

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ
КАЗАХСТАН

Казахский национальный исследовательский технический университет
имени К.И.Сатпаева

Институт Кибернетики и информационных технологий

УДК 004.932.72


На правах рукописи

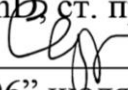
Мұса Ақбота Жанатқызы

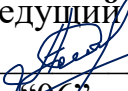
МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ


На соискание академической степени магистра

Название диссертации	Прикладное глубокое обучение и Технология Блокчейн
Направление подготовки	6М070500 – «Математическое и компьютерное моделирование»

Научный руководитель
Профессор, д-р экон. наук
 А.А. Алдашев
“06” июля 2020 г.

Рецензент
PhD ст. преп. кафедр. ИС и ВТ
 Д.С. Сабитова
“06” июля 2020 г.

Нормоконтроль
Ведущий инженер
 М.С. Амирбекова
“06” июля 2020 г.

ДОПУЩЕН К ЗАЩИТЕ
Директор НОЦ МиК
 Н.С. Даирбеков
“06” июля 2020 г.

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ
КАЗАХСТАН

Казахский национальный исследовательский технический университет
имени К.И.Сатпаева

Институт Кибернетики и информационных технологий
Научно-образовательный центр математики и кибернетики

6M070500 – «Математическое и компьютерное моделирование»

УТВЕРЖДАЮ

Директор НОЦ МиК

 Н.С. Даирбеков

“06” июля 2020 г.

ЗАДАНИЕ

на выполнение магистерской диссертации

Магистранту Мұса Ақбота Жанатқызы

Тема: Прикладное глубокое обучение и Технология Блокчейн

Утверждена приказом Ректора Университета № 452-М от 06.12.2019 г.

Срок сдачи законченной диссертации “8” июля 2020 г

Исходные данные к магистерской диссертации: Разработка методов распознавания и идентификации человека по изображению его лица в реальном масштабе времени.

Перечень подлежащих разработке в магистерской диссертации вопросов:

- а) рассмотреть различные методы распознавания человека по изображению лица
- б) сравнение алгоритмов распознавания лиц
- в) Разработка алгоритмов определения угла поворота лица, представленного на изображении, в глубину (ракурс лица), отличающиеся низкими затратами на их численную реализацию и допустимой точностью
- г) программное обеспечение для обрабатывания видеопоток со множества видеокамер
- д) приложения

Рекомендуемая основная литература:


- 1 Francis Galton, "Personal identification and description," In Nature, pp. 173-177, June 21, 1888.
- 2 W. Zaho, "Robust image-based 3D face recognition," Ph.D. Thesis, Maryland University, 1999.
- 3 R. Chellappa, C.L. Wilson and C. Sirohey, "Human and machine recognition of faces: A survey," Proc. IEEE, vol. 83, no. 5, pp. 705- 740, may 1995.
- 4 T. Fromherz, P. Stucki, M. Bichsel, "A survey of face recognition," MML Technical Report, No 97.01, Dept. of Computer Science, University of Zurich, Zurich, 1997.
- 5 T. Riklin-Raviv and A. Shashua, "The Quotient image: Class based recognition and synthesis under varying illumination conditions," In CVPR, P. II: pp. 566-571,1999.





ГРАФИК
подготовки магистерской диссертации

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Сбор материала по основным понятиям и определениям	2.07.2019 г. – 16.12.2019 г.	
Методы распознавания и идентификации человека по изображению его лица в реальном масштабе времени.	22.07.2019 г. – 1.03.2020 г	
Серверное приложение для выполнения модели ML для идентификации лица, полученной с камеры.	16.06.2020 г	
Веб-приложение для отслеживания всех посетителей, отслеживания и отправки оповещений для подозрительных людей.	29.06.2020 г	


Подписи

консультантов и нормоконтролера на законченную магистерскую диссертацию с указанием относящихся к ним разделов диссертации


Наименования разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Сбор материала по основным понятиям и определениям	Алдашев А.А., профессор, д-р экон. наук	6.07.2020 г	

Методы распознавания и идентификации человека по изображению его лица в реальном масштабе времени.	Алдашев А.А., профессор, д-р экон. наук	6.07.2020 г	
Серверное приложение для выполнения модели ML для идентификации лица, полученной с камеры.	Алдашев А.А., профессор, д-р экон. наук	6.07.2020 г	
Веб-приложение для отслеживания всех посетителей, отслеживания и отправки оповещений для подозрительных людей.	Алдашев А.А., профессор, д-р экон. наук	6.07.2020 г	
Нормоконтроль	М.С.Амирбекова, ведущий инженер	6.07.2020 г	

Научный руководитель

 Алдашев А.А.

Задание принял к исполнению магистрант

 Мұса А.Ж.

Дата “06” июля 2020 г.

АНДАТПА

Магистрлік диссертацияның негізгі мақсаты - терең білім негізінде бірнеше түрлі жағдайларда, мысалы, беттің төменгі және жоғарғы окклюзиясы, сәйкессіздік, бас позициясының әр түрлі бұрыштары, жарықтың өзгеруі, бет әлпетінің локализациясы сияқты терең білім негізінде терең білім беру негізінде зерттеу.

Адамның бейнесін алу үшін біз бұл жұмыста CNN негізіндегі терең оқыту модельдерін және компьютерлік көру алгоритмдерінің OpenCv кітапханасын қолдандық. Бұл модельдер терең оқыту моделінің әр түрлі сәйкессіздіктерге төзімді екендігін және қашықтық аралықты локализациялау қателіктеріне шыдай алатындығын көрсетеді.

АННОТАЦИЯ

Основная цель данной магистерской диссертации - это изучение представление лица на основе глубокого обучения при нескольких различных условиях, таких как нижняя и верхняя окклюзии лица, рассогласование, различные углы позы головы, изменение освещенности, дефектная локализация черт лица с использованием подходов глубокого обучения.

Для извлечения представления лица в данной работе были использованы модели глубокого обучения на основе CNN и библиотека алгоритмов компьютерного зрения OpenCv. Поскольку эти модели показывают, что модель глубокого обучения устойчива к различным типам рассогласования и может терпеть ошибку локализации межокулярного расстояния.

ANNOTATION

The main objective of this master's thesis is to study deep learning-based facial representation under several different conditions, such as lower and upper face occlusions, misalignment, different angles of head posture, changes in light, and defective localization of facial features using deep learning approaches.

To extract the face representation, CNN-based deep learning models and the OpenCv computer vision algorithm library were used in this work. Because these models show that the deep learning model is resistant to various types of misalignment and can tolerate the interocular distance localization error.

СОДЕРЖАНИЕ

	ВВЕДЕНИЕ	11
1	Распознавание лиц	13
1.1	Общий процесс распознавания лиц	13
1.2	Методы	14
1.3	Распознавание лиц	15
1.4	Выравнивание лица	18
1.5	Идентификация лица	20
1.5.1	Eigenfaces	20
1.5.2	Скрытая Марковская Модель	21
1.5.3	Нейронная сеть	22
1.6	Классификация лиц	24
1.7	Подобные системы	26
1.7.1	Цифровые Технологии Хиквижн	26
1.8	Выводы по главе 1	27
2	ТЕХНИЧЕСКОЕ ИССЛЕДОВАНИЕ	28
2.1	Язык программирования	28
2.1.1	Java	28
2.1.2	Python	29
2.1.3	Summary	29
2.2	Интерактивная Среда Разработки	29
2.3	Выбранные библиотеки	31
2.3.1	OpenCV	31
2.3.2	TensorFlow	32
2.4	Система управления базами данных	
2.4.1	MySQL	32
2.4.2	PostgreSQL	32
2.4.3	Summary	33
2.5	Выводы по главе 2	34
3	МЕТОДОЛОГИЯ РАЗРАБОТКИ СИСТЕМ	35
3.1	Scrum	35
3.2	Waterfall	36

3.3	Выводы по главе 3	37
4	МЕТОДЫ ИССЛЕДОВАНИЯ	
4.1	Введение	40
4.2	Конструкция	41
4.3	Выводы по главе 4	43
5	ПРОВЕРКА ТРЕБОВАНИЙ	44
5.1	Анализ данных	44
5.1.1	Рассылка вопросников	44
5.1.2	Интервью	49
5.2	Выводы по главе 4	49
6	СИСТЕМНАЯ АРХИТЕКТУРА	
6.1	Вступление	50
6.1.1	API	50
6.1.2	Веб-приложение	50
6.1.3	Имитатор ворот	50
6.2	Абстрактная архитектура	51
6.2.1	Проектирование системы	51
6.2.2	Проектирование баз данных	53
6.2.3	Дизайн интерфейса	56
7	ПЛАН ПРОЕКТА	
7.1	Подробности релиза	59
7.1.1	Version I	59
7.1.2	Version II	59
7.1.3	Version III	59
7.1.4	Version IV	59
7.1.5	Version V	59
7.1.6	Version VI	59
7.2	План тестирования	60
7.2.1	Модульный тест	60
7.2.2	Приемочные испытания пользователей	61
8	РЕАЛИЗАЦИЯ	
8.1	Скриншоты	62
8.2	Пример кода	64

8.2.1	Обучение	64
8.2.2	Распознавание	65
8.2.3	Регистрация	66
8.2.4	Контроллер ворот	67
8.3	Последовательность установки	68
8.2.1	Вступление	68
8.2.2	Установите анаконду	68
8.2.3	Установите необходимые модули для Python	69
8.2.4	Установите Torch	69
8.2.5	Установите OpenFace	69
8.2.6	Установите WebServer	70
8.2.7	Сконфигурируйте микроконтроллер Arduino	70
8.2.8	Запустить сервер	70
8.2.9	Запустить врата	70
9	ПРОВЕРКА СИСТЕМЫ	
9.1	Модульный тест	71
9.2	Приемочное тестирование пользователей	73
10	ВЫВОДЫ И РАЗМЫШЛЕНИЯ	74
10.1	Критическая оценка	74
	ЗАКЛЮЧЕНИЕ	76
	СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	77
	APPENDICES	79

ВВЕДЕНИЕ

Система распознавания лиц стала одним из перспективных методов биометрической и также бесконтактной идентификации человека по лицу. Поскольку мир становится все более взаимосвязанным, и все наши данные оцифровываются, заменяя традиционные способы, такие как пароль или карты доступа, в связи с чем данная система является уникальным способом авторизации. Такого рода прогресс в области инноваций и технологий приводит нас к решению одной из сложных проблем современного мира: “Безопасность”. Предлагаемое решение - система безопасности распознавания лиц, может обнаруживать злоумышленников в ограниченных или особо охраняемых зонах и минимизировать человеческие ошибки. Система состоит из двух частей: аппаратной части и программной части. Аппаратная часть состоит из камеры, в то время как программная часть состоит из программного обеспечения для распознавания лиц и алгоритмов распознавания лиц.

Актуальность

Как для личности, так и для предприятия идентификация и безопасность всегда были одной из важных вещей в жизни. Это заставляет людей задуматься о том, как защитить себя от незваных гостей. До сих пор многие компьютерные системы проверки личности основаны на традиционном подходе, когда у кого-то есть (карта доступа или ключ) или кто-то знает (пароль или pin-код). Основная проблема этого подхода заключается в том, что карта доступа или ключ могут быть украдены кем-то или даже потеряны, в то время как человеческий фактор позволяет вам забыть свой pin-код или код, который вы сделали раньше. Это приводит нас к единственному решению, которое предполагает использование чего-то, что может действительно описать конкретную личность для достижения более высокой надежности в верификации.

Биометрическая аутентификация основана на уникальных особенностях личностных характеристик. Это позволяет нам автоматически проверять или идентифицировать на принципах измеримости физические характеристики, такие как распознавание лиц.

Правильно развернутая биометрическая технология в организации предоставляет им очень огромные возможности. Экономика сегодняшнего дня развивалась по мере того, как технический прогресс сделал новый способ ведения бизнеса ведущим. Чтобы выжить в высококонкурентной бизнес-среде, современный бизнес должен быть гибким, гибким и адаптивным. Технология, которую предоставляет распознавание лиц, может принести пользу компаниям в огромном диапазоне секторов, таких как университеты, здравоохранение, розничное предприятие, обрабатывающая промышленность и т. д.

С помощью технологии распознавания лиц идентификация сотрудников и управление бизнесом стали более эффективными, точными и быстрыми.

Личность всегда будет иметь лицо, которое никогда не может быть забыто или потеряно, в отличие от паролей и магнитных карт. Отслеживание посещаемости сотрудников в организации позволит предотвратить кражу времени сотрудников и снизить уровень мошенничества. Система распознавания лиц способна автоматически рассчитать рабочее время сотрудника, что позволит сократить потери бумаги и затраты времени на ручную запись

Целью работы является разработка эффективной и защищенной методики персональной аутентификации с использованием распознавания лиц и создать демонстрационный симулятор для того, чтобы показать эффективность системы на примере реального мира.

Задачи исследования:

- 1) Построить алгоритм распознавания лиц
- 2) Создание оценки ориентира лица для выравнивания лиц на изображениях
- 3) Создать скрипт для распознавания лиц живости
- 4) Сделать скрипт для обучения картин лица для регистрации
- 5) Разработать модель машинного обучения для идентификации лиц из базы данных.
- 6) Создать серверное приложение для выполнения модели ML для идентификации лица, полученной с камеры.
- 7) Разработать базу данных и веб-приложение для отслеживания всех посетителей, отслеживания и отправки оповещений для подозрительных людей.

Научная новизна. Разработаны методы распознавания, что позволило решить важную прикладную задачу поиска и идентификации человека по изображению его лица в реальном масштабе времени. Разработаны алгоритмы определения угла поворота лица, представленного на изображении, в глубину (ракурс лица), отличающиеся низкими затратами на их численную реализацию и допустимой точностью.

Научные статьи и публикации. По теме диссертации опубликована 1 статья в научном журнале “Наука и образование” № 7(54), Июль 2020.

Объем и структура работы. Магистерская диссертация состоит из 10 разделов, заключения, изложена на 68 страницах основного текста. Работа содержит 80 формулу, 46 рисунка, 25 использованных источника и скрипт кода на программе Python.

1 Распознавание лиц

1.1 Общий процесс распознавания лиц

Распознавание лиц рассматривается как биометрическая технология. Среди всех доступных биометрических методов распознавания лиц доказано его превосходство в плане низкой интрузивности и высокой точности среди других. Оставаясь не навязчивым, его точность способна добиться физиологического подхода к идентификации лица. Поскольку требование распознавания человеческих лиц возросло по мере того, как количество приложений в реальном мире нуждается в этом, многие исследователи предложили огромное количество методик для выполнения распознавания лиц. Есть несколько проблем, которые делают распознавание лиц очень трудной задачей. База данных часто получает в качестве входных данных изображение лица индивида, приобретенное в различных условиях. Основная часть автоматической системы распознавания лиц справляется со многими различными экземплярами лица из-за параметров и фактов, таких как поза, освещение, движение, выражение лица, очки, волосы и фон [Zaho, 1999].

Распознавание лиц достаточно продвинуто, чтобы быть примененным во многих различных промышленных приложениях, таких как психология, система безопасности, личная идентификация, обработка изображений и видео, смарт-карта, корпоративная система, правоохранительные органы, наблюдение, компьютерное взаимодействие и так далее. Общая задача распознавания лиц для охранной системы может быть выполнена в двух вариантах: как изображение, так и видеоизображение сцены. Мы должны разделить систему на два различных применения: верификацию и идентификацию. На этапе идентификации входное неизвестное лицо будет сопоставлено с лицами в нашей базе данных, содержащей известные профили. На этапе верификации система отклонит или подтвердит заявленную идентичность входных данных [Fromherz, 1997].

Общий процесс распознавания лиц изображен на рис.1.1. Прежде чем система сможет распознать лицо, она должна определить, есть ли лицо в данном изображении или в видео (последовательности изображений). Этот вид процесса называется распознаванием лиц. Когда система обнаруживает лицо, она должна обрезать его, чтобы изолировать форму лица и передать ее в процесс распознавания [Edwards, 1998]

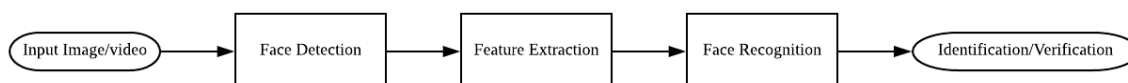


Рисунок 1.1 – Совмещенная гидравлическая характеристика

Распознавание лиц будет начинаться с обнаружения паттернов лиц в Иногда беспорядочной сцене, продолжается нормализацией входных данных лиц в профиль для освещения и геометрических изменений, вероятно, используя информацию о внешнем виде и местоположении ориентира лица, с алгоритмом классификации для соответствующей идентификации лиц, а затем обрабатывает результаты с использованием логистической обратной связи и схем на основе моделей [Chellappa, 1995]. Техника применения распознавания лиц может быть разделена на несколько основных частей: коммерческое применение и применение в правоохранительных органах.

1.2 Методы

Есть несколько различных проблем, которые делают распознавание лиц довольно сложной задачей. Поза отсутствие или наличие структурных компонентов, ориентация изображения, окклюзия, выражение лица. Процесс обнаружения черт лица заключается в обнаружении местоположения и присутствия, таких как глаза, губы, глаза, рот, уши и т. д. Чтобы завершить это, предполагается, что в изображении есть только одно лицо.

В процессе распознавания лиц входное изображение преобразуется в матрицу, а затем сравнивается с другими матрицами в базе данных. Входное изображение также называется зондом, а база данных-галереей. В качестве выходного сигнала он дает отчет о вероятности совпадения, а затем классификация делается для идентификации [Supriya Kakade, 2016].

По словам Суприи Д. Какаде, существует в основном три подхода к распознаванию лиц.:

- *Холистический подход*: в холистическом подходе все лицо берется в качестве входного сигнала в системе распознавания лиц для выполнения распознавания лиц.
- *Подход на основе признаков*: в подходе на основе признаков локальные объекты, такие как нос, глаза, сегментированы, и они могут быть использованы в качестве входных данных при распознавании лиц, чтобы облегчить задачу распознавания лиц.
- *Гибридный подход*: гибридный подход - это сочетание функционального и целостного подхода. В этом подходе в качестве входных данных для системы распознавания лиц используется как локальное, так и целое лицо.

1.3 Распознавание лиц

На первом этапе нашего конвейерного приложения мы должны обнаружить лица в изображении. Очевидно, что прежде чем мы сможем что-то сделать с лицом, мы должны сначала извлечь его из изображения. Если вы пользовались каким-либо смартфоном, сделанным за последние 10 лет, вы, вероятно, уже видели технологию распознавания лиц в действии, как показано на рис.1.2.

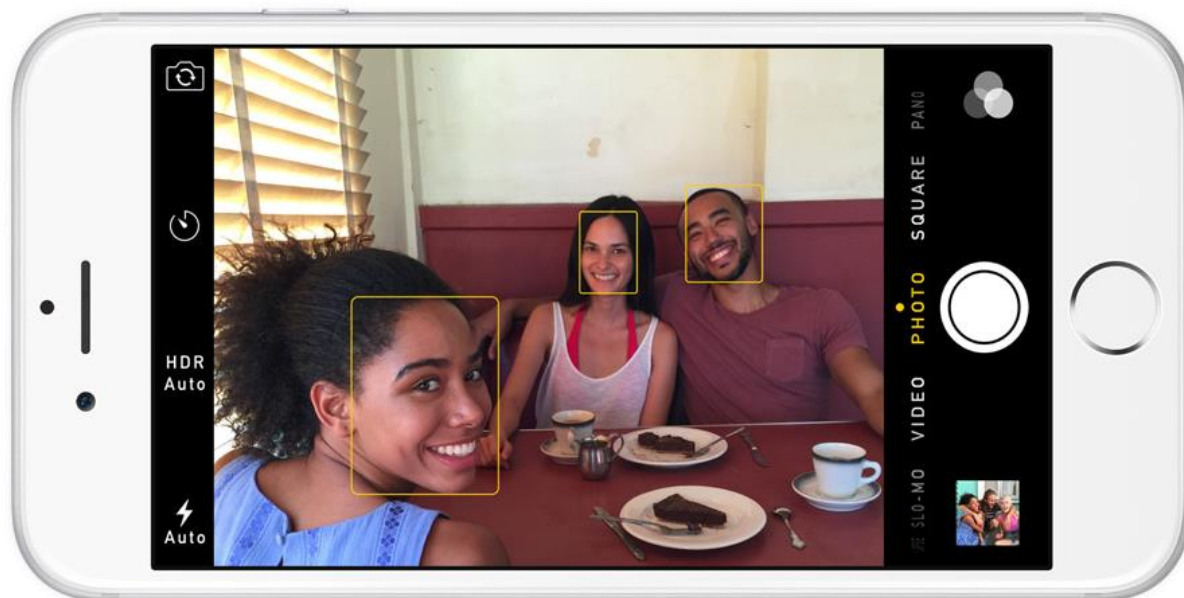


Рисунок 1.2: распознавание лиц в iPhone 6 [macrumors.com]

Распознавание лиц-это классная функция для камеры. Если камера способна автоматически выделять лица, она может убедиться, что все лица находятся в фокусе, прежде чем сделать снимок. Мы собираемся использовать этот алгоритм для другой цели – нам нужно выделить область лица на изображении, которую мы хотим передать на следующий шаг нашего конвейера.

