph Rhemann, Shahram Izadi, and Pushmeet Kohli. The global patch collider. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.
348. **[169]**
349. Greg Ward. Fast, robust image registration for compositing high dynamic range photographs from hand-held exposures. Journal of graphics tools, 8(2):17–30, 2003.
350. **[170]**
351. Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In Computer Vision (ICCV), 2013 IEEE International Conference on, pages 1385–1392. IEEE, 2013.
352. **[171]**
353. Greg Welch and Gary Bishop. An introduction to the kalman filter, 1995.
354. **[172]**
355. Laurenz Wiskott, J-M Fellous, N Kuiger, and Christoph Von Der Malsburg. Face recognition by elastic bunch graph matching. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 19(7):775–779, 1997.
356. **[173]**
357. Paul Wohlhart and Vincent Lepetit. Learning descriptors for object recognition and 3d pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015.
358. **[174]**
359. Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pages 2411–2418. IEEE, 2013.
360. **[175]**
361. Jonas Wulff and Michael J. Black. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015, June 2015.
362. **[176]**
363. Wei Xu and Jane Mulligan. Performance evaluation of color correction approaches for automatic multi-view image and video stitching. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 263–270. IEEE, 2010.
364. **[177]**
365. Li Xu, Cewu Lu, Yi Xu, and Jiaya Jia. Image smoothing via l 0 gradient minimization. In ACM Transactions on Graphics (TOG), volume 30, page 174. ACM, 2011.
366. **[178]**
367. Qingxiong Yang, Liang Wang, and Narendra Ahuja. A constant-space belief propagation algorithm for stereo matching. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 1458–1465. IEEE, 2010.
368. **[179]**
369. HK Yuen, John Princen, John Illingworth, and Josef Kittler. Comparative study of hough transform methods for circle finding. Image and Vision Computing, 8(1):71–77, 1990.
370. **[180]**
371. Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In Pattern Recognition, pages 214–223. Springer, 2007.
372. **[181]**
373. Lilian Zhang and Reinhard Koch. An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency. Journal of Visual Communication and Image Representation, 24(7):794–805, 2013.
374. **[182]**
375. Qi Zhang, Xiaoyong Shen, Li Xu, and Jiaya Jia. Rolling guidance filter. In Computer Vision–ECCV 2014, pages 815–830. Springer, 2014.
376. **[183]**
377. Qi Zhang, Li Xu, and Jiaya Jia. 100+ times faster weighted median filter (wmf). In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 2830–2837. IEEE, 2014.
378. **[184]**
379. Zhengyou Zhang. A flexible new technique for camera calibration. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22(11):1330–1334, 2000.
380. **[185]**
381. Wenyi Zhao, Rama Chellappa, P Jonathon Phillips, and Azriel Rosenfeld. Face recognition: A literature survey. Acm Computing Surveys (CSUR), 35(4):399–458, 2003.
382. **[186]**
383. Eric Christiansen David Kriegman Ziegler, Andrew and Serge J. Belongie. Locally uniform comparison image descriptor.
384. **[187]**
385. Timo Zinßer, Jochen Schmidt, and Heinrich Niemann. A refined icp algorithm for robust 3-d correspondence estimation. In Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on, volume 2, pages II–695. IEEE, 2003.
386. **[188]**
387. Zoran Zivkovic and Ferdinand van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern recognition letters, 27(7):773–780, 2006.
388. **[189]**
389. Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, volume 2, pages 28–31. IEEE, 2004.

# Приложение

Модуль для анализа изображений и распознавания сотрудников.

#include "opencv2/core.hpp"

#include "opencv2/face.hpp"

#include "opencv2/highgui.hpp"

#include <iostream>

#include <fstream>

#include <sstream>

using namespace cv;

using namespace cv::face;

using namespace std;

static void read\_csv(const string& filename, vector<Mat>& images, vector<int>& labels, char separator = ';') {

std::ifstream file(filename.c\_str(), ifstream::in);

if (!file) {

string error\_message = "No valid input file was given, please check the given filename.";

CV\_Error(Error::StsBadArg, error\_message);

}

string line, path, classlabel;

while (getline(file, line)) {

stringstream liness(line);

getline(liness, path, separator);

getline(liness, classlabel);

if(!path.empty() && !classlabel.empty()) {

images.push\_back(imread(path, 0));

labels.push\_back(atoi(classlabel.c\_str()));

}

}

}

int main(int argc, const char \*argv[]) {

if (argc != 2) {

cout << "usage: " << argv[0] << " <csv.ext>" << endl;

exit(1);

}

// Get the path to your CSV.

string fn\_csv = string(argv[1]);

// These vectors hold the images and corresponding labels.

vector<Mat> images;

vector<int> labels;

// Read in the data. This can fail if no valid

// input filename is given.

try {

read\_csv(fn\_csv, images, labels);

} catch (cv::Exception& e) {

cerr << "Error opening file \"" << fn\_csv << "\". Reason: " << e.msg << endl;

// nothing more we can do

exit(1);

}

if(images.size() <= 1) {

string error\_message = "This demo needs at least 2 images to work. Please add more images to your data set!";

CV\_Error(Error::StsError, error\_message);

}

Ptr<LBPHFaceRecognizer> model = createLBPHFaceRecognizer();

model->train(images, labels);

string buf;

int testLabel;

bool next=true;

int predictedLabel;

int threshold;

while(next)

{cout << "enter path to test image" << endl;

cin >> buf;

cout << "enter who is this(int)" << endl;

cin >> testLabel;

cout << "enter threshold" << endl;

cin >> threshold;

Mat testSample = imread(buf, CV\_LOAD\_IMAGE\_GRAYSCALE);

model->setThreshold(threshold);

predictedLabel = model->predict(testSample);

string result\_message = format("Predicted class = %d / Actual class = %d.", predictedLabel, testLabel);

cout << result\_message << endl;

cout << "Model Information:" << endl;

string model\_info = format("\tLBPH(radius=%i, neighbors=%i, grid\_x=%i, grid\_y=%i, threshold=%.2f)",

model->getRadius(),

model->getNeighbors(),

model->getGridX(),

model->getGridY(),

model->getThreshold());

cout << model\_info << endl;

testSample.release();

buf="";

cout << "Another test?" << endl;

cin >> buf;

if (buf=="no")

next=false;

}

return 0;

}