

ОТЧЕТ

о проектной работе

по теме: Распознавание объектов

по дисциплине: Проектный практикум 1

Команда: SAI

Екатеринбург

2022

СОДЕРЖАНИЕ

[Введение 3](#_heading=h.gjdgxs)

[Команда 5](#_heading=h.30j0zll)

[Целевая аудитория 6](#_heading=h.1fob9te)

[Календарный план проекта 8](#_heading=h.3dy6vkm)

[Определение проблемы 10](#_heading=h.17dp8vu)

[Анализ аналогов 11](#_heading=h.26in1rg)

[Требования к продукту](#_heading=h.lnxbz9) (MVP) [12](#_heading=h.lnxbz9)

[Стек для разработки 15](#_heading=h.35nkun2)

[Прототипирование 19](#_heading=h.1ksv4uv)

[Разработка системы 20](#_heading=h.44sinio)

[Заключение 21](#_heading=h.2jxsxqh)

[Приложение 23](#_heading=h.z337ya)

[Список литературы 24](#_heading=h.3j2qqm3)

ВВЕДЕНИЕ

Человечество, всегда, даже на самых ранних этапах своего развития, стремилось взять под контроль происходящие вокруг процессы. Для этого мы постоянно работаем над вопросом получения как можно большего количества информации об окружающем нас мире. Одна из сторон этой потребности заключается в обеспечении безопасности своей жизни, жизни родных и близких людей, а также сохранности материальных ценностей и личного имущества. В наше время информационные источники постоянно совершенствуются. Наконец, настало время, когда у человека появилась-таки возможность увидеть вещи, которые могут быть скрыты от его взгляда. Реализовать идею «всевидящего ока» помогают системы видеонаблюдения.

Видеонаблюдение решает вопрос о безопасности контролируемого объекта на самом высоком уровне. Видеонаблюдение дома позволяет контролировать ситуацию в жилище человека круглосуточно. Видеонаблюдение в офисе позволит руководителю отслеживать действия своих сотрудников на протяжении рабочего дня. Современные системы видеонаблюдения и видеозаписи позволят не только увидеть подконтрольный объект в реальном времени, они предоставляют возможность просмотреть события, попавшие в поле зрения видеокамеры в любой момент прошедшего времени. Данные системы пользуются огромной популярностью благодаря стремительному росту и совершенствованию технологий видеонаблюдения.

Актуальность данной работы заключается в том, чтобы дать любому человеку возможность получить в своё распоряжение систему видеонаблюдения. Уникальностью этой системы будет её реализация в виде приложения для смартфона и наличие функции распознавания лиц с высокой точность.

Целью данной работы является разработка мобильного приложения,

позволяющего с помощью встроенной камеры смартфона или при подключении к внешней камере получить полноценную систему видеонаблюдения с функцией распознавания лиц.

Исходя из поставленной цели, в работе определены следующие задачи:

* Изучить возможные варианты применения приложения
* Проанализировать существующие аналоги
* Разработать приложение с функцией распознавания лиц

КОМАНДА

* Мамаев Алексей Игоревич РИ-110945 – Тимлид
* Плешивцев Денис Владимирович РИ-110948 – Дизайнер
* Балаба Софья Николаевна РИ-110940 – Программист

ЦЕЛЕВАЯ АУДИТОРИЯ

Для определения целевой аудитории мы использовали методику [5W](https://www.insales.ru/blogs/university/metodika-5w-marka-sherringtona?) Марка Шеррингтона. Это наиболее распространенный способ определения целевой аудитории и психологических характеристик, которыми обладают потенциальные потребители. Сегментация рынка проводится по 5 вопросам:

## Что?

Это сегментация по типу товара: что мы предлагаем потребительской группе? Какие товары/услуги?

Мы предлагаем приложение, позволяющее получить любому человеку в своё распоряжение систему видеонаблюдения с функцией распознавания лиц. Приложение предоставляет возможность использования встроенной камеры телефона или подключения к внешним камерам.