В 2000 году распознавание лиц стало мейнстримом, когда Пол Виола и Майкл Джонс разработали метод распознавания лиц, который в то время был довольно быстр для выполнения его в слабых устройствах. Но сегодня существуют более эффективные и надежные методы. В этом проекте разработчик будет реализовывать алгоритм, предложенный Навнетом Далалом и Биллом Триггсом, называемый гистограммой ориентированных градиентов или сокращенно HOG. Чтобы обнаружить лицо на изображении, мы должны преобразовать цветное изображение в оттенки серого, потому что нам не нужна информация RGB, что снизит эффективность нашего времени вычислений [рис1.3].



Рисунок 1.3: преобразование оттенков серого

Затем мы будем последовательно проходить через каждый пиксель изображения. Но наша цель состоит не в том, чтобы смотреть на каждый отдельный пиксель, нам нужно выяснить, насколько темный текущий пиксель по сравнению с пикселями, которые его окружают. После этого мы нарисуем стрелку в том направлении, где пиксели будут темнее, как показано на рис.1.4.

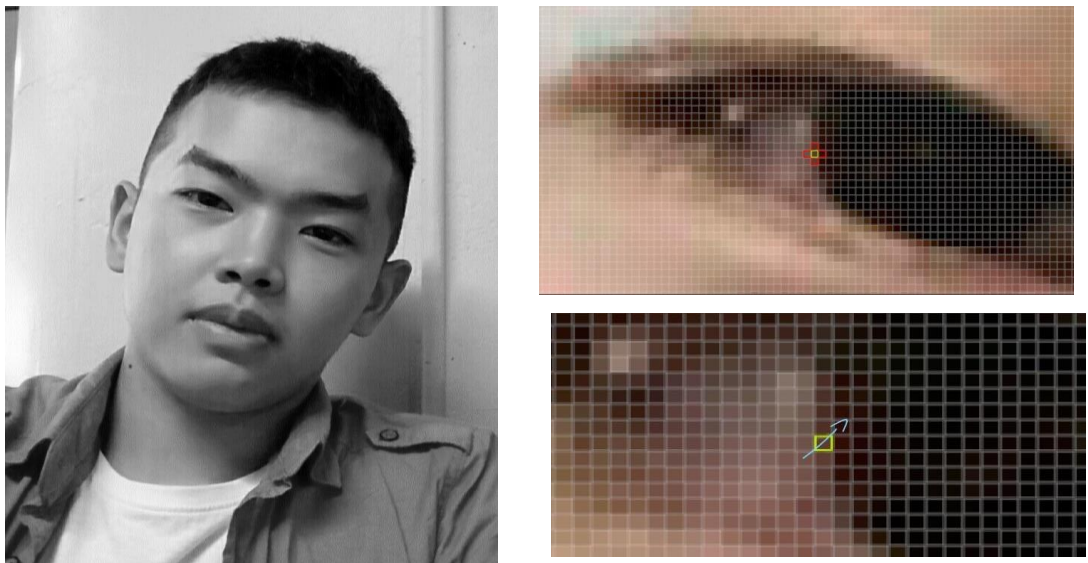


Рисунок 1.4: визуализация алгоритма HOG

Сделав эту процедуру для каждого отдельного пикселя в изображении, мы можем заменить каждый пиксель стрелкой. Название этой стрелки-

градиент, и она показывает направление от светлых пикселей к темным пикселям во всем изображении [рис. 1.5]. Похоже, что мы выбираем этот метод случайным образом, но есть очень веская причина для замены пикселя стрелками. Анализируя каждый пиксель непосредственно, мы можем заметить, что темные и светлые пиксели одного человека будут иметь различные состояния яркости пикселя. Но, если мы будем рассматривать только направление изменения яркости, то получим ту же картину, которая не зависит от яркости входного изображения. Это упростит нашу задачу.

Однако сохранение `gradient` для каждого отдельного пикселя - это слишком много информации, и ее не обязательно знать, поэтому мы можем избежать этого бесполезного вычисления, выделив только общие направления, где мы увидим направление яркости на более высоком уровне. Таким образом, мы сможем получить общую картину изображения.

Чтобы избежать такого рода проблем, мы должны разделить изображение на небольшие квадраты размером 16×16 пикселей каждый. В каждом квадрате мы будем вычислять, сколько точек градиентов повернуто в каждом общем направлении. Затем мы заменим этот квадрат стрелкой, по направлению большинства градиентов.

Наконец, на выходе мы получим сгенерированное изображение, поэтому базовая структура лица будет очень четкой, как показано в Примере на рис.1.5.

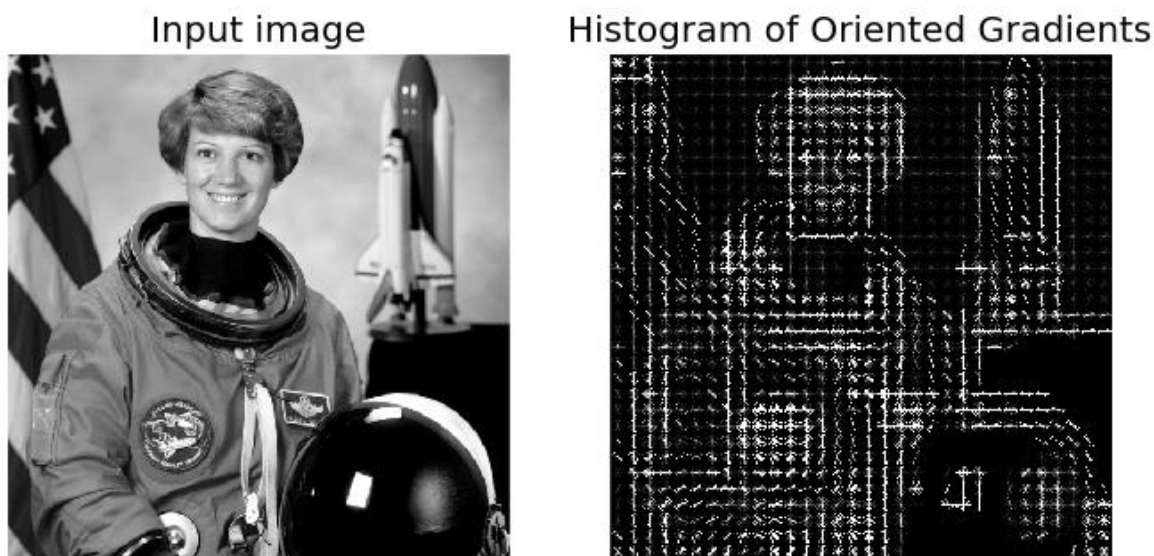


Рис. 1.5: сгенерированное изображение свињи [hypraptive.github.io]

Для того чтобы обнаружить лицо в изображении, сгенерированном свиньей, нам нужно найти кусочек нашего изображения, который похож на знакомый образ свињи, который является результатом обучения нейронной сети через дату многих лиц.

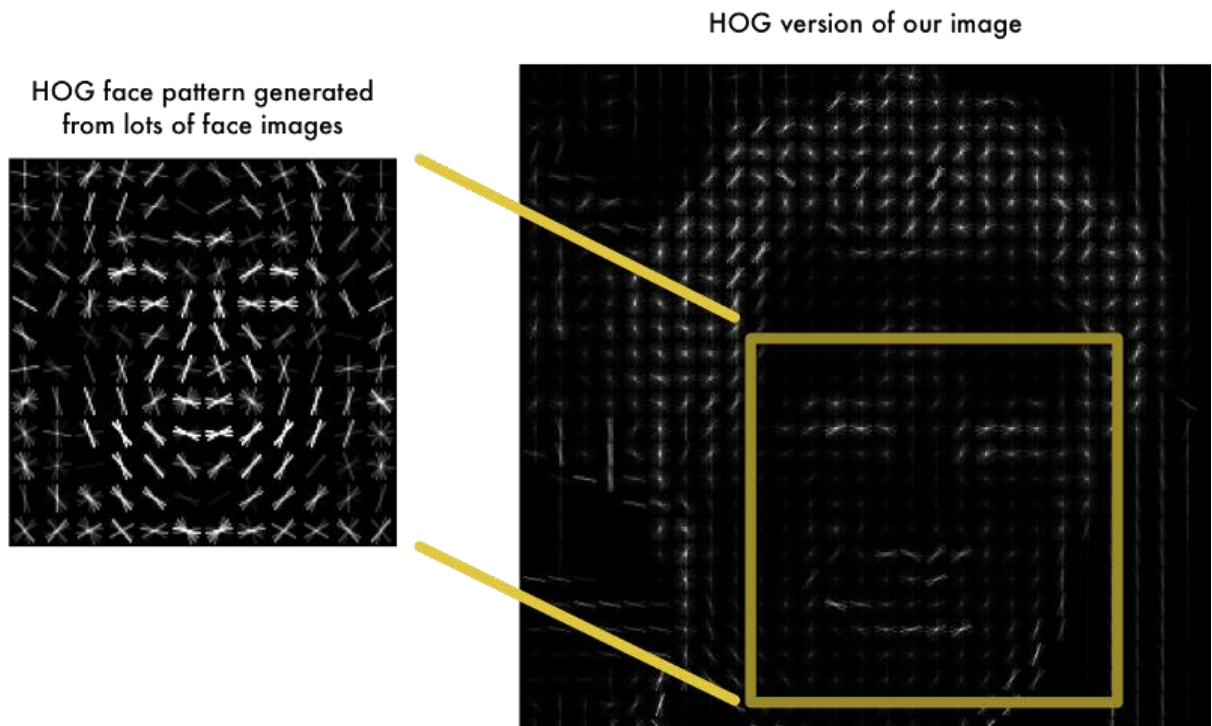


Рис. 1.6: сравнение рисунка морды свињи [datascience.stackexchange.com]

Используя эту технику, мы легко можем обнаружить любое лицо на изображении.

1.4 Выравнивание лица

Как только мы холодно изолируем лицо от изображения, нам нужно решить другую проблему, где лица, повернутые в разные стороны, кажутся очень разными нашему компьютеру, как показано на рис.1.7.

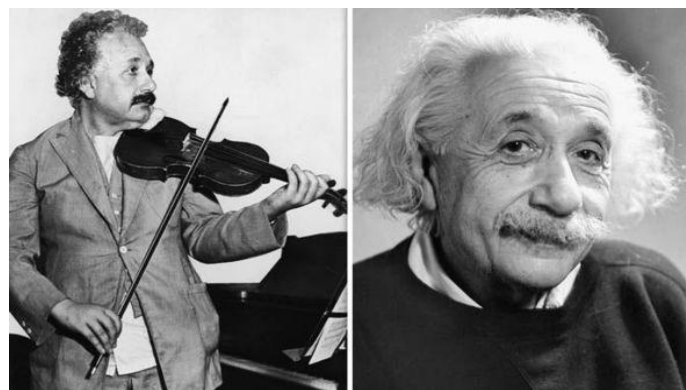
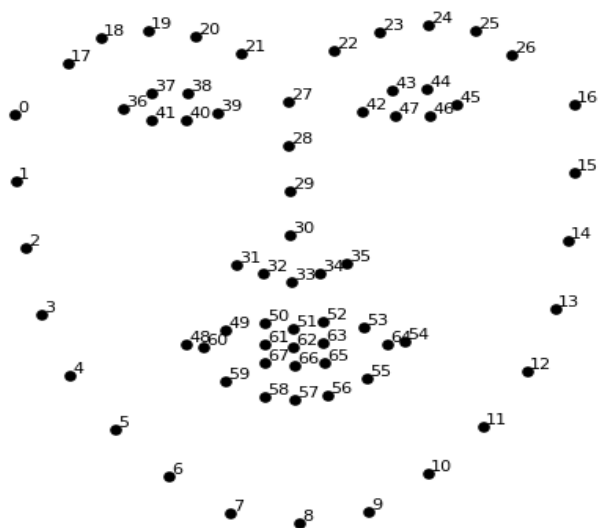


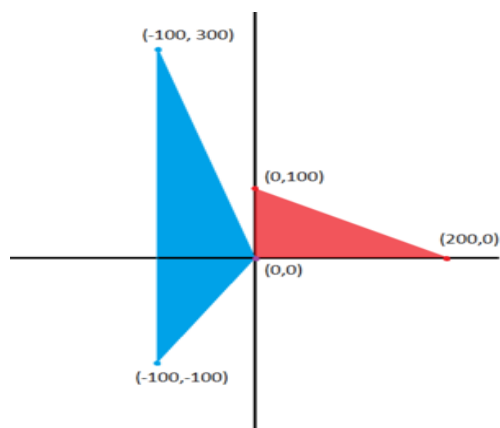
Рис. 1.7: различное направление одной и той же грани [express.co.uk]

Один из подходов, который мы можем сделать, чтобы решить эту проблему, - это передать каждое изображение, так что глаза и рот будут расположены в каком-то конкретном месте. Это упростит задачу распознавания лиц на следующих этапах.



мы будем использовать алгоритмы оценки ориентиров лица, разработанный Вахидом Каземи и Джозефиной Салливан в 2014 году. Основная идея состоит в том, чтобы упомянуть 68 конкретных точек (ориентиров), которые существуют в каждом лице - внешняя точка каждого глаза, внутренняя точка каждой брови, верхняя часть подбородка и т. д. Затем мы обучим наш алгоритм машинного обучения, чтобы идентифицировать эти конкретные 68 точек в любом лице.

Рисунок 1.8: ориентиры [answers.opencv.org]



Теперь, когда мы выясним, где находятся глаза и рот, мы можем вращать, двигать и масштабировать изображение, чтобы глаза и рот могли лучше концентрироваться на изображении. Рис. 2.9. описывается понятие аффинного преобразования: Красный треугольник превращается в синий при аффинном преобразовании $x, y \rightarrow (y-100, 2*x+y-100)$, если новые координаты отображаются в том же базисе.

Рисунок 1.9: аффинное преобразование [Википедия]

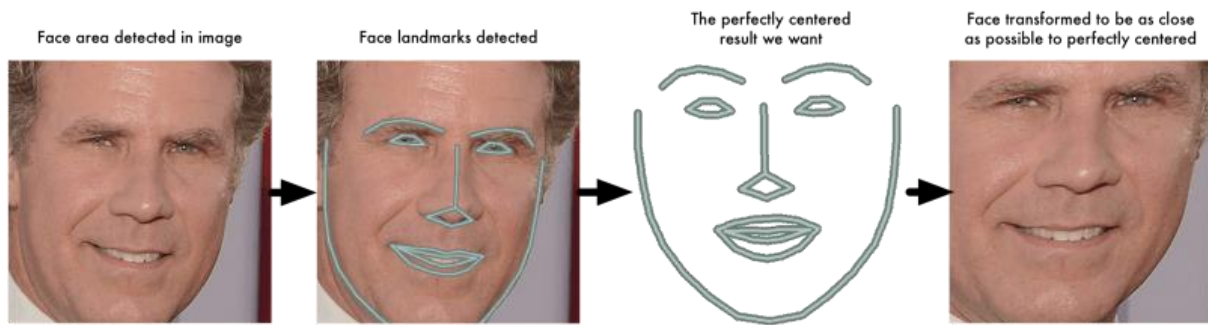


Рис. 1.10: процесс наземной разметки [algotravelling.com]

В результате, независимо от того, как поворачивается лицо, мы можем централизовать глаза и рот в одном месте изображения. Это повысит точность последующих шагов.

1.5 Идентификация лица

Самый простой способ идентификации лица-сравнить неизвестное лицо, которое мы обнаружили на предыдущем шаге, с остальными другими лицами в нашей базе данных. Но, к сожалению, такой подход потребует огромного количества времени для расчетов. Потому что мы хотим, чтобы наша система реагировала в течение нескольких миллисекунд, а не часов. Давайте рассмотрим другие методы распознавания, такие как собственные поверхности, ННМ и нейронная сеть, которые могут соответствовать нашей цели.

1.5.1 Eigenfaces

Вероятно, собственные поверхности являются наиболее исследованными методами среди двух других. Сирович и Кирби в своей работе описали, что изображение лица может быть восстановлено по малому массиву значений для каждого лица и собственной картины. Эти значения были присвоены изображением лица собственной проекции изображения.

С математической точки зрения, собственная поверхность-это набор компонентов распределения граней, в то время как собственные векторы сортируются для представления количества вариаций всех граней. Каждая грань точно может быть представлена в виде линейной комбинации собственных поверхностей. Авторы тестировали свою систему с базой данных, содержащей 2500 изображений 16 профилей, и, как сообщается, 64%, 96% и 85% правильных классификаций по вариациям размера, освещенности и ориентации.

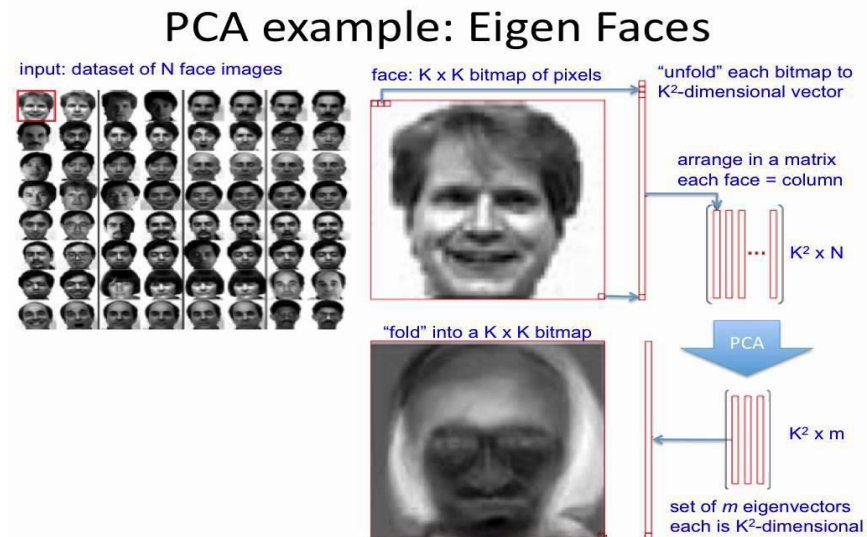


Рис. 1.11: собственные поверхности [скриншот из видео в [youtube.com/user/victorlavrenko](https://www.youtube.com/user/victorlavrenko) канал]

1.5.2 Скрытая Марковская Модель

Эта модель была успешно использована для речевого применения. Самария и Фоллсайд в своих исследованиях реализовали эту методику. Они разделяют лицо на различные части, такие как нос, глаза, рот и т. д., которые могут быть представлены в виде скрытой марковской модели. Поскольку изображения являются двумерными, ХММ требует преобразования их в одномерные.

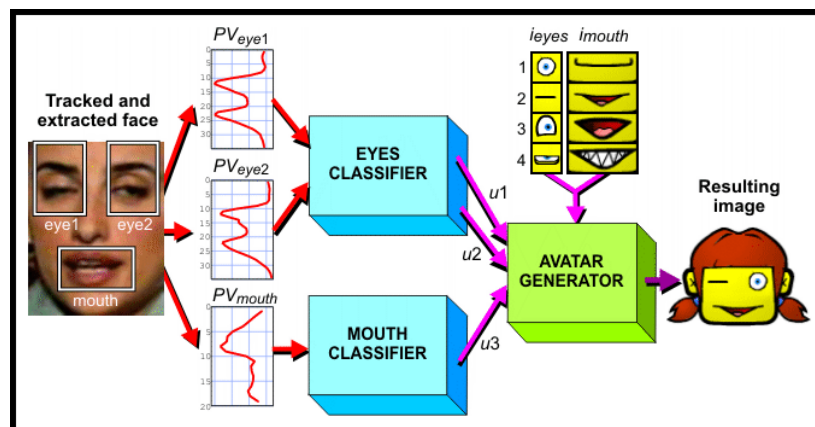


Рис. 1.12: Марковская модель [Дис.единой системы обмена сообщениями. Эс/~ginesgm/fip/problems.html]

Самария и Хартер на втором семинаре IEEE представляли каждую грань одномерным вектором наблюдения пиксельных рядов. Каждый вектор

наблюдения является блоком, который содержит линии X и y, пересекающиеся между успешными наблюдениями. Входное изображение сначала отбирается в наблюдательную последовательность, а затем сопоставляется с каждой скрытой Марковской моделью в базе данных граней. На выходе мы получим лучший матч.

Этот метод НММ был достигнут 87% точности, используя базу данных ORL с 400 изображениями, если 40 профилей. Согласно более позднему эксперименту, описанному в "идентификации человеческих лиц" Гольдштейна, который использовал 2D НММ, было сообщено о 95% точности распознавания лиц.

1.5.3 Нейронная сеть

Оказалось, что вещи, которые кажутся очевидными для людей, чтобы узнать лицо, не имеют никакого смысла для компьютеров. Исследователи пришли к выводу, что наиболее точный подход заключается в том, чтобы позволить машине измерять те особенности, которые она считала важными для себя, и мы можем достичь этого подхода с помощью глубокого обучения.

Решение этой задачи заключается в создании сверточной нейронной сети. Во время обучения сеть будет анализировать три лица одновременно: обучающее изображение лица неизвестного профиля, другое изображение лица того же профиля, изображение лица другого профиля. Затем алгоритм рассмотрит размеры каждого из этих трех изображений. После этого он настроит нейронную сеть таким образом, чтобы размеры первого и второго изображения были максимально близки, а размеры второго и третьего наименее похожи [рис.1.13].

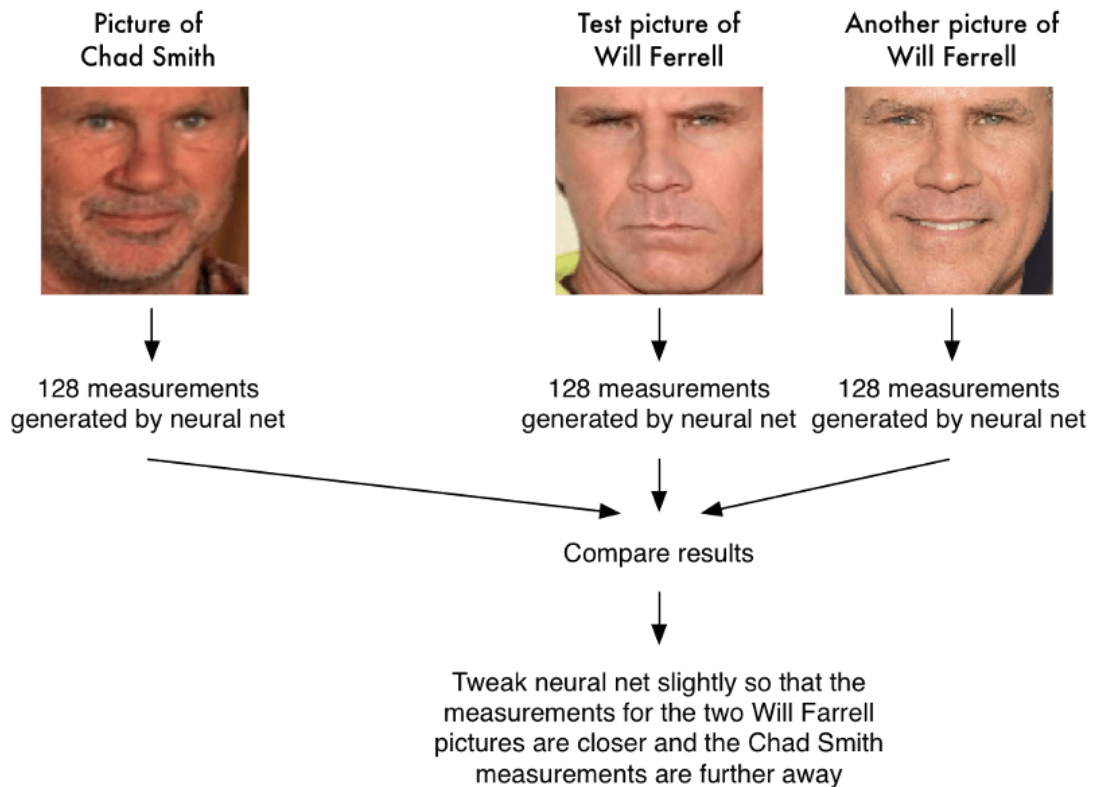


Рис. 1.13: шаг обучения триплетов [algotravelling.com]

Если мы повторим процедуру, как показано на рис.1.13. миллион раз для миллионов изображений тысяч профилей нейронная сеть научится создавать размеры граней для каждой грани. Любые несколько изображений лиц одного профиля должны давать на выходе почти одинаковые размеры.

Измерения, которые мы получили в качестве выходных данных, называются картой. Очень важно передать машине массив необработанных данных, сгенерированных в списке компьютерных чисел, особенно его `coms` много для машинного обучения. Для этого разработчик будет использовать подход идентификации лиц, разработанный в 2015 году исследователями из Google. Но многие альтернативные подходы все еще существуют.

Процесс обучения сверточной нейронной сети для получения точной карты требует огромного объема данных и очень мощного компьютера. Даже если мы будем использовать дорогостоящий графический процессор Nvidia Telsa, нам потребуется 24 часа непрерывной тренировки, чтобы получить высокую точность.

Но как только сеть будет обучена, она сможет генерировать размеры для любых граней, даже для тех граней, которые она видит впервые, и время классификации будет все еще актуальным и меньше 0,5 секунды. Это значит, что тренироваться нужно только один раз. Брэндон Амос и его команда уже

разработали и опубликовали несколько обученных моделей для бесплатного использования.

Таким образом, после обучения нам нужно только позволить изображениям пройти через обученную нейронную сеть, чтобы получить Dimensions (128 измерений) для каждого изображения лица. На рисунке ниже вы можете увидеть и пример этого процесса.

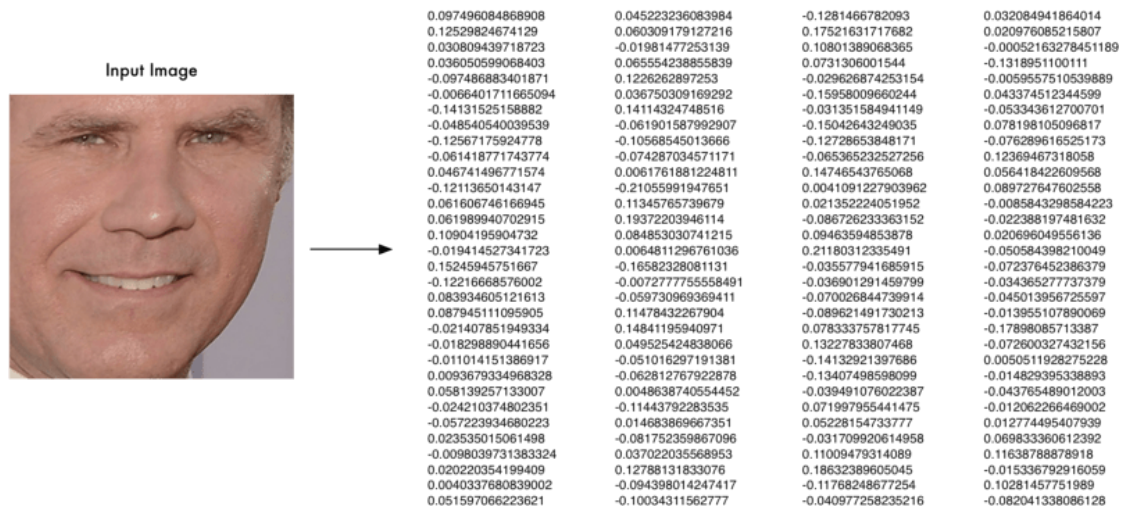


Рис. 1.14: 128 измерений, полученных на основе изображения [algotravelling.com]

1.6 Классификация лиц

На последнем этапе все, что нам нужно сделать, это найти определенное изображение лица в нашей базе данных уже известных профилей. Мы можем сделать это с помощью любого базового алгоритма классификации в машинном обучении.

Наша задача состоит в том, чтобы обучить классификацию, которая будет принимать наши размеры изображения в качестве входных данных и даст нам, какой известный профиль в основном соответствует и выглядит похожим. Для решения этой задачи будет использован алгоритм support vector machine или SVM. Правило простой логистической регрессии выглядит следующим образом:

$$\begin{aligned} \text{класс} &= 1, & \text{if } h(\text{класс}) > 0 \\ \text{класс} &= 0, & \text{if } h(\text{класс}) < 0 \end{aligned}$$

Таким образом, наша гипотетическая функция будет выглядеть $:h(\text{класс}) = \sigma(h(\text{класс}))$, где $\sigma(x) = \frac{1}{1 + e^{-x}}$, форма использует "Сигмоидную

функцию", также называемую "логистической функцией". Функция $g(z)$ или сигмовидная функция, дает любое число в диапазоне от 0 до единицы. Эти выходные значения очень удобны для преобразования произвольно-значной функции в функцию, более подходящую для классификации. $h\theta$ говорит только о проценте вероятности того, что наш выход равен 1. Например, $h\theta=0.9$ дает нам вероятность 90%, что наш выход равен 1, как показано ниже:

$$h_{\sigma}(\mathbf{x}) = \sigma(\mathbf{x}; \theta) = 1 - \sigma(\mathbf{x}; \theta)$$

$$\sigma(\mathbf{x}; \theta) + \sigma(\mathbf{x}; \theta) = 1$$

Функция затрат - это функция, которая отображает событие или значения одной, или нескольких переменных на действительное число, интуитивно представляющее некоторую "стоимость", связанную с событием. Цель выполнения функции затрат состоит в том, чтобы выяснить, что лучше всего подходит для некоторых заданных входных данных

$$J(\theta) = \frac{1}{n} \sum_{i=1}^n \left[\sigma(\mathbf{x}^{(i)}; \theta) y^{(i)} + (1 - \sigma(\mathbf{x}^{(i)}; \theta)) (1 - y^{(i)}) \right] + \frac{\lambda}{2n} \sum_{i=1}^n \theta_i^2$$

$$\frac{\partial J(\theta)}{\partial \theta} = \left(\frac{1}{n} \sum_{i=1}^n (y^{(i)} - \sigma(\mathbf{x}^{(i)}; \theta)) \mathbf{x}^{(i)} \right) + \lambda \theta$$

$$= \frac{1}{n} \sum_{i=1}^n (y^{(i)} - \sigma(\mathbf{x}^{(i)}; \theta)) \mathbf{x}^{(i)} + \lambda \theta$$

На самом деле, то, что эта функция вычисляет гипотетическое значение минус фактическое значение, а затем все квадратное. Наконец, мы получим одно число. И какое бы наименьшее число мы ни получили, оно лучше всего нам подходит.

Градиентный спуск - это итерационный алгоритм оптимизации первого порядка для нахождения минимума функции. Если вместо этого сделать шаги, пропорциональные положительному значению градиента, то можно приблизиться к локальному максимуму этой функции; тогда процедура называется градиентным подъемом.

1.7 Подобные системы

1.7.1 Цифровые Технологии Хиквижн



Рис. 1.17: камера Hikvision [hikvision.com]

Компания Hikvision, ведущий мировой поставщик инновационных продуктов и решений для видеонаблюдения, продолжила выпуск первых в мире видеорегистраторов глубокого обучения с новой серией IP-камер. Новая серия IP-камер "Deepin View" обеспечивает мощь и интеллект для повышения производительности систем видеонаблюдения в широком диапазоне приложений безопасности и управления.

Алгоритмы глубокого обучения Hikvision имеют гораздо более глубокое Программирование по сравнению с обычными интеллектуальными алгоритмами, которые работают только на поверхностном уровне. Эти алгоритмы выполняют функциональное обучение и обеспечивают удивительно точную и последовательную производительность анализа видеоконтента (VCA). В сочетании с высокоскоростным графическим процессором камеры Hikvision Deepin View демонстрируют более быстрые вычисления с большими объемами данных [hikvision.com].

Распознавание лиц может быть развернуто во многих случаях безопасности и контроля в рамках расширения программ, чтобы предупредить операторов гаджетов о присутствии людей, представляющих интерес. Это достигается путем моделирования изображения лица и вычисления подобия, заплетенного в систему. Эта эпоха становится решающим инструментом против занесенных в черный список сотрудников, заходящих в казино, потому что личность персонажа может быть использована для оповещения охранников о присутствии признанного преступника, позволяя сотрудникам охраны отказаться от доступа в это казино. На этом пути они потенциально могут спасти вас от нечестного поведения. Эта технология так же хорошо работает в предотвращении допуска рассматриваемых правонарушителей в места проведения наряду со спортивными мероприятиями стадионы или

рестораны.

Соответствие лица

По сути, устройство популярности лица можно определить как процедуру сопоставления лиц, застрявших на видео в режиме реального времени, с базой данных ранее сохраненных фотографий лиц. Лица, накопленные в базе данных, могут быть разделены на уникальные классы. Например, то, что обычно называют “белым списком”, - это список мужчин и женщин, которые должны быть допущены в определенную зону и могут содержать персонал, почетных посетителей, VIP-клиентов и так далее. В качестве альтернативы “черный список “или” список наблюдения ” могут включать лиц, которые по той или иной причине нежелательны.

Технические условия

- *Компьютеры и сеть.* Программное обеспечение для распознавания лиц основано на вычислениях сложных математических алгоритмов, что требует более мощного ноутбука или сервера, чем то, что обычно требуется для обычных систем видеонаблюдения. IP-сеть позволяет интегрировать с различными IP-преимущественно основанными устройствами наблюдения или пакетами, состоящими из get entry для управления системами.
- *Сетевая камера высокого разрешения.* Предпочтение отдается камерам с высоким разрешением (минимум 1080p), поскольку они делают операции гораздо более эффективными и надежными, а также незаметными и чрезвычайно универсальными. Однако, независимо от камер с избыточным разрешением, лицо все равно должно быть близко к камере.

1.8 Выводы по главе 1

До сих пор мы рассматривали некоторые исследовательские работы, которые помогут нам разработать систему, к которой стремится разработчик. Последовательность наших действий следующая: 1) закодировать изображение с помощью алгоритма HOG для создания упрощенного изображения, чтобы найти ту часть изображения, которая выглядит как сгенерированная HOG кодирующая картинка лица; 2) выяснить положение лица в кадре изображения, упомянув конкретные точки лица. Как только мы получим эти очки, нам нужно будет централизовать этот образ глазами и ртом; 3) генерировать это централизованное Изображение через нейронную сеть, которая может измерять символы лица и сохранить его вывод в виде 128-мерного массива; 4) сравнивать эти размеры с другими и находить наиболее похожие с помощью машинного обучения.

2 ТЕХНИЧЕСКОЕ ИССЛЕДОВАНИЕ

2.1 Язык программирования

2.1.1. Java

История Java началась в 1991 году, и хотя до 1995 года она не была выпущена. Sun Microsystem под руководством Джеймса Кослинга предложила язык, который может программировать интерактивную телевизионную систему. Он был разработан специально для того, чтобы быть портативным языком интернета, особенно в браузерах. В то время это было довольно тяжело для рынка, на который они нацеливались, и они потеряли пространство браузера для JavaScript и HTML. Но, тем не менее, Java может занять место на рынке серверных приложений и университетских аудиторий. Это помогло Java остаться в рейтинге как одному из самых доминирующих языков.

Его относительная эффективность и вероятность являются частью его ценности и привлекательности. Java-это компилируемый язык, а не такие родные языки, как C и c++. Java имеет более централизованную модель выполнения машин, по сравнению с чисто интерпретируемыми языками, такими как Perl и Python. Java-это больше, чем просто библиотека и язык, Java-это также экосистема и виртуальная машина.

Виртуальная машина Java или JVM в сортировке - это портативная и идеализированная платформа для запуска кода Java. Вместо того чтобы беспокоиться о специфике аппаратного обеспечения и необходимости переносить код на новые платформы, обещание Java было написано один раз и запущено в любом месте. Это означает, что пока существует JVM, все, что скомпилировано в его байт-код, может легко работать и взаимодействовать со всем, что написано для JVM. Существует множество языков JVM, в том числе более скриптоподобный Groovy, функциональный Clojure, объектно-функциональный гибридный Scala и даже вариант Python, Jython [Kevlin Henney, 2017].

Java-это язык ООП, такой же, как синтаксис C/C++, который хорошо известен многим программистам. Он динамически связан и позволяет загружать и запускать новый код, но не динамически набирается. Эволюция Java была довольно медленной как языка, только недавно она выпустила функцию, поддерживающую функциональное программирование. С другой стороны, философия как языка, так и виртуальной машины состояла в том, чтобы рассматривать обратную совместимость как основную директиву [спецификация языка Java].

Когда Sun была куплена Oracle, компилятор и сам язык были в конечном итоге как проект с открытым исходным кодом. А оценка языка ориентируется

на процесс Java Community, в котором участвуют компании и частные лица, не входящие в Oracle.

2.1.2. Питон

Python был реализован в 1991 году Гвидо ван Россумом. С момента своего создания python был открытым исходным кодом. Программный Фонд Python управляет стандартизацией и проектированием языка и его библиотек. Процесс предложения по улучшению Python руководил его разработкой [Kevlin Henney, 2017].

2.1.3. Summary

Language	Преимущества	Недостатки
Python	Легкий для изучения Поддерживаются несколько систем Может работать с Raspberry Pi Быстрое развитие Позволяет масштабировать приложения Меньший объем кода Используется мировыми лидерами,	Скорость Не очень хорошо для мобильной разработки Плохо подходит для задач с интенсивной памятью Ограничения доступа к базе данных
Java	Межплатформенный Безопасность Большое сообщество поддержки	Скорость Медленный сборщик мусора Сложность

Таблица 2.1: Python vs Java

2.2 Интерактивная Среда Разработки

До сих пор, чтобы построить программное приложение (основную часть), разработчик выбирает Python. PyCharm является одним из наиболее широко используемых IDE для Python. В настоящее время он используется огромными компаниями, такими как HP, Twitter и т. д. PyCharm была разработана в качестве кросс-платформенной интегрированной среды разработки от JetBrains. Все функции и инструменты, предоставляемые этой

IDE, помогают программистам эффективно и быстро создавать многие виды программных приложений. Если программисты пожелают, они даже могут настроить пользовательский интерфейс PyCharm в соответствии со своими потребностями.

Редактор кода

PyCharm предоставляет интеллектуальный редактор кода, который может помочь программистам писать код в высоком качестве. Это позволит программистам легко читать код: предложение завершения кода, выбор соответствующего стиля кода, автоматическая вставка отступов в новые строки и т. д.

В то же время программисты могут также использовать редактор для использования фрагментов кода, расширения блока кода до выражения или логического блока, выявления ошибок и опечаток, форматирования базы кода, автоматической генерации кода, обнаружения дубликатов кода. Кроме того, PyCharm обеспечивает анализ кода и идентификацию ошибок при написании кода.

Навигатор Кодов

PyCharm может помочь программистам улучшить и отредактировать код без каких-либо дополнительных усилий и времени с помощью интеллектуальных опций навигации по коду. Это сделает его более удобным, чтобы перейти к файлу, классу или другим объявлениям из ссылок.

Рефакторинг

PyCharm позволяет разработчикам быстро и эффективно внедрять как локальные, так и глобальные изменения. Разработчики могут даже воспользоваться преимуществами рефакторинга, предоставляемого IDE, при написании простого кода Python и работе с фреймворками Python. Они могут воспользоваться рефакторингом переименования и перемещения файлов, классов, функций, методов, свойств, параметров и локальных/глобальных переменных. Кроме того, они могут улучшить качество кода, извлекая переменные, поля, константы и параметры. Кроме того, PyCharm позволяет программистам разбивать более длинные классы и методы с помощью метода извлечения

Инструмент Базы Данных

Помимо поддержки различных библиотек и фреймворков Python, PyCharm позволяет разработчикам работать с рядом реляционных баз данных, включая Oracle, SQL Server, MySQL и PostgreSQL. Разработчики могут также использовать IDE для выполнения запросов, редактирования кода SQL, просмотра данных, изменения табличных данных и изменения/анализа схем. PyCharm также поддерживает библиотеку SQLAlchemy и вводит SQL-код в

код, написанный на различных языках программирования. Профессиональная версия IDE также облегчает разработчикам эффективную обработку больших объемов данных с помощью сеток данных.

Тестирование программного обеспечения

Как и другие IDE, PyCharm также поставляется с функциями и инструментами для упрощения тестирования приложений Python. Он позволяет разработчикам выполнять модульное тестирование с помощью популярных платформ тестирования Python, таких как Nose, Attest и Doctests. Тестировщики даже имеют возможность запускать отдельные или несколько тестовых файлов и тестовых классов. Они могут дополнительно интегрировать IDE с Coverage.py для измерения покрытия кода во время тестирования приложений. При тестировании многопоточных приложений тестировщики могут использовать функцию визуализации параллелизма потоков, предоставляемую IDE, для полного и эффективного управления приложением. В то же время PyCharm позволяет пользователям предоставлять высококачественное программное обеспечение, реализуя разработку, основанную на поведении.

2.3 Выбранные библиотеки

2.3.1 OpenCV

OpenCV имеет около 2000 оптимизированных алгоритмов, включая полный набор современных и классических алгоритмов машинного обучения и компьютерного зрения. Эти алгоритмы могут быть использованы для различных задач, таких как распознавание лиц, распознавание объектов, классификация действий человека, отслеживание объектов, извлечение 3D-моделей объектов и т.д. В сообществе OpenCV задействовано более 47 тысяч человек, а предполагаемое количество скачиваний составляет около 14 миллионов. Эта библиотека широко используется в компаниях, исследовательских группах и правительственных органах [Realtime Computer Vision with OpenCV, 2012].