## Кто?

Это сегментация по типу потребителя: кто приобретает товар/услугу? Какой пол, возраст?

Процентное распределение ЦА:

* 30% – частные лица и предприниматели малого бизнеса. Сюда входят все клиенты, которые планируют оборудовать видеокамерами участок, дачу, частный дом, магазин и иные объекты
* 30% – муниципальные учреждения. Сюда относятся все государственные объекты, начиная от поликлиник и школ и заканчивая надзорными органами (налоговая инспекция, суды, отделы МВД и т. д.).
* 40% – коммерческие структуры. Это строительные компании, средние и крупные предприниматели и иные представители бизнеса, нуждающиеся в обновлении оборудования или монтаже видеокамер в новом помещении.

Информация взята с [Выгодный-вклад.рф](https://xn----8sbebdgd0blkrk1oe.xn--p1ai/biznes-plan/yslygi/biznes-plan-po-ustanovke-sistem-videonablyudeniya.html)

Почему?  
Это сегментация по типу мотивации к совершению покупки и потребления: какова потребность или мотивация клиента? какую проблему решает товар/услуга?

Видеонаблюдение – неотъемлемая часть современных систем безопасности. С развитием нашего общества развивается и преступность. Люди все больше и больше хотят себя обезопасить от нежелательных вторжений и посягательств на их жизнь или имущество. Без камер слежения трудно себе представить надежную систему безопасности.

Наше приложение вместе с системой видеонаблюдения предлагает функцию распознавания лиц. Благодаря этому открывается больше различных сценариев использования данной системы. (Например: Использование функции распознавания лиц на пропускном пункте)

## Когда?

Это сегментация по ситуации, в которой приобретается продукт: когда потребители хотят приобрести товар/услугу?

Нашим приложением потребитель захочет воспользоваться при открытии малого бизнеса и в любое время при желании взять свою территорию под более надёжный контроль.

Где?  
Это сегментация по месту покупок: в каком месте происходит принятие решения о покупке и сама покупка? — имеются в виду точки контакта с клиентом, где можно повлиять на решение.

Потребитель сможет ознакомиться со всей необходимой информацией непосредственно перед покупкой нашего приложения.

Точка контакта с клиентом — магазин приложений.

КАЛЕНДАРНЫЙ ПЛАН ПРОЕКТА

Название проекта: Приложение по распознаванию лиц MasterEye

Руководитель проекта: Астафьева Анна Викторовна

Таблица 1 – Календарный план проекта

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | Задача | Ответственный | Дедлайн | Неделя | | | | | | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Идея проекта | | | | | | | | | | | | | | | | | | |
| 1.1 | Выбор идеи проекта | Вся команда | 20.02.2022 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1.2 | Формирование команды | Вся команда | 27.02.2022 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1.3 | Регистрация команды | Тимлид | 02.03.2022 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Заполнение карточки проекта | | | | | | | | | | | | | | | | | | |
| 2.1 | Создание логотипа | Дизайнер | 13.03.2022 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.2 | Описание технического стека | Программист | 20.03.2022 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.3 | Заполнение карточки проекта | Тимлид | 23.03.2022 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.4 | Создание календарного плана | Тимлид | 08.04.2022 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Начало работы | | | | | | | | | | | | | | | | | | |
| 3.1 | Написать и обучить нейросеть или использовать обученную | Программист | 10.04.2022 |  |  |  |  |  |  | | |  |  |  |  |  |  |  |
| 3.2 | Создание иконки и макета приложения | Дизайнер | 17.04.2022 |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |
| 3.3 | Создать "пустое" приложение | Дизайнер | 20.04.2022 |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |
| 3.4 | MVP V1 | Программист | 17.04.2022 |  |  |  |  |  |  |  | | |  |  |  |  |  |  |
| 3.5 | Тестирование | Программист | 24.04.2022 |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |
| 3.6 | Проверка и представление проделанной работы | Тимлид | 24.04.2022 |  |  |  |  |  |  |  | | | |  |  |  |  |  |
| Доработка приложения | | | | | | | | | | | | | | | | | | |
| 4.1 | Соединить нейросеть с приложением | Дизайнер | 08.05.2022 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4.2 | Рефакторинг кода | Программист | 08.05.2022 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4.3 | Правки в макете приложения | Дизайнер | 01.05.2022 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4.4 | Тестирование | Программист | 08.05.2022 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4.5 | Устранение возможных проблем | Программист | 15.05.2022 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Защита проекта | | | | | | | | | | | | | | | | | | |
| 5.1 | Написание сценария защиты | Тимлид | 16.05.2022 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5.2 | Составление презентации | Тимлид | 20.05.2022 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5.3 | Дизайн презентации | Дизайнер | 20.05.2022 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5.4 | Пару слов о дизайне | Дизайнер | 20.05.2022 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5.5 | Информация о тех. Стеке и процессе разработки | Программист | 20.05.2022 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5.6 | Защита | Вся команда | 29.05.2022 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | | | | | | | | | | | |

# 

# 

# ОПРЕДЕЛЕНИЕ ПРОБЛЕМЫ

Мы воспользовались рядом эффективных способов выявления основных проблем покупателя:

1. Проанализировали конкурентов;
2. Изучили информацию с различных платформ
3. Опросили потенциальных клиентов

Рассмотрев и обобщив данную информацию, мы выявили что:

* Малому бизнесу, не имеющему биометрию для контроля доступа, приходится задействовать для этих целей персонал
* Частным лицам хотелось бы максимально обезопасить своё имущество
* Установка системы видеонаблюдения с функцией распознавания лиц может потребовать больших затрат и поэтому не все владельцы малого бизнеса (или частные лица) могут позволить себе это.

# АНАЛИЗ АНАЛОГОВ

Мы обнаружили два прямых конкурента, у которых аналогичное программное обеспечение на аналогичном рынке и работающее с целевой аудиторией, что и наш разрабатываемый продукт.

Аналоги:

1. Распознавание лиц Light

Применяется для автоматизации и контроля доступа людей на закрытую / охраняемую территорию

1. Терминал с распознаванием лица RADAR Plus КТ07

Бесконтактное автоматическое определение температуры тела, распознавание лица и одновременное получение высокоточного инфракрасного измерения температуры человека, быстро и точно

Отличия:

1. В случае с RADAR Plus КТ07 требуется терминал весом 1.9 кг + RADAR Plus КТ07 по большей части предназначен для измерения температуры человека
2. Не требует большого количества оборудования (для минимальной работоспособности нужен только телефон)
3. Распознавание лиц Light является продуктом больших объектов (банки, казино, гостиницы, рестораны и т. д.)
4. Наше приложение не требует больших затрат

ТРЕБОВАНИЕ К ПРОДУКТУ (MVP)

Таблица 2 – Требования к продукту (MVP)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Код** | **Требование** | **Источник** | **Приоритет** | **Примечание**  **Взаимосвязи с другими требованиями** |
| Устройство сайта | | | | |
| Т1\_1 | Название приложения «MasterEye» | Команда проекта | Высокий |  |
| Т1\_2 | Дизайн приложения должен быть в темных тонах | Команда проекта | Средний |  |
| Т1\_3 | Текст в приложении должен легко читаться | Команда проекта | Высокий | Связь с T\_2  Цвет и шрифт текста подбирается под цветовое решение приложения |
| Т1\_4 | Открытии приложения на экране загрузки должен быть логотип приложения и название команды разработчиков | Команда проекта | Высокий |  |
| Т1\_5 | Приложение должно быть просто в использовании | Пользователь | Высокий |  |
| Т1\_6 | До использования встроенной камеры смартфона пользователь добирается не более, чем за 2 клика | Пользователь | Высокий | Связь с Т\_2 |
| Т1\_7 | Подключение к внешней камере занимает не более 4 кликов | Пользователь | Высокий | Связь с Т\_2 |
| Структура приложения | | | | |
| Т2\_1 | Возможность вернуться в меню в один клик | Заказчик | Высокий |  |
| Т2\_1 | Возможность написать команде разработчиков | Команда проекта | Высокий |  |
| Т2\_3 | Возможность добавления новых и удаления уже не нужных фото людей | Заказчик | Средний |  |
| Т2\_4 | Высокая точность определения личности человека | Заказчик | Высокий |  |
| Технические требования | | | | |
| Т4\_1 | Приложение должно быть написано на языке программирования Dart с использованием фреймворка Flutter | Команда проекта | Высокий |  |
|  |  |  |  |  |