Наряду со многими стартапами, такими как VideoSurf, Zeitera и Applied Minds, есть много хорошо зарекомендовавших себя компаний, таких как Yahoo, Microsoft, Google и т. д., которые широко используют OpenCV. Развернутое использование OpenCV охватывает диапазон от обнаружения вторжений в видеонаблюдения в Израиле, сшивания изображений street view вместе, помощи роботам в навигации и подборе объектов в гараже Willow, мониторинга шахтного оборудования в Китае, запуска интерактивного искусства в Испании и Нью-Йорке, обнаружения аварий с утоплением

плавательных бассейнов в Европе, проверки взлетно-посадочных полос на предмет мусора в Турции, проверки этикеток на продуктах на заводах по всему миру до быстрого обнаружения лиц в Японии.

Он имеет интерфейсы MATLAB, Java, Python и C++ и поддерживает Mac OS, Android, Linux, Windows. OpenCV использует преимущества SSE и MMX и склоняется к системам видения в реальном времени. Полнофункциональные интерфейсы OpenCL и CUDA хорошо построены прямо сейчас. Существует более 500 алгоритмов и примерно в 10 раз больше функций, которые составляют эти алгоритмы. Изначально OpenCV написан на C++, и у него есть интерфейс, который хорошо работает с библиотекой STL.

2.3.2 TensorFlow

TensorFlow-это система Google Brain второго поколения. Это математический фреймворк, а также используемый для приложений машинного обучения, таких как нейронные сети. Начиная с 2011 года, Google Brain был сконструирован DistBelief. Это была проприетарная система машинного обучения, основанная на глубоком обучении нейронных сетей. Он быстро рос, охватывая различные алфавитные компании как в коммерческих приложениях, так и в исследованиях [Википедия].

Даже реализация приложения, выполняемая на одной машине, TensorFlow может быть выполнена на нескольких графических процессорах и процессорах. Он доступен на 64-битных компьютерах macOS, Linux, Windows и мобильных вычислительных устройствах, включая iOS и Android. Гибкая архитектура TensorFlow позволяет нам легко развертывать вычисления на различных платформах, от пограничных и мобильных устройств до кластеров серверов и настольных компьютеров [Dean, 2015].

Процессы вычисления тензорного потока выражаются как график потока данных. Название TensorFlow происходит от операций, которые такие нейронные сети выполняют с многомерными массивами данных. Эти массивы называются "тензорами". В июне 2016 года Дин заявил, что 1500 репозитория на GitHub упоминали TensorFlow, из которых только 5 были от Google.

2.4 СУБД

2.4.1 MySQL

MySQL - это система управления базами данных с открытым исходным кодом, основанная на структурированном языке запросов. MySQL может быть выполнен на всех платформах, включая Windows, UNIX и Linux. Кроме того, он может использоваться в огромном диапазоне систем, но в основном ассоциируется с веб-приложениями.

MySQL стал важной особенностью LAMP, которая представляет собой корпоративный стек с открытым исходным кодом. LAMP-это веб-приложение, которое работает на операционной системе Linux и веб-сервере Apache. MySQL был приобретен Sun Microsystems в 2008 году, затем Oracle, когда они купили Sun, но первоначально он был задуман шведской компанией MySQL AB. Любой может использовать MySQL под общей публичной лицензией, но корпоративные пользователи должны приобрести коммерческую лицензию у Oracle [MySQL reference manual].

Сегодня MySQL является СУБД, стоящей за многими ведущими веб-сайтами в мире и бесчисленными корпоративными и потребительскими веб-приложениями, включая YouTube, Twitter и Facebook.

2.4.2 PostgreSQL

PostgreSQL, или сокращенно Postgres, - это объектно-реляционная система управления базами данных, которая делает акцент на соответствие стандартам и расширяемости. Эта база данных может обрабатывать рабочие нагрузки от небольших до больших приложений или для хранения данных с большим количеством одновременных пользователей. Для macOS Server базой данных по умолчанию является PostgreSQL, а также она доступна для Linux и Microsoft Windows [документация по PostgreSQL].

PostgreSQL имеет материализованные представления и обновляемые представления, внешние ключи, триггеры; хранимые процедуры и поддерживающие функции, а также другие возможности расширения. PostgreSQL был создан глобальной группой разработчиков PostgreSQL, отдельными участниками и разнообразной группой многих компаний. Он является бесплатным и открытым исходным кодом, выпущенным на условиях лицензии PostgreSQL, разрешительной лицензии на программное обеспечение [PostgreSQL Global Development Group, 2016].

2.4.3 Вывод

2.4.3 Вывод

Database	Преимущества	Недостатки
MySQL	Простой в использовании Широкое распространение Большая производительность открытый исходный код Двигатель InnoDB Использование силы памяти	Трудно масштабировать Производительность ухудшается
PostgreSQL	Иммунитет к чрезмерному разворачиванию	Медленнее, чем MySQL

	Лучшая поддержка, чем у проприетарных поставщиков Значительная экономия на кадровых расходах Расширяемый	
--	--	--

Таблица 2.2: MySQL vs PostgreSQL

2.5 Выводы по главе 2

Согласно нашему проекту, приложение должно взаимодействовать с камерой и микроконтроллером, который может сканировать лицо, как только индивид приблизится к нашему входу. И основываясь на нашей сравнительной таблице, нет особых различий между Java и Python, за исключением того, что Python может работать с микроконтроллером Raspberry Pi. Кроме того, как вы можете видеть из таблицы 3.1, есть несколько сильных преимуществ использования Python, который превосходит Java: меньше времени и меньше денег, он включает в себя меньше кода, он используется мировыми лидерами, он узаконен и одобрен учеными по обработке данных, и он ползет за Java как восходящая звезда номер 1.

Приложение распознавания лиц будет выполнять очень сложные вычисления, и из-за этого время и стоимость являются основными преимуществами Python, по сравнению с Java. Таким образом, из-за этого факта подходящим языком программирования, который соответствует нашим потребностям, будет Python. А среда PyCharm была выбрана для разработки проекта, так как она обладает всеми необходимыми функциями и позволит разработчику больше сосредоточиться на концепции программирования.

Когда дело доходит до выбора базы данных, как вы можете видеть из таблицы 3.2. MySQL и PostgreSQL кажутся подходящими для использования в нашем проекте. Но, согласно таблице 3.2, у использования MySQL есть несколько преимуществ: он прост в использовании и имеет решающее значение для экономичной доставки и быстрого создания как улучшенных, так и новых приложений; одной из отличительных черт MySQL является исключительная производительность и масштабируемость, именно поэтому так много веб-компаний используют MySQL; и, наконец, поскольку MySQL более широко распространен, а это означает, что если разработчик получит какую-либо проблему или проблему с базой данных, он сможет быстрее получить решение от других разработчиков сообщества MySQL.

3 МЕТОДОЛОГИЯ РАЗРАБОТКИ СИСТЕМ

3.1 Scrum

Scrum-это практика гибкой методологии, которая позволяет команде работать как кросс-функциональное подразделение с самоорганизацией и ежедневным общением. Это облегчает непрерывную регулировку на каждом этапе для достижения наилучшего качества конечного продукта. Использование методов Scrum помогает более эффективно управлять процессами проекта и иметь полный контроль над их расписанием и состоянием [Claudia Gutierrez, 2016].

Преимущества:

- Будучи гибким, принимает обратную связь от заинтересованных сторон и клиентов
- Короткие спринты позволяют гораздо легче вносить изменения на основе обратной связи
- Индивидуальные усилия каждого члена команды видны во время ежедневных встреч scrum
- Scrum обеспечивает эффективное использование времени и денег
- Разработки кодируются и тестируются во время спринтерского обзора
- Хорошо работает для быстро движущихся проектов развития
- Команда получает четкую видимость через scrum-встречи

Недостатки:

- Внедрения методологии Scrum в больших коллективах сложно
- Ежедневные встречи иногда расстраивают членов команды
- Если какой-либо член команды покидает проект в середине, это оказывает огромное негативное влияние на проект
- Scrum часто приводит к ползучести объема из-за отсутствия определенной конечной даты
- Вероятность провала проекта высока, если люди не очень привержены или сотрудничают
- Фреймворк может быть успешным только с опытными членами команды
- Качество трудно реализовать, пока команда не пройдет через агрессивный процесс тестирования

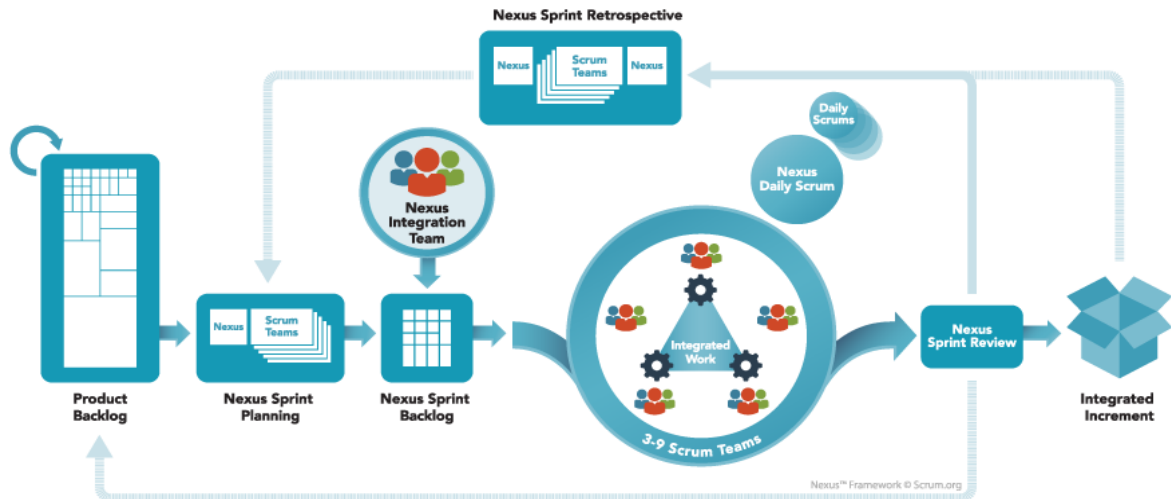


Рисунок 3.1. Методология Scrum [scrum.org]

3.2 Waterfall

Модель водопада-это особенно линейный последовательный подход к компоновке для положительных областей инженерной компоновки. В совершенствовании программного обеспечения это, как правило, некоторые из гораздо менее итеративных и гибких процессов, поскольку прогресс протекает в значительной части одного курса через уровни идеи, инициации, оценки, проектирования, производства, тестирования, развертывания и технического обслуживания [Википедия].

Версия улучшения водопада возникла внутри обрабатывающей промышленности и строительной промышленности; где специально основанные физические среды предполагали, что корректировка дизайна стала непомерно дорогостоящей намного быстрее внутри процедуры разработки. В то время как сначала следовали за улучшением программного обеспечения, не было диагностированных альтернатив для инновационных картин, основанных на ноу-хау.

Преимущества

- В этой модели этапы обрабатываются и завершаются по одному за раз. Фазы не пересекаются.
- Модель водопада хорошо работает для небольших проектов, где требования очень хорошо поняты.
- Эта модель проста и удобна для понимания и использования.
- Им легко управлять из – за жесткости модели-каждый этап имеет конкретные результаты и процесс обзора.

Недостатки

- Высокий уровень риска и неопределенности.
- Плохая модель для длительных и текущих проектов.
- Не подходит для проектов, где требования подвергаются умеренному или высокому риску изменения.
- Как только приложение находится на стадии тестирования, очень трудно вернуться назад и изменить что-то, что не было хорошо продумано на стадии концепции.
- Ни одно рабочее программное обеспечение не производится до конца жизненного цикла.
- Не очень хорошая модель для сложных и объектно-ориентированных проектов.

3.3 Выводы

Scrum	Waterfall
Архитектура неофициальных и дополнительных	Архитектура очень хорошо документирована и завершена до начала кодирования
Разработчики разделяют право собственности на код	Каждый застройщик отвечает за одну область
Непрерывная интеграция	Интеграция выполняется в конце или после контрольных точек
Сосредоточьтесь на завершении историй в коротких итерациях	Сосредоточьтесь на завершении модулей на различных крупных этапах
Опирается на инженерные практики	Не обязательно полагаться на инженерные практики
Легкий процесс и документация	Тяжелый процесс и документация
Требуются кросс-талантливые разработчики, знающие все необходимые технологии	Опирается на небольшую группу разработчиков
Главные роли: разработчики	Основные роли: архитектура и разработчик

Таблица 3.1: Scrum vs Waterfall

Водопад никогда не движется вперед и не пойдет дальше, пока предыдущий шаг не будет завершен, он опирается на следующую последовательность шагов. Этот вид структуры подходит для небольших проектов, таких как мой, с результатами, которые легко определить в самом начале.

По сравнению с другими методологиями, водопад больше всего ориентирован на определенный и четкий набор шагов. Он имеет очень простую структуру, как показано на рис. 3.2. Кроме того, Waterfall обладает интуитивным развитием и, в отличие от scrum, не требует специальной сертификации для сотрудников или менеджеров проектов. Еще одним преимуществом использования водопада является его приверженность конечной цели или достижению в самом начале. Для текущего проекта FYP, где цель ясна, он будет знать разработчику общую цель с самого начала, с меньшей вероятностью заблудиться по мере продвижения проекта вперед

Во время развития нашего проекта будет очень меньше взаимодействия с клиентами. И он может быть продемонстрирован пользователям, как только проект будет полностью завершен. Все требования уже известны, и разработчик предполагает, что в дальнейшем дополнительных возможностей не возникнет. Учитывая все эти факты, разработчиком была выбрана Водопадная методология (рис. 3.2). Модель водопада делится на 6 этапов: выявление требований, проектирование, внедрение, тестирование, развертывание и выпуск.

Step	Description
Требование	На этом этапе разработчик определит, что должно делать приложение. Например: 1) камера должна начать сканировать лицо пользователя, только когда он пересек определенную линию; 2) приложение должно распознавать поддельные и живые лица. 3) система должна отслеживать всех посетителей в базе данных и должна отслеживать все это на веб-сайте; и так далее.
Дизайн	Этот этап в значительной степени охватывает технические требования к проектированию. Разработчик сам решит, какой язык программирования лучше всего подходит для распознавания лиц. Поскольку распознавание лиц включает в себя множество алгоритмических задач, вероятно, будут использоваться различные методы и языки программирования.

Реализация	Фактический исходный код приложения, наконец, пишется на этом третьем этапе, реализуя все модели, бизнес-логику и сервисные интеграции, которые были указаны на предыдущих этапах. После выполнения каждой задачи мы должны проверить ее правильность.
Тестирование	Разработчик должен систематически обнаруживать и сообщать о проблемах в приложении, которые необходимо решить. Приложение распознавания лиц - это конвейерная система, а это значит, что первые задачи системы влияют на точность следующих. Например, чтобы идентифицировать лицо, нам нужно сначала обнаружить лицо на изображении, если точность распознавания лица будет низкой, это снизит точность идентификации лица.
Развертывание	На этом этапе разработчик активирует весь функционал системы распознавания лиц как целостного интегрированного проекта. Он обеспечит, чтобы общий процесс развертывания состоял из нескольких взаимосвязанных мероприятий с возможными переходами между ними.
Выпуск	Наконец, приложение готово к выпуску в живую среду. Этап эксплуатации включает в себя не только развертывание приложения, но и последующую поддержку и техническое обслуживание, которые могут потребоваться для поддержания его функциональности и актуальности.

Таблица 3.2: Waterfall steps

4 МЕТОДЫ ИССЛЕДОВАНИЯ

4.1 Вступление

Людей, которые принимают участие в исследованиях, связанных с экспериментами, могут попросить сделать некоторые тесты для оценки их когнитивных способностей на бумаге или компьютере. После этого сравниваются результаты разных групп. Участники не должны быть обеспокоены, а должны просто сделать все возможное для достижения результатов. Основная цель этих тестов состоит не в том, чтобы судить о людях или измерять их интеллект, а в том, чтобы найти связь между производительностью и другими факторами [Alzheimer Europe, 2018].

Сбор и оценка статистических данных могут быть полезны для качества результатов проекта с помощью поддержки понимания характера, состояния и функций конкретных результатов, значимости результатов для всего проекта, конкретных подходов к поставке и оплате результатов, количества времени, необходимого для завершения результатов, далее, альтернатив и преимуществ в дополнение к негативным аспектам каждого результата проекта распознавания лиц. Для этого проекта в качестве основных методов сбора данных были выбраны интервью и анкета.

Вопросники-это очень хороший способ получить записи от большого числа людей и / или людей, у которых может не хватить времени на интервью или участие в экспериментах. Они позволяют людям не торопиться, обдумать это и вернуться к анкете позже. Участники могут высказывать свои взгляды или чувства в частном порядке, не травмируя при этом возможную реакцию исследователя. К сожалению, некоторые люди все еще могут быть готовы попытаться предложить социально идеальные решения. Людей следует поощрять к тому, чтобы они отвечали на вопросы как можно проще, чтобы вы могли избежать того, чтобы исследователи делали фальшивые выводы из их взгляда на [Alzheimer Europe, 2018].

Система распознавания лиц может быть установлена в любом типе организаций, компаний или даже для индивидуального использования. Следователь проведет беседу с управляющим одной из южных квартир, который отвечает за безопасность и видеонаблюдение. Это позволит добиться высокой скорости отклика и прояснить любые неясности.

4.2 Дизайн

Вопросы, подготовленные для анкеты, и цель каждого вопроса представлены в таблице ниже:

#	Question	Objective
---	----------	-----------

1	Какие технологии вы считаете наиболее популярными среди различных приложений биометрии?	Чтобы определить технологию, которую люди часто встречают
2	Вы знакомы с технологией распознавания лиц?	Чтобы определить насколько хорошо люди знакомы с технологиями распознавания лиц.
3	Какие биометрические технологии вы используете или в них участвовали?	Определить наиболее широко используемую технологию в настоящее время
4	Верите ли вы в эффективность технологий распознавания лиц при идентификации личности?	Определить уровень доверия к технологиям распознавания лиц.
5	Некоторые лица обманным путем подделывают личность других лиц с целью совершения незаконных действий. Вы когда-нибудь читали или слышали о людях, которые делают это в любой из следующих ситуаций?	Определить количество мошенничеств, совершенных людьми, использующими чужую личность
6	Какая технология, по вашему мнению, в большей степени способствует будущему биометрии?	Определить технологию, которая будет использоваться в будущем
7	Какую информацию вы ожидаете узнать о распознавании лиц?	Определить пробелы в знаниях людей о технологии распознавания лиц.
8	Видите ли вы, что распознавание лиц заменит использование карт доступа в зонах безопасности в течение следующих 5 лет?	Определить вероятность замены традиционной аутентификации в будущем

9	Технология распознавания лиц является более безопасной, чем традиционные методы	Чтобы определить предпочтения пользователя с точки зрения безопасности
10	Готовы ли вы использовать технологии распознавания лиц для защиты конфиденциальной информации в моей организации?	Определить точку зрения пользователя на эффективность распознавания лиц в своей компании

Таблица 4.1: Вопросы и ее цели

Вопросы подготовленные для интервью с менеджером One South Apartment и цель каждого вопроса представлены в таблице ниже:

#	Question	Objective
1	С какой проблемой вы столкнулись из-за использования аутентификации по карте доступа?	Для определения вопроса контроля карты доступа
2	Сколько времени требуется, чтобы зарегистрировать нового арендатора и предоставить ему карту доступа?	Чтобы определить время, которое пользователи должны зарегистрировать карту доступа
3	Не хотите ли вы внести какие-либо улучшения в свою аутентификацию?	Чтобы определить их планы или предложения по улучшению их технологии доступа. Думают ли они о биометрии?
4	Вы знакомы с биометрическими технологиями?	Определить свои знания в области биометрических технологий
5	Как вы думаете, распознавание лиц может заменить текущую	Определить их предположение о замене действующей системы

	проверку подлинности безопасности?	
6	Сколько затрат вам нужно для того, чтобы заменить существующую систему распознавания лиц?	Определить расходы по рассрочке признания лица

Таблица 4.2: вопросы задаваемые на собеседованиях и его цели

4.3 Выводы по главе 4

Методы, которые разработчик будет использовать для сбора данных, - это интервью и анкетирование. Собеседование позволит получить более подробную информацию о проекте и даст некоторые указания по протеканию процессов. В то время как анкета обеспечит эффективное, быстрое и относительно дешевое получение необходимого и большого количества информации от многих людей. Быстрота оправдывается тем, что исследователь не будет присутствовать при заполнении анкеты. Такой метод, если он полезен для больших групп населения во время интервью, был бы непрактичен.

Однако проблема с анкетированием, которая может возникнуть, заключается в том, что респонденты могут сказать неправду из-за социальной желательности. Большинство людей хотят представить позитивный образ самих себя и поэтому могут лгать или искажать правду, чтобы выглядеть хорошо, например, ученики будут преувеличивать продолжительность пересмотра.