# СТЕК ДЛЯ РАЗРАБОТКИ

**Back-end**

[Face recognition](https://docs.google.com/document/d/1WslkpmrRsJrJYJJg84KfWaIdJCxyRs8u/edit?usp=sharing&ouid=110900402661578047925&rtpof=true&sd=true) – обёртка над библиотекой распознавания лиц [dlib](http://dlib.net/) построена на основе глубокого обучения. Модель имеет точность 99.38% на [Labeled Faces in the Wild](http://vis-www.cs.umass.edu/lfw/) benchmark. Библиотека также предоставляет простое средство командной строки, которое позволяет face recognition делать распознавание лиц из папки с изображениями из командной строки. dlib предоставляет обученные модели распознавания черт лица и кодирования лиц, используемые в face recognition.

Dlib - библиотека с [открытым исходным кодом](http://dlib.net/license.html), инструментарий C++, содержащий алгоритмы машинного обучения и инструменты для создания сложного программного обеспечения на C++ для решения реальных задач. Она используется как в промышленности, так и в научных кругах в широком спектре областей, включая робототехнику, встроенные устройства, мобильные телефоны и большие высокопроизводительные вычислительные среды.

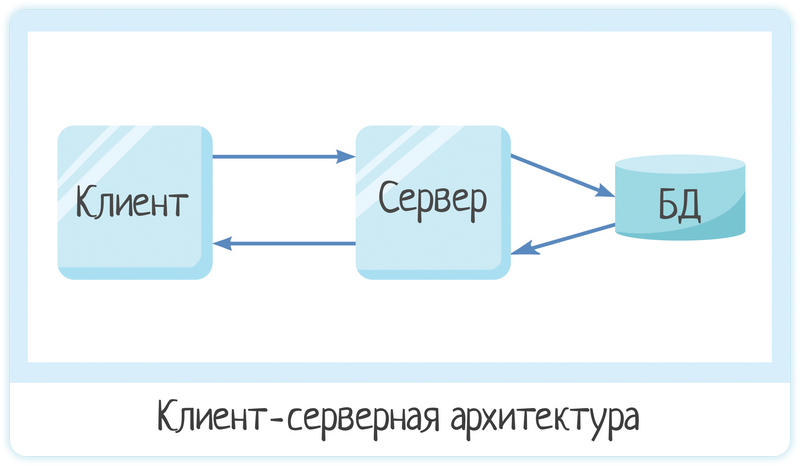
[Модель: lib\_face\_recognition\_resnet\_model\_v1.dat.bz2](https://githubhelp.com/davisking/dlib-models?ysclid=l4i6o8ay78199696722#:~:text=are%20summarized%20below.-,dlib_face_recognition_resnet_model_v1.dat.bz2,-This%20model%20is)

[Набор данных VGG Face](https://translated.turbopages.org/proxy_u/en-ru.ru.864191ed-62ac50d6-e1744660-74722d776562/https/www.robots.ox.ac.uk/~vgg/data/vgg_face/)

[Набор данных скраба лица](https://stefan.winklerbros.net/Publications/icip2014a.pdf)

Также в основе всей обработки и анализа данных используются библиотеки Python, такие как numpy, scipy, scikit-image, pillow.

Необходимостью стало создание сервера для соединения нейронной сети и клиента (мобильное приложение). По стандарту для будущей реализации выбрана привычная клиент-серверная архитектура. Для написания сервера лучшим оказался фреймворк [Flask](https://flask.palletsprojects.com/en/2.1.x/), который значительно проще множества web-фреймворков. Flask - микрофреймворк для Python, основанный на [Werkzeug,](https://werkzeug.palletsprojects.com/en/2.1.x/) [Jinja 2](https://jinja.octoprint.org/) и отличных решениях



**Front-end**

Для разработки мобильного приложения был использован фреймворк [Flutter](https://flutter.dev/), который поддерживается компанией Google и позволяет создавать приложения на Android и IOS с использованием языка программирования [Dart](https://dart.dev/).

Преимущества:

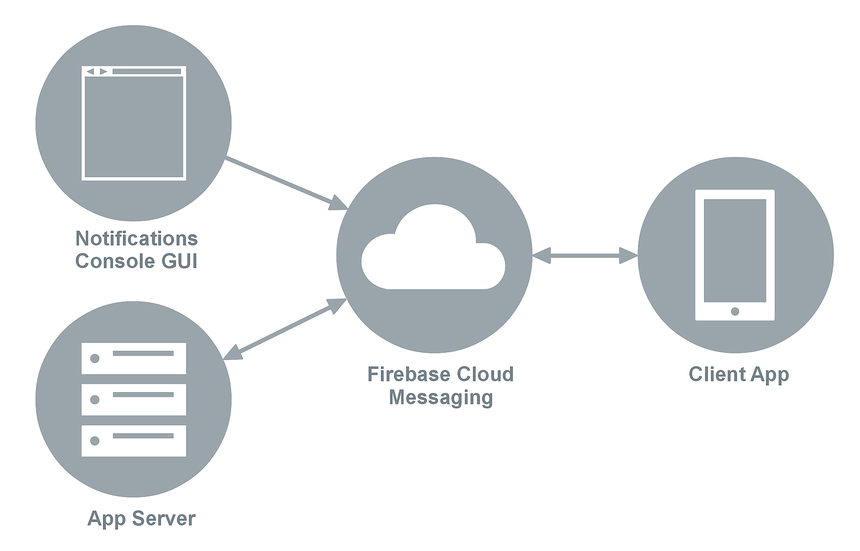
* Кроссплатформенность
* Поддержка Google. Поддержка минимум на 5 лет обеспечена.
* высокая графическая производительность (возможность отображения 120 кадров в секунду) — основная заявленная особенность
* наборы виджетов и средства разработки (Flutter DevTools).
* Простота и выразительность вёрстки

**Обработка видеокадров в режиме реального времени** **реализуется при помощи плагина** [**ML kit**](https://developers.google.com/ml-kit/guides)**.**

ML Kit предоставляет опыт Google в области машинного обучения разработчикам мобильных приложений в эффективном и простом в использовании пакете. Сделайте свои приложения для iOS и Android более привлекательными, персонализированными и полезными с помощью решений, оптимизированных для работы на устройстве.Распознавание лиц выполняется на устройстве и достаточно быстро, чтобы его можно было использовать в приложениях реального времениМы использовали инструменты поддерживаемые компанией Google, которая постоянно совершенствует свои модели машинного обучения и не ещё долгое время не прекратит срок поддержки.

[**Firebase Cloud Messaging**](https://firebase.google.com/docs/cloud-messaging?hl=en) **(FCM)** — это кроссплатформенная служба, которая обрабатывает отправку, маршрутизацию и очередь сообщений между серверными приложениями и мобильными клиентскими приложениями. FCM является преемником Google Cloud Messaging (GCM), и он основан на службах Google Play.