5 ПРОВЕРКА ТРЕБОВАНИЙ

5.1 Анализ данных

5.1.1 Рассылка вопросников

Всего в анкете, разработанной в Google Forms, было представлено 8 вопросов. В этом сборе дат приняли участие 89 человек. Анализ данных основан на круговых диаграммах ответов на каждый вопрос следующим образом:

1. Вы знакомы с технологией распознавания лиц?

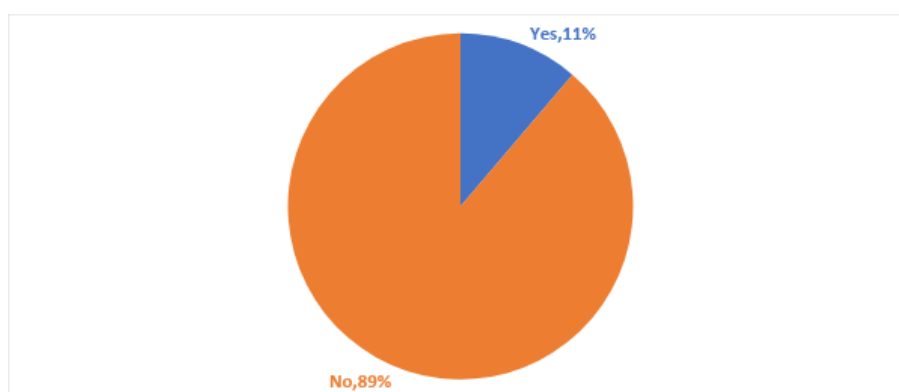


Рисунок 5.1.

Как видно из рисунка 5.1, большинство людей (89%) не знакомы с системой распознавания лиц, что подчеркивает, что эта технология еще не была хорошо адаптирована к нашей повседневной среде.

2. Какие технологии вы считаете наиболее популярными среди различных приложений биометрии?

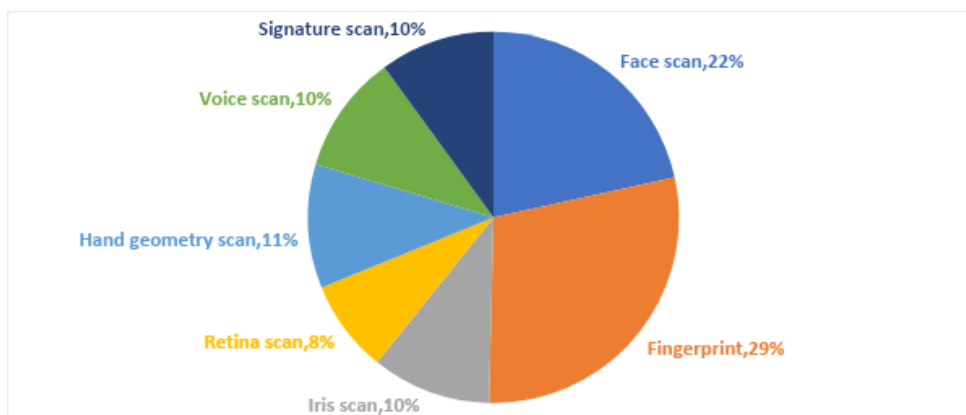


Рисунок 5.2

Из приведенной выше диаграммы видно, что люди больше занимаются биометрией отпечатков пальцев. Вероятно, это связано с тем, что почти все современные смартфоны используют эту функцию. И эта технология наиболее хорошо изучена по сравнению с другими. Распознавание лиц, поскольку самые инновационные компании, такие как Apple и Samsung, начали включать эту функцию в свои недавно выпущенные смартфоны, становится все более популярным.

3. Какие биометрические технологии вы используете или в них участвовали?

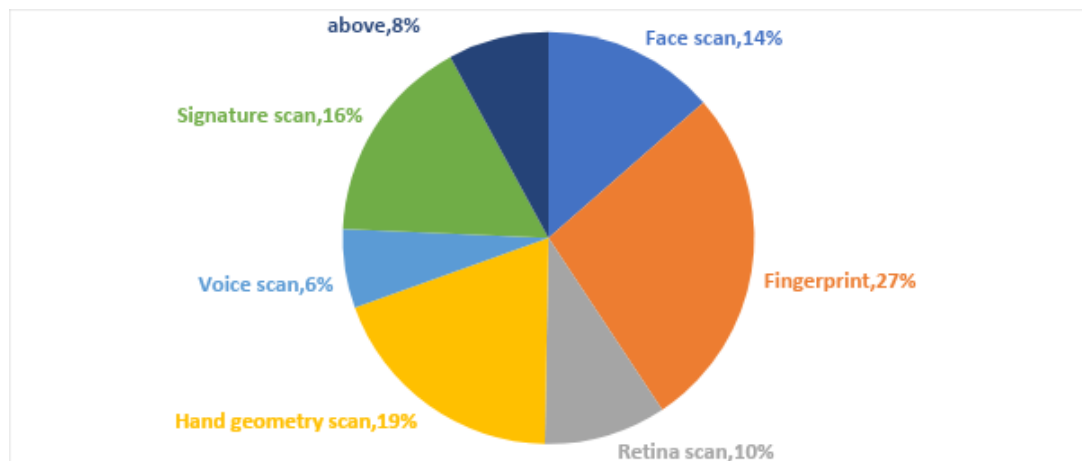


Рисунок 5.3

Рис.5.3. доказано то, что было сказано в предыдущем вопросе.

4. Верите ли вы в эффективность технологий распознавания лиц при идентификации личности?

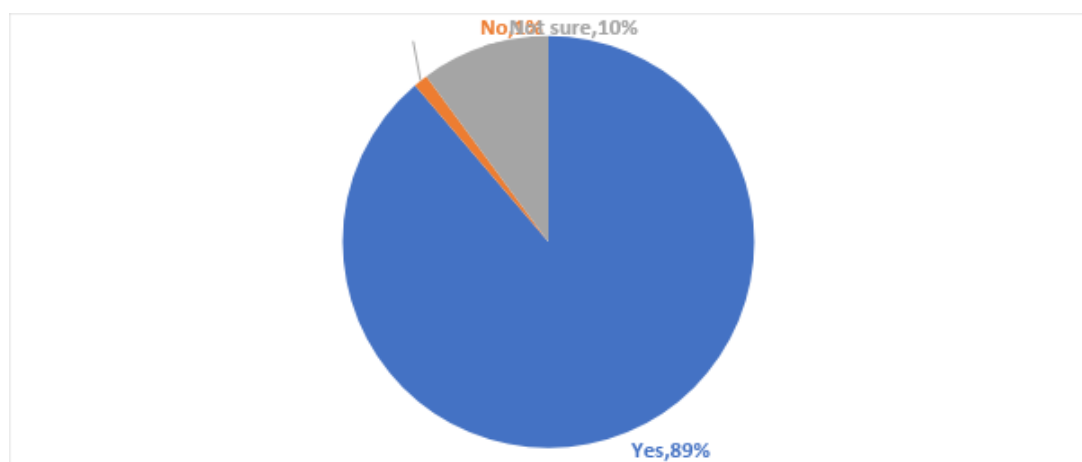


Рисунок 5.4

В представлении большинства людей технология биометрии получила потенциальные преимущества в будущем для их личной идентификации.

5. Некоторые лица обманным путем подделывают личность других лиц с целью совершения незаконных действий. Вы когда-нибудь читали или слышали о людях, которые делают это в любой из следующих ситуаций?

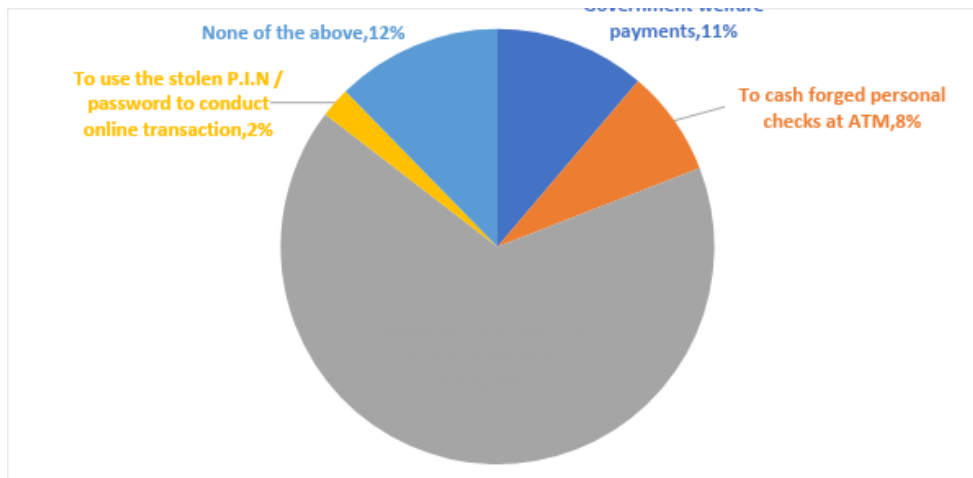


Рисунок 5.5

Как уже упоминал разработчик, проект направлен на разработку системы безопасности, когда физические лица не смогут делиться своими данными доступа, что согласно рис.5.5. это обычная практика, которую

6. Какая технология, по вашему мнению, в большей степени способствует будущему биометрии?

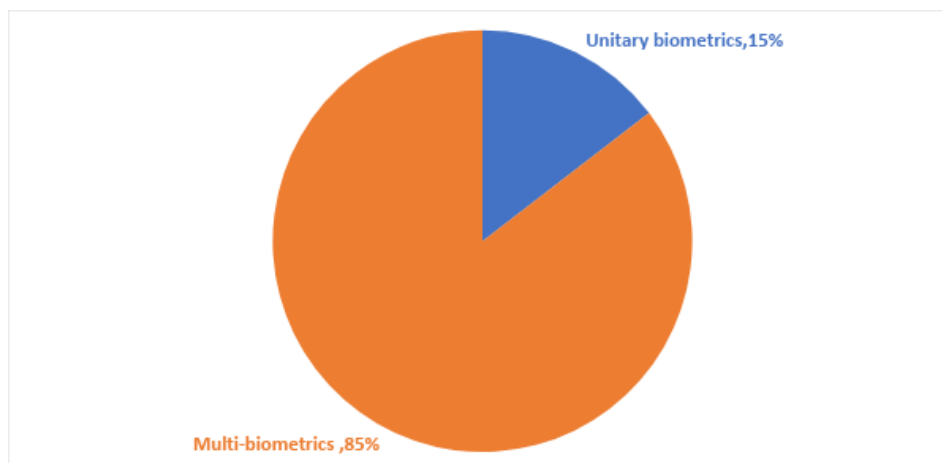


Рисунок 5.6

Рис. 5.6. покажите, что люди видят будущее биометрии, которая будет использоваться в сотрудничестве с другой, что повысит точность верификации.

7. Какую информацию вы ожидаете узнать о распознавании лиц?

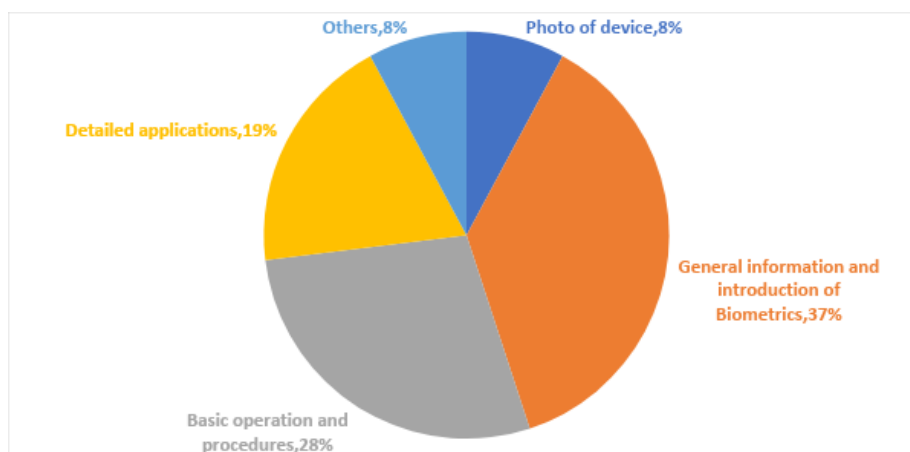


Рисунок 5.7

На приведенной выше диаграмме показаны пробелы в знаниях наших пользователей. И это говорит о том, что они не очень хорошо знакомы с общей информацией о технологии биометрии и процессе ее эксплуатации.

8. Видите ли вы, что распознавание лиц заменит использование карт доступа в зонах безопасности в течение следующих 5 лет?

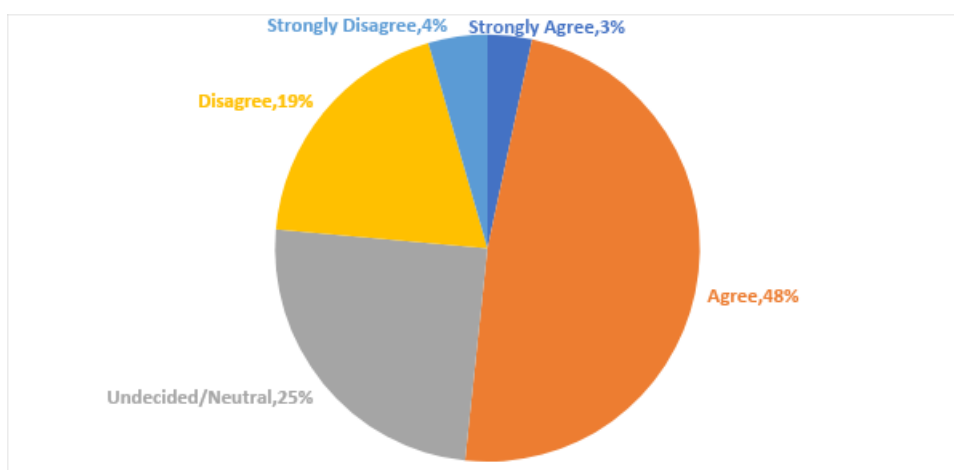


Рисунок 5.8

Рис. 5.8. это говорит о том, что 51% людей думают, что проект, который стремится сделать разработчик, будет иметь спрос в ближайшем будущем, который может заменить традиционную идентификацию профиля, которая была взята через карты доступа.

9. Технология распознавания лиц более безопасна, чем традиционные методы?

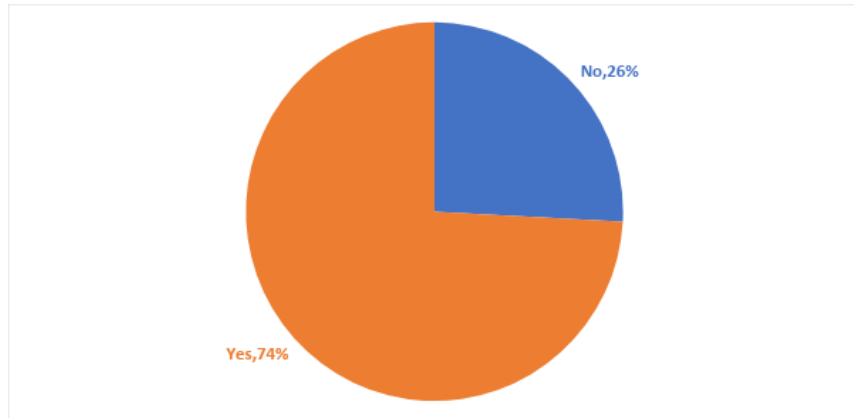


Рисунок 5.9

Диаграмма ниже показывает, что 74% толпы видят распознавание лиц более защищенным подходом, по сравнению с традиционным методом.

10. Готовы ли вы использовать технологии распознавания лиц для защиты конфиденциальной информации в моей организации?

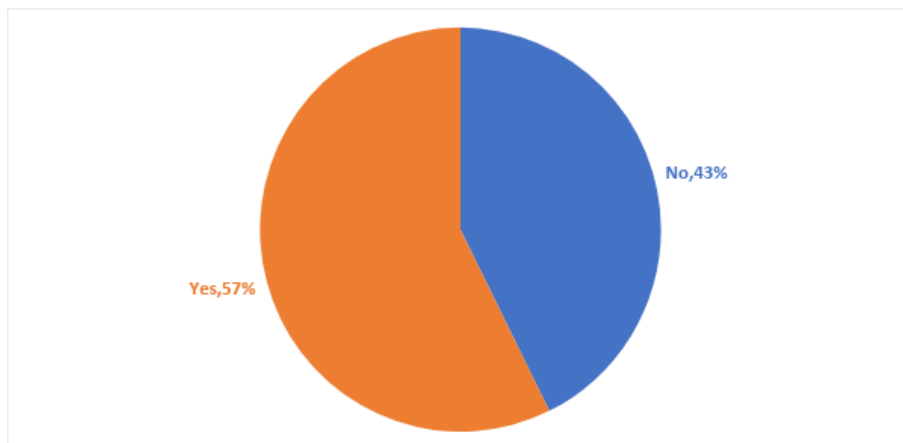


Рисунок 5.10

На рис. 5.10 показано, что большинство людей (57%) установило бы систему распознавания лиц на своем рабочем месте.

5.1.2 Интервью

Как видно из рисунка 5.5 анкеты, основная проблема безопасности заключается в том, что несанкционированные люди получают доступ к охраняемой зоне, используя удостоверения других людей. Во время интервью

интервьюер также доказал этот факт. И каждые два месяца они проверяют карточки жильцов, когда они входят в квартиру. На каждой итерации они идентифицируют, что минимум 20 человек использовали карты доступа другого человека. Как я уже упоминал 1.4.1.1. одним из преимуществ системы распознавания лиц является время и стоимость. В интервью говорилось, что для того, чтобы зарегистрировать новую карту доступа, арендатор должен подождать минимум 3 дня и заплатить 20 долларов. Этот вид вопроса на самом деле очень неудобен, даже кажется, что 3 дня-это короткий срок, а 20 долларов это дешево. В то время как регистрация в системе распознавания лиц может занять несколько минут. Еще одна вещь, которую я замечаю во время собеседования, если арендатор теряет свою карту доступа, он должен будет сделать регистрацию снова. Но в системе распознавания лиц вы можете зарегистрироваться только один раз, и ваша личность не будет потеряна. Также важно отметить, что когда следователь спросил о замене существующей системы распознавания лиц аутентификацией, менеджер не смог дать удовлетворительный ответ, из-за того, что люди не очень хорошо знакомы с технологией распознавания лиц [рис.5.1].

5.2 Вывод по главе 5

Учитывая тот факт, что многие люди еще не знакомы с распознаванием лиц, текущий проект предоставит интуитивно понятный пример того, как это может быть сделано, чтобы продемонстрировать, насколько хорошо он может быть принят в реальной рабочей среде.

Проанализировав все найденные данные анкетного исследования, которое было применено, следователь принимает решение о переподготовке исходных требований. Причина в том, что собранная информация была идентична ожиданиям.

6 СИСТЕМНАЯ АРХИТЕКТУРА

6.1. Вступление

6.1.1. ИНТЕРФЕЙС ПРИКЛАДНОГО ПРОГРАММИРОВАНИЯ

Первым шагом проекта была разработка веб-сервиса RESTful API, который сделает эту систему доступной для любого устройства через HTTP-методологии, определенные протоколом RFC 2616. Он разбивает транзакцию, чтобы создать серию различных модулей. Каждый модуль обращается к определенной базовой части транзакции. Эта модульность обеспечивает разработчикам большую гибкость. Доступные запросы: вставка, удаление, изменение, распознавание.

6.1.2. Веб-приложение

Когда back-end API будет готов, мы сможем перейти к созданию лицевой стороны приложения. Будет полезно отслеживать журнал работы системы и выполнять операции с использованием графического интерфейса вместо команд bash.

6.1.3. Имитатор ворот

Имитатор ворот будет создан микроконтроллером Arduino. Мы должны создать сценарий, который будет записывать видео перед воротами. После того, как он обнаружил лицо, кадр этой записи должен быть отправлен на веб-сервер для распознавания. Ответ будет возвращен в виде текста JSON. Если сервер вернул какое-то имя пользователя, то оно должно быть выведено на экран ворот и открыть вход, отправив определенную команду на микроконтроллер, оснащенный сервоприводом.