Как показано на следующей схеме, FCM выступает в качестве посредника между отправителями сообщений и клиентами. Клиентское приложение — это приложение с поддержкой FCM, которое выполняется на устройстве. Сервер приложений (предоставленный вами или вашей компанией) — это сервер с поддержкой FCM, с которым ваше клиентское приложение взаимодействует через FCM. В отличие от GCM, FCM позволяет отправлять сообщения клиентским приложениям напрямую с помощью графического пользовательского интерфейса уведомлений консоли Firebase:



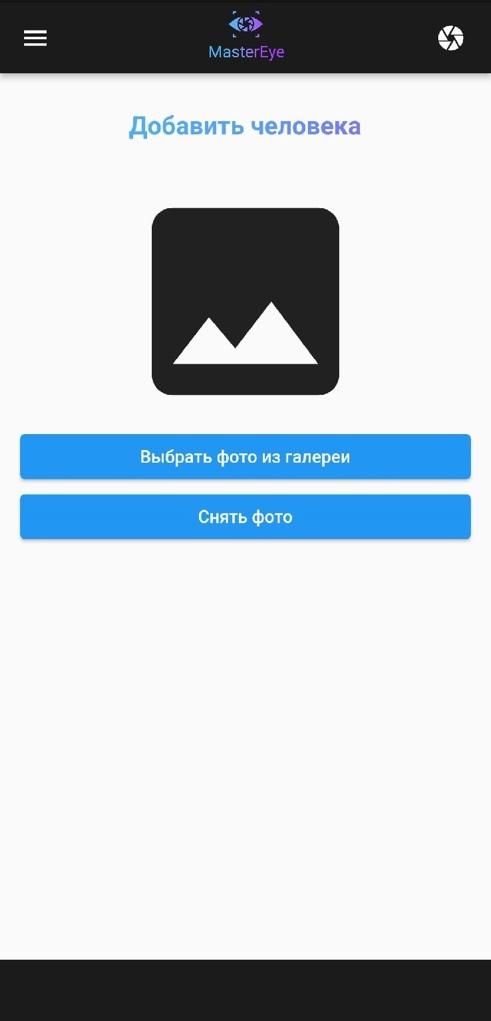
ПРОТОТИПИРОВАНИЕ

  Главней экран Определение лица

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как человек, внутренний, мужчина, очки

Автоматически созданное описание

Меню Добавление фотоИзображение выглядит как текст

Автоматически созданное описание

РАЗРАБОТКА СИСТЕМЫ

Алгоритм работы приложения:

1. В меню приложения пользователь выбирает между использованием камеры телефона и подключением к внешней камере
2. Выбор доступной камеры
3. Подключение к камере
4. Добавление фото людей
5. Удаление фото (при необходимости)
6. Приложение в режиме реального времени передает видеопоток на сервер
7. На сервере видео обрабатывается нейросетью
8. Видео с распознанными и отмеченными лицами людей отправляется обратно в приложение
9. Полученное видео выводится пользователю
10. При обнаружении человека приложение записывает его личность и время обнаружения

ЗАКЛЮЧЕНИЕ

Рассмотренные вопросы в рамках данной проектной работы, являются основными при использовании нейронных сетей в системах видеонаблюдения. Приведены определенные правила и принципы, которые позволят в дальнейшем правильно сопровождать, поддерживать, и грамотно разрабатывать новый функционал для данной разработки.

Основные технические решения, которые были предложены в данной проектной работе, включают в себя следующие аспекты:

* Платформа создания макета приложения – Figma - графический онлайн-редактор для совместной работы. Интенсивно применяется для создания прототипа интерфейса приложения. Позволяет обсудить правки с коллегами в реальном времени.
* Платформа для выгрузки и хранения кода – GitHub - своеобразная социальная сеть для разработчиков, позволяющая им просматривать код друг друга, оставлять комментарии, а также помогать в разработке.
* Платформа для создания приложений под мобильные устройства – Android Studio – интегральная среда разработки (IDE) для работы с платформой Android.
* Фреймворк Flutter – Комплект средств разработки и фреймворк с открытым исходным кодом для создания мобильных приложений под Android и iOS, веб-приложений, а также настольных приложений под Windows, macOS и Linux с использованием языка программирования Dart, разработанный и развиваемый корпорацией Google.