6.2 Абстрактная архитектура

6.2.1 Проектирование системы

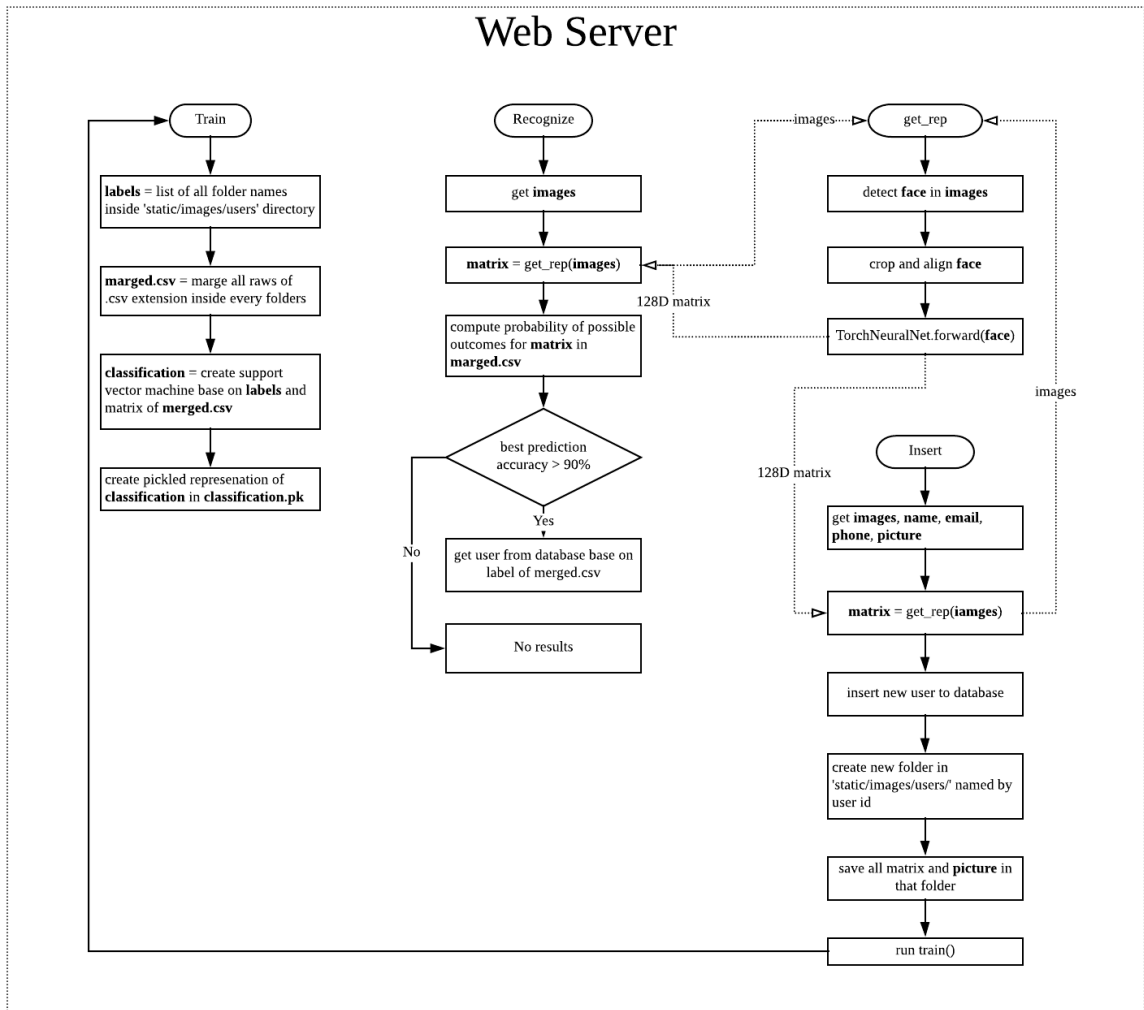


Рис. 6.1: блок-схема веб-сервера

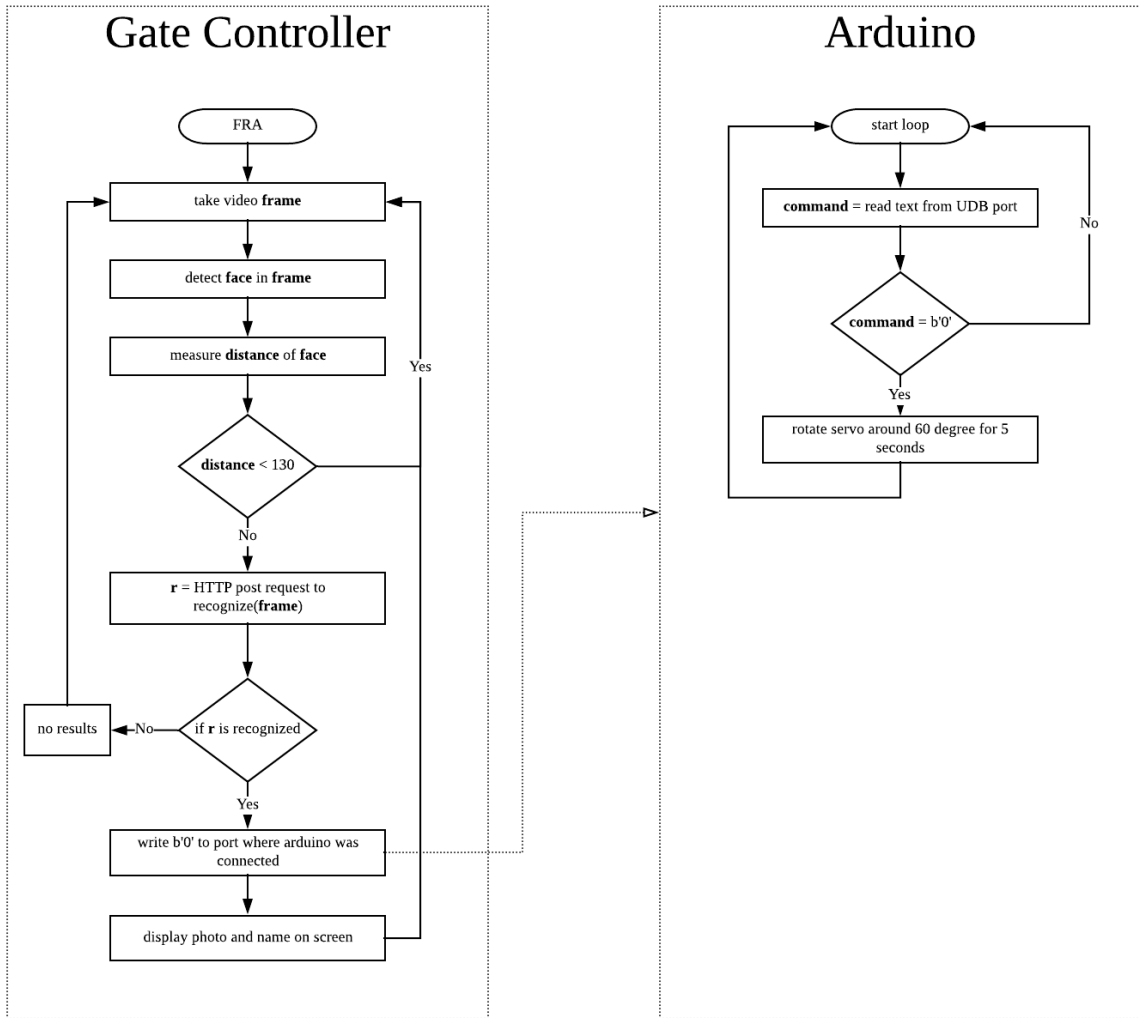


Рис. 6.2: контроллер затвора и блок-схема Arduino

6.2.2. Проектирование баз данных

6.2.2.1. Диаграмма отношений сущностей

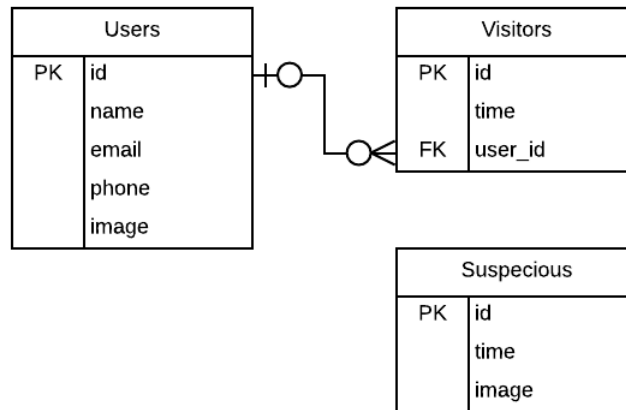


Рисунок 6.3. Database

6.2.2.2. Структура таблицы базы данных

6.2.2.2.1. Пользователи

Table: Users				
Column name	Primary key	Datatype	Not null	Notes
id	YES	Integer	YES	-
name	NO	String	YES	-
email	NO	String	YES	-
phone	NO	String	YES	-
image	NO	String	YES	Only extension of image is necessary, since rest of path is known by default

Таблица 6.1: таблица пользователей

```

9 class Users(Base):
10     __tablename__ = "users"
11
12     id = Column(Integer, primary_key=True, autoincrement=True)
13     name = Column(String(250), nullable=False)
14     email = Column(String(250), nullable=False)
15     phone = Column(String(250), nullable=False)
16     image = Column(String(250), nullable=False)
17
18     @property
19     def serialize(self):
20         return {
21             'id': self.id,
22             'name': self.name,
23             'email': self.email,
24             'phone': self.phone,
25             'image': self.image
26         }
27

```

Рисунок 6.4: таблица пользователей

6.2.2.2.2. Visitors

Table: Visitors				
Column name	Primary key	Datatype	Not null	Notes
id	YES	Integer	YES	-
time	NO	String	YES	Time when user came in
user_id	NO	String	YES	Foreign key. Related to id column of user table
user	NO	String	YES	-

Таблица 6.2: таблица посетителей

```

28
29 class Visitors(Base):
30     __tablename__ = "visitors"
31
32     id = Column(Integer, primary_key=True, autoincrement=True)
33     time = Column(DateTime, default=func.now())
34     user_id = Column(Integer, ForeignKey('users.id'))
35     user = relationship(Users)
36
37     @property
38     def serialize(self):
39         return {
40             'id': self.id,
41             'time': self.time,
42             'user_id': self.user_id,
43             'user': self.user
44         }
45

```

Рисунок 6.5: таблица посетителей

6.2.2.2.3. Подозреваемые

Table: Suspicious				
Column name	Primary key	Datatype	Not null	Notes
id	YES	Integer	YES	-
time	NO	String	YES	-
image	NO	String	YES	-

Таблица 3.1: подозрительная таблица

```
46
47 class Suspicious(Base):
48     __tablename__ = "suspicious"
49
50     id = Column(Integer, primary_key=True, autoincrement=True)
51     time = Column(DateTime, default=func.now())
52     image = Column(String(250), nullable=False)
53
54     @property
55     def serialize(self):
56         return {
57             'id': self.id,
58             'time': self.time,
59             'image': self.image
60         }
61
```

Рис. 6.6: подозрительная таблица

6.2.3. Дизайн интерфейса

6.2.3.1. Контроллер ворот

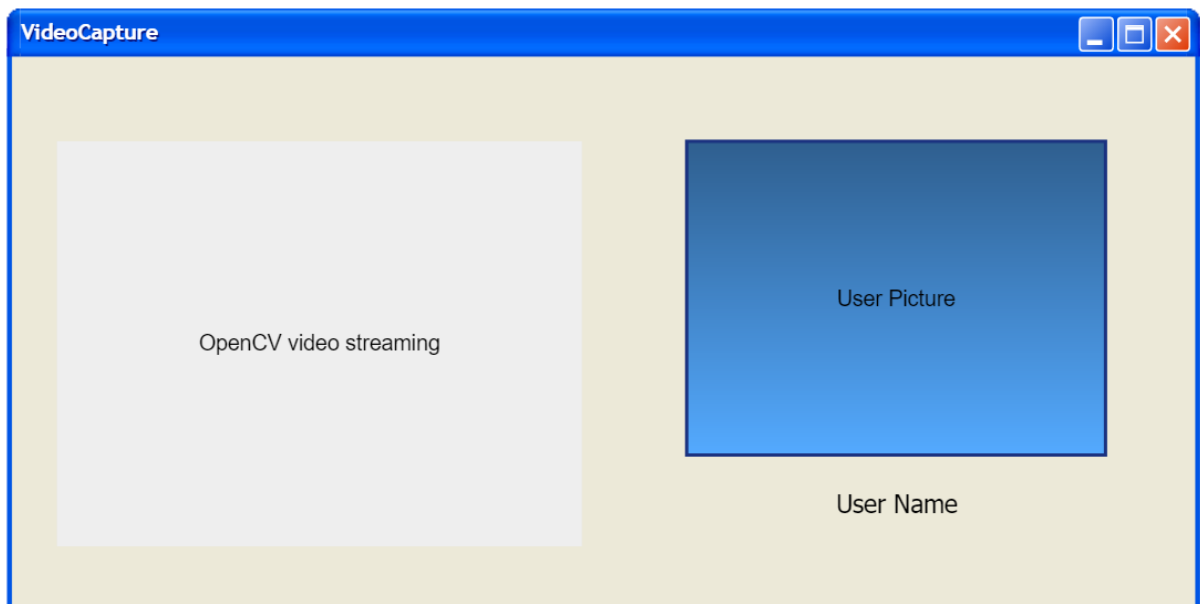


Рис.6.4. Графический интерфейс контроллера ворот

6.2.3.2. *Страница входа*

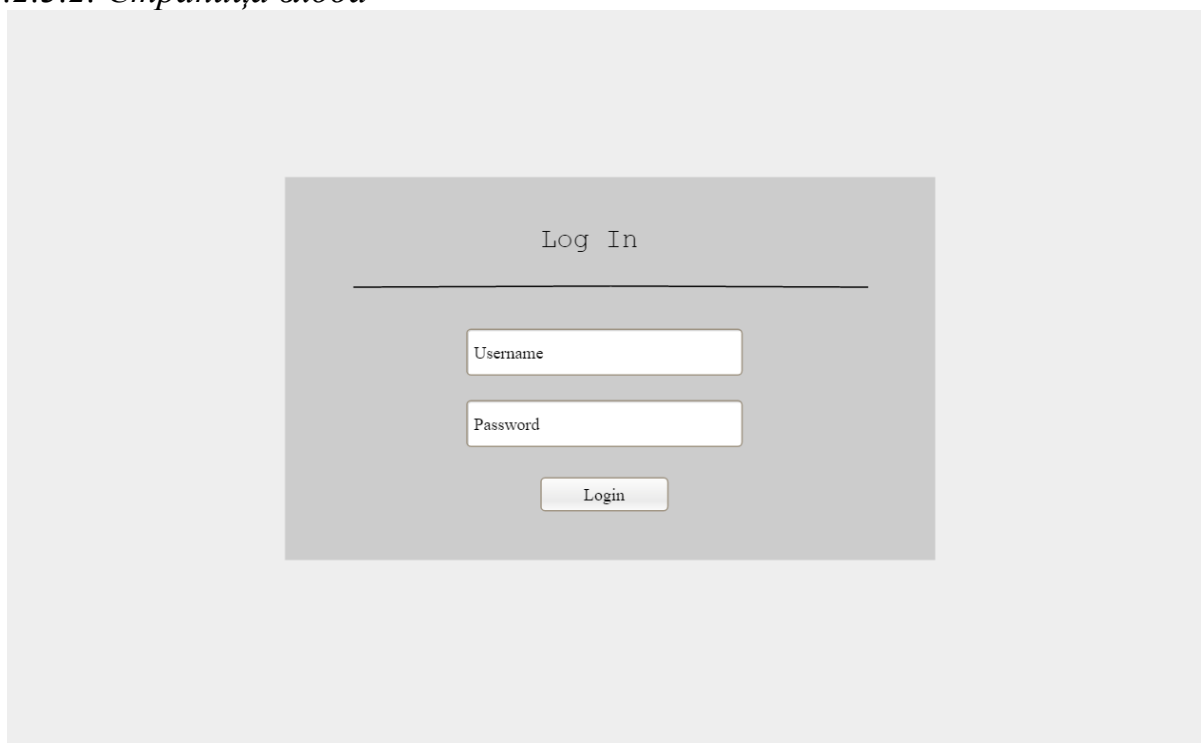


Рисунок 6.5: страница входа в систему

6.2.3.3. *Страница регистрации пользователя*

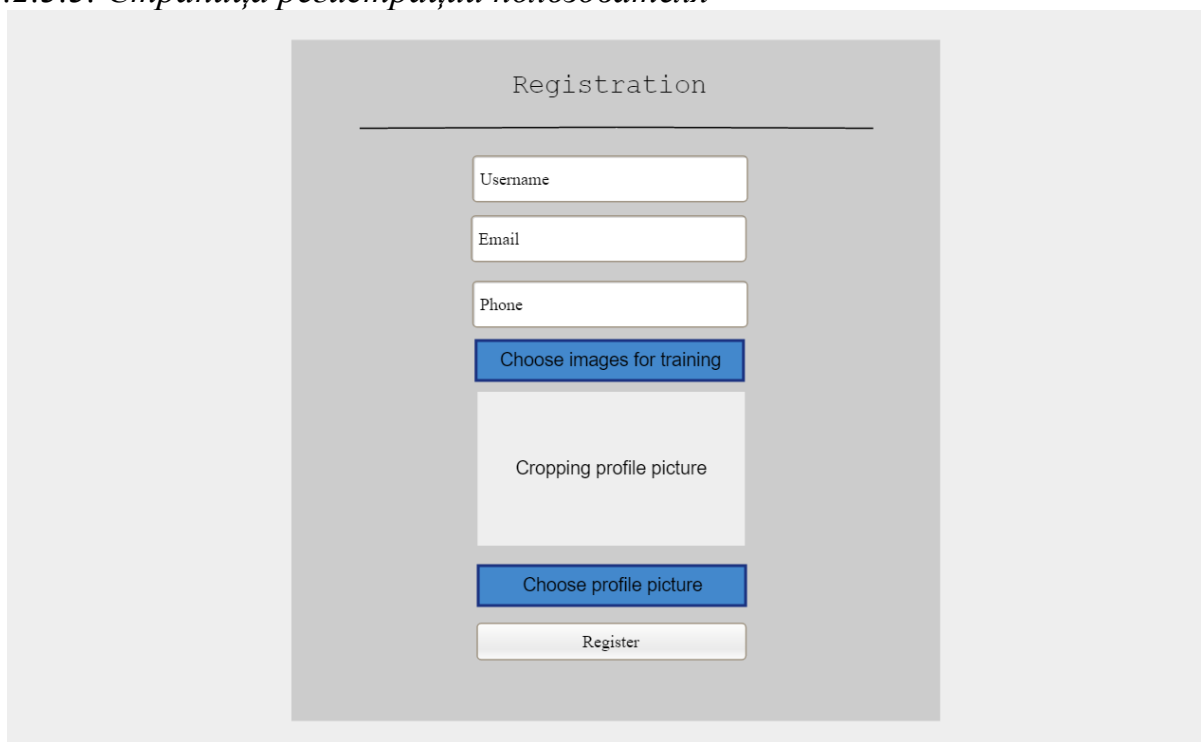


Рисунок 6.6: Страница регистрации

6.2.3.4. Пользователи / посетители / подозрительные страницы

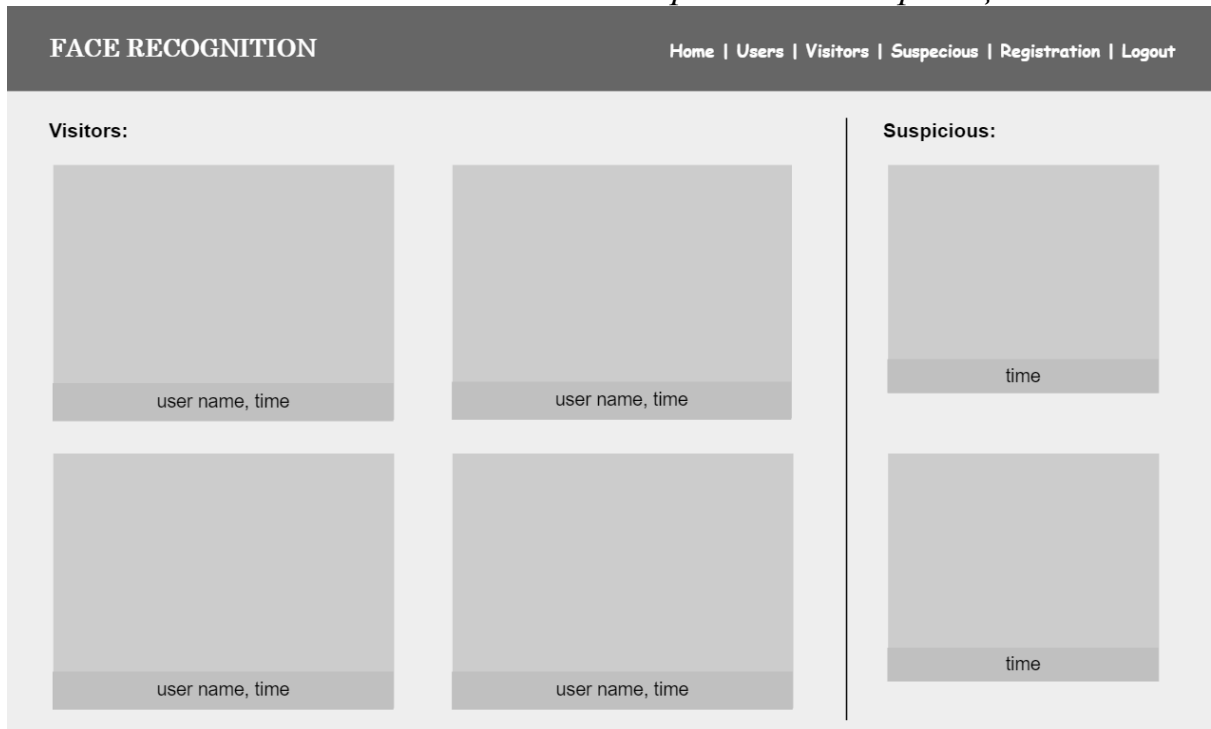


Рисунок 6.7: Главная страница

6.2.3.5. Страница изменения сведений о пользователе

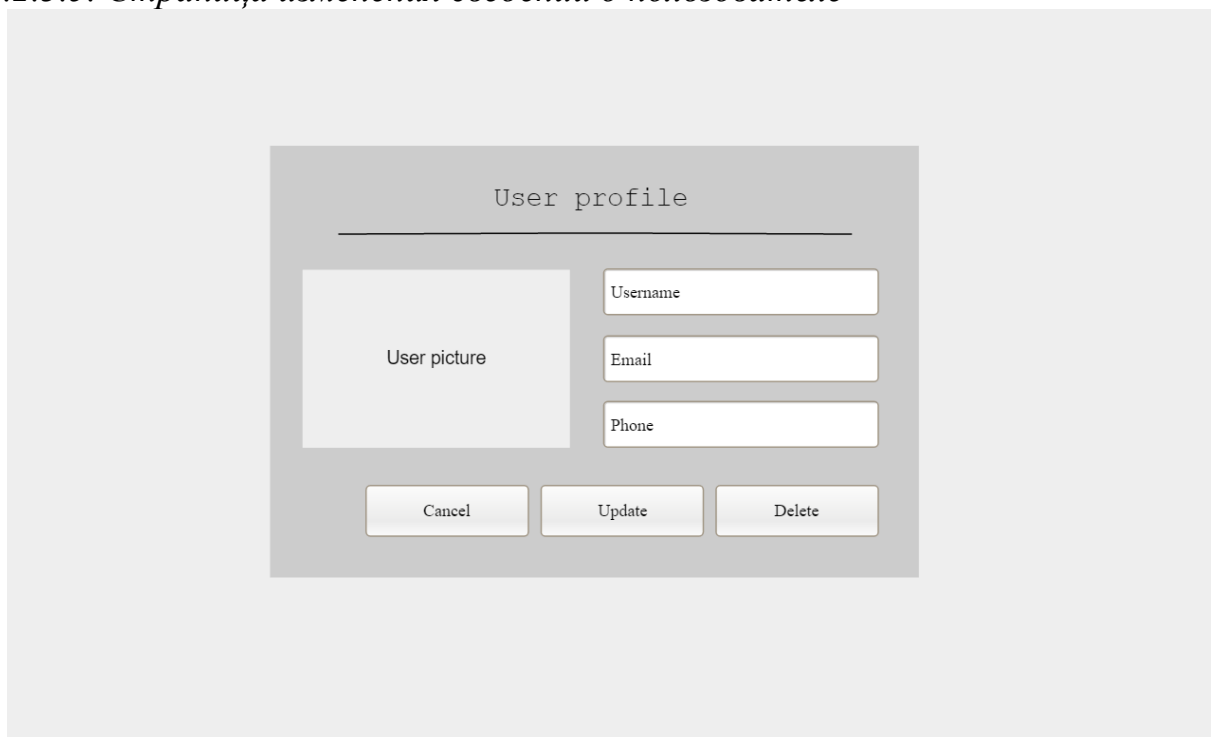


Рисунок 6.8: сведения о пользователе

7 ПЛАН ПРОЕКТА

7.1. Подробности релиза

7.1.1. Версия I

Система должна быть оснащена множеством инструментов и зависимостей. На начальном этапе разработки программного обеспечения все необходимые инструменты устанавливаются в среду, в которой будет работать система. Все тонкости окружающей среды должны быть удовлетворены потребностями окружающей среды.

7.1.2. Версия II

Вторая версия системы состоит из структурированной базы данных SQLite build in object-relational map, называемой SQLAlchemy, и шаблона для server foundation, который полностью выполнен в Flask, со всеми будущими маршрутами с его методами HTTP-запросов.

7.1.3. Версия III

Система способна зарегистрировать нового пользователя и распознать его по заданной фотографии. В качестве ответа эти две операции возвращают информацию в формате JSON.

7.1.4. Версия IV

В этой версии система позволяет администратору посещать веб-сайт и отслеживать базу данных в режиме онлайн. Кроме того, он позволяет администратору изменять или удалять данные пользователя и регистрировать новых пользователей.

7.1.5. Версия V

Когда веб-сервер будет полностью реализован, следующим этапом будет разработка такого устройства, которое будет прослушивать входящий поток USB-порта с ПК. Как только он получает определенную команду, он должен открыть ворота и закрыть их через несколько секунд.

7.1.6. Версия VI

Окончательная версия будет состоять из настольного приложения, которое будет захватывать видеопоток с камеры. Когда лицо было обнаружено в этом потоке, он отправляет изображение кадра на веб-сервер для распознавания. Если лицо на этом изображении было распознано, приложение посылает специальную команду на устройство через USB, чтобы открыть вход.

7.2. План тестирования

7.2.1. Модульный тест

№	Цель	Ввод	Ожидаемый результат
1	Распознавание лиц	Куча человеческих образов	Нарисуйте прямоугольники вокруг всех граней
2	Распознавание лиц	Куча изображений без людей в нем	Дисплей 0 количество обнаруженных лиц
3	Выравнивание лица	Обрезанные изображения лиц	Выровненные изображения лиц
4	Классификация	Куча картинок	Дисплей 128d матрица для каждой грани
5	Распознавание	Последовательность обучение фотографии пользователя	Отображение идентификатора, имени и точности
6	Распознавание	Последовательность неизвестных человеческих образов	Отображение ' Null'
7	Контроллер ворот	Прямая трансляция в передней части ворота	Нарисуйте прямоугольник, если человек может приблизиться к 1 метру. И сделайте звуковое уведомление, если лицо было распознано или нет.
8	Микроконтроллер Arduino	Последовательность команд через USB-порт	Поверните сервопривод примерно на 60 градусов для определенной команды
9	Веб-приложение	Получить запрос на главную страницу	Два ряда пользователей, которые были признаны, а кто нет

1 0	Регистрация	Пользовательские данные	Отображение идентификатора пользователя, количества изображений и лица было обнаружено
1 1	Обновление Удаление	/ Изменение пользовательских данных	Измененные пользовательские данные

Таблица 7.1: модульное тестирование

7.2.2. Приемочные испытания пользователей

Чтобы проверить принятие пользователей, я возьму несколько пользователей. Некоторые из них уже будут зарегистрированы в моей системе. Все они, стоя в очереди перед входом, должны пройти через ворота. В конце я сделаю вывод в процентах, сколько правильных и неправильных пользователей было признано системой.

8. РЕАЛИЗАЦИЯ

8.1. Скриншоты

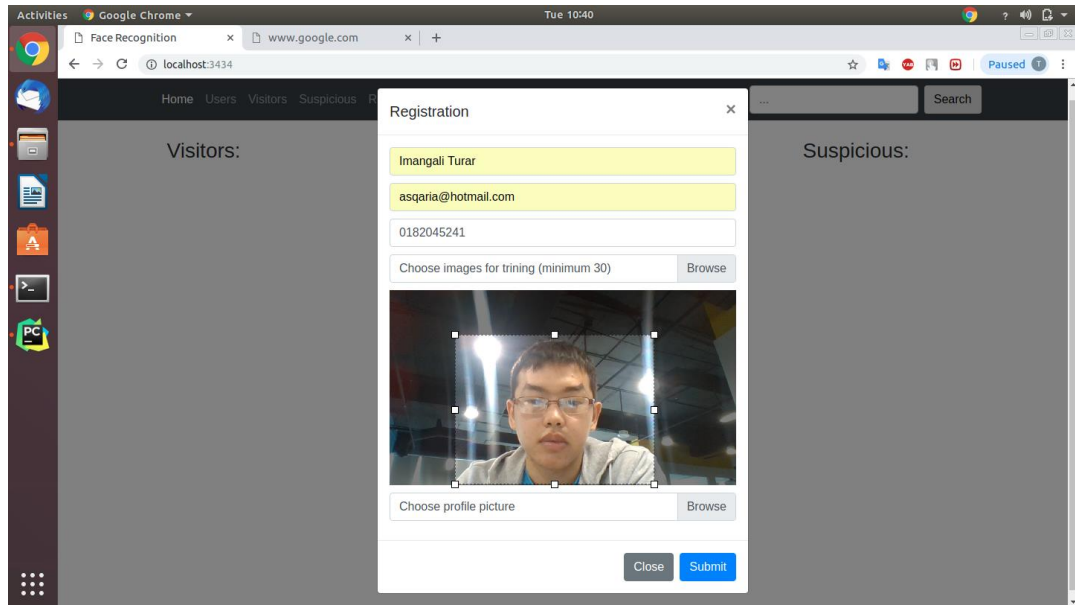


Рисунок 7.1: Регистрация

Регистрация позволяет добавить нового пользователя в базу данных и обучить его. Для того, чтобы сделать правильное распознавание позже, минимальное количество изображений, которые будут загружены, составляет около 30.

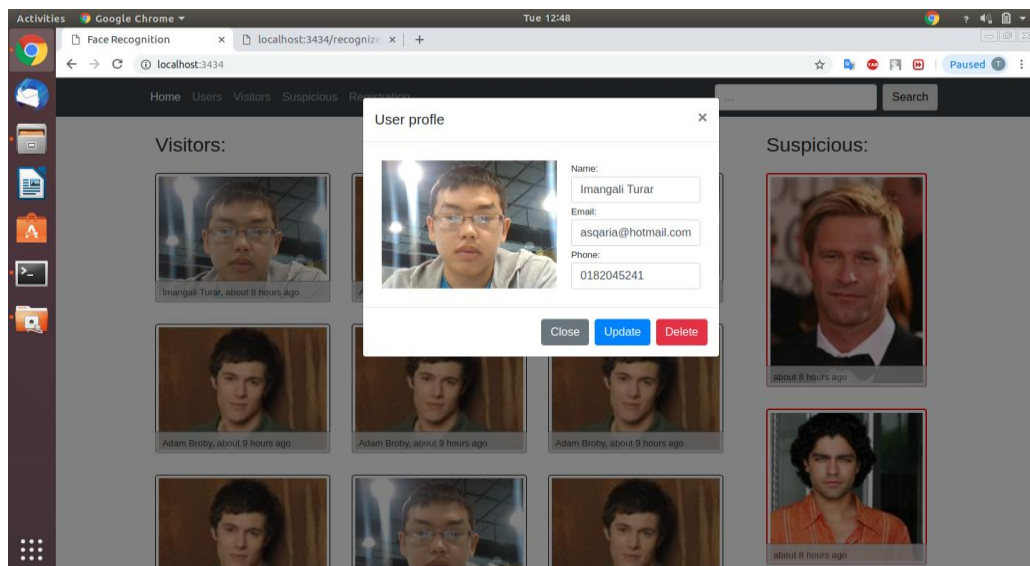


Рисунок 7.2: Модификация

Модификация позволяет администратору вносить изменения в сведения о пользователях.

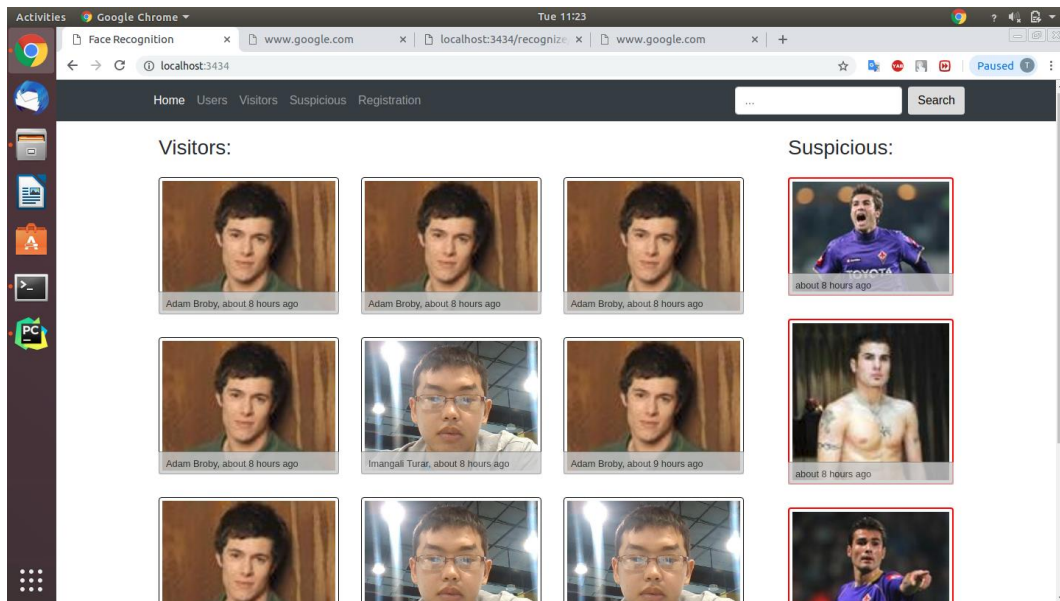


Рисунок 7.3: Главная страница

На главной странице администратор может просматривать и отслеживать посетителей и людей, которые не были идентифицированы системой.

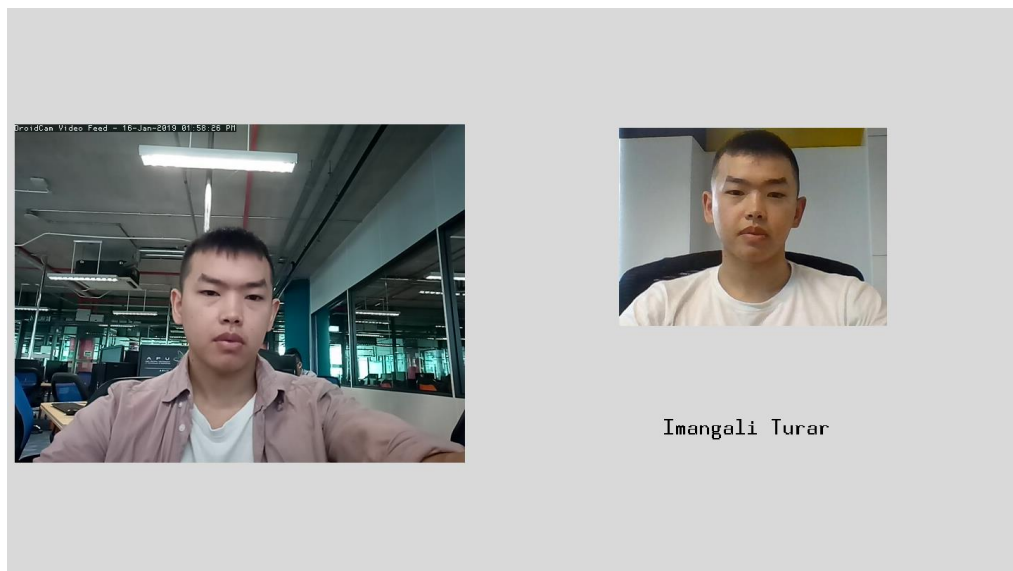


Рис. 7.4: Контроллер Ворот

Это окно представлено на переднем мониторе ворот. Он отображает текущий видеопоток, и если переданный человек идентифицирован, он покажет его/ее фотографию профиля и имя.

8.2. Пример кода

8.2.1. Обучение

```
10 def train(dir, workdir):
11     labels = list()
12     merged_csv_path = os.path.join(workdir, 'merged.csv')
13     merged = open(merged_csv_path, 'w')
14
15     for filename in sorted(os.listdir(dir)):
16         new_dir = os.path.join(dir, filename)
17         for col in sorted(os.listdir(new_dir)):
18             if col.endswith('.csv'):
19                 labels.append(filename)
20                 fullpath = os.path.join(dir, filename, col)
21                 f1 = open(fullpath)
22                 for line in f1:
23                     merged.write(line)
24                 f1.close()
25     merged.close()
26
27     le = LabelEncoder().fit(labels)
28     labels_num = le.transform(labels)
29     embeddings = pd.read_csv(merged_csv_path, header=None).as_matrix()
30
31     clf = SVC(C=2, kernel='linear', probability=True)
32     clf.fit(embeddings, labels_num)
33
34     fname = "[%s]classifer.pkl" % os.path.basename(workdir)
35     print("Saving classifier to [%s]" % os.path.join(workdir, fname))
36     with open(fname, 'w') as f:
37         pickle.dump((le, clf), f)
38
39     with open('classifier', 'w') as f:
40         if sys.version_info[0] < 3:
41             (le, clf) = pickle.load(f)
42         else:
43             (le, clf) = pickle.load(f, encoding='latin1')
```

Рис. 7.5: функция train()

Этот метод будет принимать каталог выходных матриц и каталог с функциями, необходимыми для выполнения обучения. Все матрицы с его метками будут объединены в один файл. При извлечении меток и матриц пользователей программа будет выполнять модели машинного обучения с опорным вектором и связанные с ними алгоритмы обучения, которые анализируют данные, используемые для классификации граней. Когда он наконец обучится через SVM, он сохранит файл в виде двоичных протоколов для сериализации и десериализации структуры объекта Python с помощью модулей pickle.

8.2.2. Распознавание

```
120
121 @app.route("/recognize/", methods=['POST'])
122 def recognize():
123     images = request.files.getlist("images")
124     total_tasks, matrices = get_req(images)
125
126     labels = list()
127     names = list()
128     confidences = list()
129
130     for m in matrices:
131         predictions = clf.predict_proba(m).ravel()
132         best = np.argmax(predictions)
133
134         id = int(le.inverse_transform(best).decode('utf-8'))
135         user = session.query(users).filter_by(id=id).one()
136         labels.append(id)
137         names.append(user.name)
138         confidences.append(predictions[best])
139
140     if len(confidences) > 0 and max(confidences) > THRESHOLD:
141         best_idx = confidences.index(max(confidences))
142
143         user = session.query(users).filter_by(id=labels[best_idx]).one()
144         visitor = visitors(visitor_id=user.id, user=user)
145         session.add(visitor)
146         session.commit()
147         return jsonify({
148             'id': labels[best_idx],
149             'name': names[best_idx],
150             'confidence': confidences[best_idx]
151         })
152     else:
153         pic = Image.open(images[0].stream)
154         extension = images[0].filename.split('.')[1]
155         suspicious = suspicious_images+extension
156         session.add(suspicious)
157         session.commit()
158         path = "static/images/suspicious/%s.%s" % (suspicious.id, extension)
159         pic.save(path)
160         return jsonify({
161             'id': "null",
162             'name': "null",
163             'confidence': "null"
164         })
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Рис. 7.6: функция распознавания()

Этот фрагмент кода ожидает изображение, отправленное по почтовому запросу из протокола HTTP. Изображение будет обработано с помощью функции `get_req()` для преобразования его грани в 128-мерную матрицу. Эта матрица будет передана склеарну `SVM.SVC.predict_proba()` для того, чтобы получить вероятность возможных исходов. Если лучшее значение прогноза выше 90%, то в результате будет получена карта `jsonify`.

8.2.3. Регистрация

```
190 @app.route('/insert/', methods=['POST'])
191 def insert():
192     name = request.form['name']
193     email = request.form['email']
194     phone = request.form['phone']
195     picture = request.files.get('picture')
196     images = request.files.getlist('images')
197
198     if name == '' or email == '' or phone == '' or len(images) < MIN_IMG or not picture:
199         return "You are required to fill up all format!"
200
201     total_images = len(images)
202     total_faces, matrices = get_rep(images)
203
204     if total_faces < MIN_IMG or total_images < MIN_IMG:
205         return "Required minimum 3d face images per user: %d MIN_IMG"
206
207     if picture and allowed_file(picture.filename):
208         extension = picture.filename.split('.')[1]
209         user = User(name=name, email=email, phone=phone, images=extension)
210         session.add(user)
211         session.commit()
212         filename = 'picture.%s' % extension
213         path = os.path.join('static/images/users/%s' % user.id, filename)
214         pic_path = os.path.join(path, filename)
215         os.mkdir(path)
216         picture.save(pic_path)
217
218         # crop image
219         x = float(request.form['x'])
220         y = float(request.form['y'])
221         w = float(request.form['w'])
222         h = float(request.form['h'])
223         coords = (x, y, w, y+h)
224         image_obj = Image.open(pic_path)
225         cropped_image = image_obj.crop(coords)
226         cropped_image.save(pic_path)
227
228         # create matrix for each face
229         index = 0
230         for matrix in matrices:
231             filename = '%s.csv' % index
232             path = os.path.join('static/images/users/%s' % user.id, filename)
233             with open(path, 'w') as csvFile:
234                 spamwriter = csv.writer(csvFile, delimiter=',', quotechar='"', quoting=csv.QUOTE_MINIMAL)
235                 spamwriter.writerow(matrix)
236             index += 1
237         else:
238             return "Error while uploading picture"
239
240     result = {
241         'id': user.id,
242         'images': total_images,
243         'detected_faces': total_faces
244     }
245
246     # update class/learn
247     train(dir='static/images/users', verbose='features')
248     return jsonify(result)
```

Рисунок 7.7: Регистрация

Запрос Insert POST требует иметь 5 констант: имя, адрес электронной почты, телефон, фотографию профиля и минимум 30 изображений для обучения. Система создаст специальную папку для этого пользователя с именем его идентификационного номера. В этой папке изображение изображения, и все 128d матрицы внутри .расширение csv, которое обрабатывается с помощью

функции `get_per ()`, будет сохранено. В результате он будет отвечать на общее количество обнаруженных лиц, идентификаторов пользователей и общего количества изображений в представлении `json`.

8.2.4. Контроллер ворот

```
57
58     def video_loop(self):
59         try:
60             # keep looping over frames until we are instructed to stop
61             while not self.stopEvent.is_set():
62                 # grab the frame from the video stream
63                 self.frame = self.stream.read()
64
65                 face_locations = face_recognition.face_locations(self.frame)
66                 for (top, right, bottom, left) in face_locations:
67                     self.ampl = right - left
68                     if self.ampl < 130:
69                         print('Far distance: %s' % self.ampl)
70                         continue
71                     elif self.ctn:
72                         print('Scanning')
73                         imgname = "frame.jpg"
74                         cv2.imwrite(imgname, self.frame)
75                         images = {'images': open(imgname, 'rb')}
76                         r = requests.post('http://localhost:3434/recognize/', files=images)
77
78                         j = r.json()
79                         if j['id'] == 'Null':
80                             print("Not recognized")
81                             mixer.music.load("2.wav")
82                         else:
83                             print("Welcome %s (%.2f)" % (j['name'], j['confidence']))
84                             self.name.config(text=j['name'])
85                             self.ser.write(b'0')
86                             new_img_path = '../static/images/users/%s/picture.jpg' % j['id']
87                             new_img = ImageTk.PhotoImage(Image.open(new_img_path).resize((400, 300), Image.ANTIALIAS))
88                             self.picture.itemconfig(self.img_on_canvas, image=new_img)
89                             mixer.music.load("1.wav")
90                             mixer.music.play()
91
92                             self.ctn = False
93                             self.ctn_i = 0
94
95                             cv2.rectangle(self.frame, (left, top), (right, bottom), (255, 0, 0), 1)
96
97                             if self.ctn_i > 50:
98                                 self.ctn = True
99                                 self.name.config(text="YOUR NAME HERE")
100                                 self.picture.itemconfig(self.img_on_canvas, image=self.img)
101                             else:
102                                 self.ctn_i += 1
103
104                             # need to swap channels, then convert to PIL and ImageTk format
105                             image = cv2.cvtColor(self.frame, cv2.COLOR_BGR2RGB)
106                             image = Image.fromarray(image)
107                             image = ImageTk.PhotoImage(image)
108
109                             # if the panel is not None, we need to initialize it
110                             if self.panel is None:
111                                 self.panel = tki.Label(image=image)
112                                 self.panel.image = image
113                                 self.panel.pack(side="left", padx=10, pady=10)
114
115                             # otherwise, simply update the panel
116                             else:
117                                 self.panel.configure(image=image)
118                                 self.panel.image = image
119
120         except RuntimeError, e:
121             print("[INFO] caught a RuntimeError")
```

Рис. 7.8: контроллер ворот

Контроллер ворот будет захватывать видеопоток в режиме реального времени. Он должен обнаружить все лица, которые приближаются к воротам,

примерно на 40 метров. Когда лицо будет обнаружено, кадр будет отправлен на сервер для распознавания. Ответ будет отправлен в виде текста JSON. После этого код выполнит аудио с результатами запроса, и если результат будет положительным, ворота должны быть автоматически открыты.

8.3. Последовательность установки

8.2.1. Вступление

Этот проект был разработан, установлен и протестирован в операционной системе Ubuntu. Вся предусмотренная установка будет обеспечена с помощью Ubuntu Bash. Прежде чем приступить к установке, вы должны иметь инструмент Git version control system, который можно скачать по этой ссылке: <https://en.wikipedia.org/wiki/Git>.

8.2.2. Установите анаконду

Anaconda стремится упростить управление пакетами и их развертывание. Версии пакетов управляются системой управления пакетами conda. Он может быть установлен следующими командами:

```
asqar@DESKTOP-1UR1MIO MINGW64 ~/Desktop
$ wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh; bash Miniconda3-latest-Linux-x86_64.sh
```

Для того чтобы создать новую среду в anaconda и активировать ее, выполните следующую команду:

```
asqar@DESKTOP-1UR1MIO MINGW64 ~/Desktop
$ conda create --name openface python=2.7; source activate openface
```

8.2.3. Установите необходимые модули для Python

В этом проекте используется различное количество модулей, которое доступно в репозитории conda-forge. Следующая команда добавит этот канал репозитория и установит все необходимые библиотеки в нашу среду.

```
asqar@DESKTOP-1UR1MIO MINGW64 ~/Desktop
$ conda config --add channels conda-forge; conda install opencv numpy pandas scipy scikit-learn scikit-image dlib txai twisted autobahn OpenSSL pyopenssl ima
gehash service_identity
```

8.2.4. Установите Torch

Torch-это научная вычислительная платформа с широкой поддержкой алгоритмов машинного обучения. Мы можем установить его с помощью следующих команд:

```
asqar@DESKTOP-1UR1MIO MINGW64 ~/Desktop
$ git clone https://github.com/torch/distro.git ~/torch --recursive; cd ~/torch; bash install-deps; bash install.sh
```

Как только вы установили Torch, на данный момент командная строка `th` должна быть доступна в вашей оболочке. После того как факел станет доступен, вы должны добавить к нему дополнительные расширения, такие как:

- `tds` (только для обучения DNN)
- `cutorch` и `cunn` (только с CUDA)
- `fblualib` (только для тренировки DNN)
- `torchx` (только для обучения DNN)
- `optnet` (опционально, только для обучения DNN)
- `dpnn`
- `nn`
- `optim`
- `csvigo`

Все это расширение можно установить с помощью следующей команды:

```
asqar@DESKTOP-1UR1MIO MINGW64 ~/Desktop
$ for NAME in dpnn nn optim optnet csvigo cutorch cunn fblualib torchx tds; do luarocks install $NAME; done_
```

8.2.5. Установите OpenFace

Наконец, OpenFace - это Python и Torch-реализация распознавания лиц с помощью глубоких нейронных сетей, основанная на документе CVPR 2015 FaceNet: унифицированное встраивание для распознавания лиц и кластеризации Флориана Шроффа, Дмитрия Калениченко и Джеймса Филбина из Google.

```
asqar@DESKTOP-1UR1MIO MINGW64 ~/Desktop
$ source activate openface; git clone https://github.com/cmusatyalab/openface.git ~/openface; cd openface; python setup.py install; bash ./models/get-models.sh
```

8.2.6. Установка Веб-Сервера

Весь написанный исходный код для этого проекта доступен в моей личной учетной записи GitHub и может быть установлен с помощью команды `git`:

```
asqar@DESKTOP-1UR1M1O MINGW64 ~/Desktop
$ git clone https://github.com/asqaria/Face_Recognition_API.git ~/Projects
```

8.2.7. Сконфигурируйте микроконтроллер Arduino

Для этого проекта я использовал Arduino Uno и сервопривод MG995, сервопривод должен быть подключен к микроконтроллеру через контакт 7. Исходный код для Arduino доступен в каталоге, созданном на предыдущем этапе ‘~/Projects/Face_Recognition_API/arduino.Ино.

8.2.8. Запустить сервер

Начальная точка сервера конфигурируется в webserver.py файл. Перед запуском сервера убедитесь, что среда anaconda активирована

```
asqar@DESKTOP-1UR1M1O MINGW64 ~
$ source activate openface; python ~/Projects/Face_Recognition_API/webserver.py
```

8.2.9. Запустить врата

Для того чтобы запустить ворота, вы должны подключить arduino и дать разрешение вашему компьютеру общаться с микроконтроллером через USB. Необходимая команда была записана в port.sh файл.

```
asqar@DESKTOP-1UR1M1O MINGW64 ~
$ bash port.sh; python ~/Projects/Face_Recognition_API/gate_controller/run_gate.py
```

9 ПРОВЕРКА СИСТЕМЫ

9.1. Модульный тест

№	Target	Input	Expected output	Test 1	Test 2	Test 3
1	Face detection	Bunch of human images	Draw rectangles around all faces	Rectangles appears in wrong location	As expected	As expected
2	Face detection	Bunch of images with no humans in it	Display 0 amount of face detected	As expected	As expected	As expected
3	Face alignment	Cropped face images	Aligned face images	As expected	As expected	As expected
4	Classification	Bunch of images	Display 128D matrix for each face	Cannot convert input image to numpy array	As expected	As expected
5	Recognition	Sequence of trained user images	Display id, name and accuracy	As expected	As expected	As expected
6	Recognition	Sequence of unknown human images	Display 'Null'	As expected	As expected	As expected
7	Gate controller	Video stream in front of gate	Draw rectangle if human can close up to 1 meter. And make audio notification	Video stream is very slow	Notification is appearing with some lagging	As expected

			if face was recognized or not.			
8	Arduino microcontroller	Sequence of command through USB port	Rotate servo around 60 degrees for certain command	USB port requires permission to make transaction	As expected	As expected
9	Web application	Get request to main page	Two rows of users who was recognized and who was not	As expected	As expected	As expected

Таблица 9.1: модульное тестирование

В соответствии с приведенными выше тестами были внесены следующие корректировки:

Test	Problem	Solution
1-1	Прямоугольники появляются в неправильном месте	Модуль Dlib дает вам точное расположение верхнего левого угла, ширины и высоты. Ошибка заключалась в неправильной последовательности параметров, которые я передал методу рисования прямоугольника.
4-1	Не удается преобразовать входное изображение в массив numpy	После проверки документации модуля numpy я узнал, что функция, которая преобразует изображение в массив numpy, требует потоков изображений, но не пути к изображению.
7-1	Видеопоток очень медленный	После некоторого изменения размера видеопотоков я достиг 12 кадров в секунду, что кажется разумным для потоковой передачи видео.

10-1	Отображает неверный ввод данных	При реализации я хотел, чтобы все фотографии профиля были одинакового размера, что будет выглядеть на сайте более удобно. Оказывается, мой скрипт, который обрезал изображение, передал данные в неправильном типе.
7-2	Уведомление появляется с некоторым запаздыванием	Реализован алгоритмический трюк: когда лицо было распознано, скрипт должен остановиться, чтобы обнаружить лица в течение нескольких секунд, чтобы дать человеку пройти внутрь, и воспроизвести уведомление.
8-1	USB порт требует разрешения на совершение транзакции	Написал новый скрипт, который находится в '~/.gate_controller/port.sh-который даст разрешение на отправку команд через USB-порт
11-1	После удаления накопилась ошибка при посещении главной страницы	Ошибка была накоплена из-за связи в базе данных. Когда пользователь был удален из таблицы "пользователи", он все еще остается в таблице "посетители". Чтобы решить эту проблему, нам нужно удалить его ссылки в разделе "посетители"

Таблица 9.2: журнал изменений

9.2. Приемочное тестирование пользователей

<i>Name</i>	<i>Status</i>	<i>Test 1</i>	<i>Test 2</i>	<i>Test 3</i>	<i>Test 4</i>
<i>User I</i>	Registered	OK	OK	OK	OK
<i>User II</i>	Registered	OK	WRONG	OK	WRONG
<i>User III</i>	Not registered	OK	OK	OK	OK
<i>User IV</i>	Not registered	OK	OK	OK	OK

Таблица 9.3: приемочное тестирование пользователей

Как видно из приведенной выше таблицы, коэффициент принятия пользователя II составляет 2 из 4. Причина такой низкой точности кроется в данных, которые использовались для обучения. Количество изображений для обучения пользователя II составляло всего 8, в то время как для остальных пользователей-около 30.

10 ВЫВОДЫ И РАЗМЫШЛЕНИЯ

10.1 Критическая оценка

Система была протестирована пользователями и разработчиком. После некоторых изменений и перенастройки все функциональные возможности системы работали так, как ожидалось, чтобы достичь цели проекта. Однако естественные проблемы, такие как качество видеопотока, освещение окружающей среды, могут вызвать проблемы с точностью распознавания. Это означает, что устройство ввода и окружающая среда должны быть выбраны в стратегическом месте.

Так как техника машинного обучения сама по себе имеет ограничения в точности и очень редко способна достичь до 100%. Как я уже упоминал ранее, система основана на модели, которая была обучена в сверточной нейронной сети исследователями из Google (facenet). Согласно их тестированию, его точность составляет 92%, что делает его одной из самых точных моделей с открытым исходным кодом для распознавания лиц.

Более того, когда речь заходит о машинном обучении, точность в большей степени зависит от количества и качества данных (большая часть данных является частной), а не от алгоритмов (которые доступны каждому). Данные, в которых обучался facenet, базировались на FaceScrub и casia-webface dataset, которые а именно состоят из европейских граней. Этот факт означает, что он будет влиять на точность для неевропейских людей.



Рис. 10.1: пример набора данных FaceScrub

Эта проблема решается с помощью обучающей модели в наборе данных других рас. Например, китайское правительство запустило программу распознавания лиц для жителей Китая. Он утверждает, что является самым продвинутым распознаванием лиц в мире.

Еще одна проблема, с которой сталкивается проект, - это безопасность с точки зрения обнаружения живости. В настоящее время любой желающий может использовать только фотографию авторизованного пользователя. Эту проблему можно решить, установив глубинную камеру. Глубинная камера, такая как Kinect, способна воссоздавать объекты в 3D. реализуя некоторый анализ, система способна сделать вывод, находится ли объект человека впереди или нет. Из-за некоторого ограничения бюджета разработчик не включил это в окончательный проект.

Собрав все вместе, разработчик может сделать вывод, что система может быть размещена только в ограниченных обстоятельствах, а это означает, что существует ограниченное количество мест, где вы можете использовать систему в реальной среде. Например, он совершенно неприменим для защиты ваших личных данных, таких как данные кредитной карты или доступ к вашему собственному дому. Потому что эта штука требует высочайшего уровня защиты. Однако его целесообразно использовать в корпоративной среде, например в кондоминиуме, школе или компании, где допустимы некоторые сбои.

ЗАКЛЮЧЕНИЕ

Во время работы над этим проектом разработчик узнал много новых приемов, которые помогут в дальнейшем опыте работы. Для выполнения системных требований потребовалось несколько навыков, которые не преподавались в университетских классах: машинное обучение, разработка API и системное программирование.

Было много исследований, прежде чем разработчик начал строить систему. Необходимо было знать, какие алгоритмы, какие модули использовать для более эффективного достижения целей проекта. Еще до начала расследования разработчик планировал создать архитектуру, которая уже была предоставлена проектом `facenet open source`. После некоторых исследований разработчик признал, что этот способ вызовет некоторые проблемы при реализации и интеграции. Разработчик был исследован о разработке RESTfull с CRUD-моделью, которая облегчает обслуживание системы. Более того, это дает возможность взаимодействовать с любыми другими платформами.

Все требования и цели системы были успешно достигнуты. Хотя, есть некоторые дополнительные реализации, которые могут повысить безопасность и точность работы системы. Он реализует глубинную камеру для обнаружения граней живости и модели тарирования с большим количеством данных, чтобы получить более высокую точность. Также рассматривали некоторые исследовательские работы, которые помогли нам разработать систему. Последовательность наших действий следующая: 1) закодировать изображение с помощью алгоритма НОГ для создания упрощенного изображения, чтобы найти ту часть изображения, которая выглядит как сгенерированная Ног кодирующая картинка лица; 2) выяснить положение лица в кадре изображения, упомянув конкретные точки лица. Как только мы получим эти очки, нам нужно будет централизовать этот образ глазами и ртом; 3) генерировать это централизованное Изображение через нейронную сеть, которая может измерять символы лица и сохранит его вывод в виде 128-мерного массива; 4) сравнивать эти размеры с другими и находить наиболее похожие с помощью машинного обучения. Следует также отметить, что данная нейронная сеть справляется с задачей классификации текстов иногда лучше, чем все остальные рассмотренные методы

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Francis Galton, "Personal identification and description," In *Nature*, pp. 173-177, June 21, 1888.
2. W. Zaho, "Robust image-based 3D face recognition," Ph.D. Thesis, Maryland University, 1999.
3. R. Chellappa, C.L. Wilson and C. Sirohey, "Human and machine recognition of faces: A survey," *Proc. IEEE*, vol. 83, no. 5, pp. 705- 740, may 1995.
4. T. Fromherz, P. Stucki, M. Bichsel, "A survey of face recognition," MML Technical Report, No 97.01, Dept. of Computer Science, University of Zurich, Zurich, 1997.
5. T. Riklin-Raviv and A. Sashua, "The Quotient image: Class based recognition and synthesis under varying illumination conditions," In *CVPR*, P. II: pp. 566-571, 1999.
6. G.j. Edwards, T.f. Cootes and C.J. Taylor, "Face recognition using active appearance models," In *ECCV*, 1998
7. S.H. Lin, S.Y. Kung, and L.J. Lin, "Face recognition/detection by probabilistic decision-based neural network," *IEEE Trans. Neural Networks*, vol. 8, pp. 114-132, 1997.
8. Jigar M. Pandya, Devang Rathod, Jigna J. Jadav, "A Survey of Face Recognition approach", *International Journal of Engineering Research and Applications (IJERA)* ISSN: 2248-9622 www.ijera.com Vol. 3, Issue 1, January -February 2013, pp.632-635
9. Jyoti S. Bedre, Shubhangi Sapkal, "Comparative Study of Face Recognition Techniques: A Review", *Emerging Trends in Computer Science and Information Technology – 2012(ETCS)*
10. L. Sirovich and M. Kirby, "Low-Dimensional procedure for the characterisation of human faces," *J. Optical Soc. of Am.*, vol. 4, pp. 519-524, 1987.
11. M. Kirby and L. Sirovich, "Application of the Karhunen- Loève procedure for the characterisation of human faces," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, pp. 831-835, Dec. 1990.
12. M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neuroscience*, vol. 3, pp. 71-86, 1991.
13. F. Samaria and F. Fallside, "Face identification and feature extraction using hidden markov models," *Image Processing: Theory and Application*, G. Vernazza, ed., Elsevier, 1993.
14. F. Samaria and A.C. Harter, "Parameterisation of a stochastic model for human face identification," *Proc. Second IEEE Workshop Applications of Computer Vision*, 1994
15. A.J. Goldstein, L.D. Harmon, and A.B. Lesk, "Identification of human faces," *Proc. IEEE*, vol. 59, pp. 748, 1971

16. Gosling, James; Joy, Bill; Steele, Guy; Bracha, Gilad. "The Java Language Specification, 2nd Edition".
17. Supriya D. Kakade, A Review Paper on Face Recognition Techniques, Vol-02, Issue 02, MAY 2016.
18. Guttag, John V. (2016-08-12). Introduction to Computation and Programming Using Python: With Application to Understanding Data. MIT Press. ISBN 978-0-262-52962-4.
19. "What is PostgreSQL?". PostgreSQL 9.3.0 Documentation. PostgreSQL Global Development Group. Retrieved 2013-09-20.
20. PostgreSQL: History". PostgreSQL Global Development Group. Retrieved 27 August 2016.
21. "History of MySQL". MySQL 5.1 Reference Manual. MySQL AB. Retrieved 26 August 2011.
22. Gutiérrez, C. (2018). 5 Easy Steps to Implement Scrum in Ecommerce Development - MagmaLabs Technical Blog. [online] MagmaLabs Technical Blog. Available at: <http://blog.magmalabs.io/2016/06/30/5-easy-steps-to-implement-scrum-in-ecommerce-development.html?cv=1> [Accessed 5 Sep. 2018].
23. En.wikipedia.org. (2018). Waterfall model. [online] Available at: https://en.wikipedia.org/wiki/Waterfall_model [Accessed 5 Sep. 2018].
24. Dean, Jeff; Monga, Rajat; et al. (November 9, 2015). "TensorFlow: Large-scale machine learning on heterogeneous systems". TensorFlow.org. Google Research. Retrieved November 10, 2015.
25. Alzheimer-europe.org. (2018). Alzheimer Europe - Research - Understanding dementia research - Types of research - Research methods. [online] Available at: <https://www.alzheimer-europe.org/Research/Understanding-dementia-research/Types-of-research/Research-methods> [Accessed 5 Sep. 2018].