Кроме вышеупомянутых технических решений, позволяющих разработать и доработать функционал приложения, в проектной работе затрагиваются обязательные вопросы эффективной работоспособности новых компонентов. А именно вопросы качественных показателей и масштабируемости решений, вопросы тестирования.

В итоге представлен эффективный продукт, позволяющий потребителю:

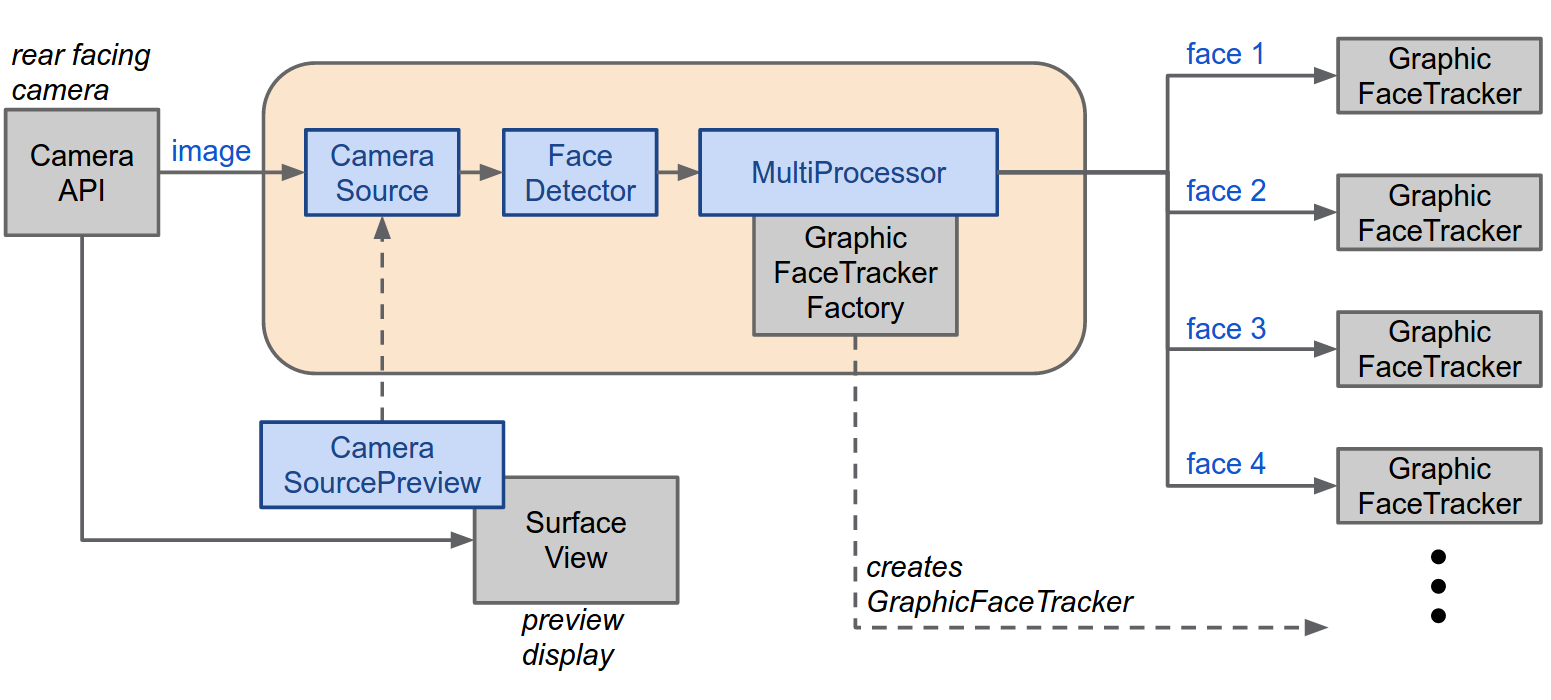
* Подключить телефон к внешней камере
* Распознавать лица с помощью камер в реальном времени
* Использовать встроенную камеру телефона

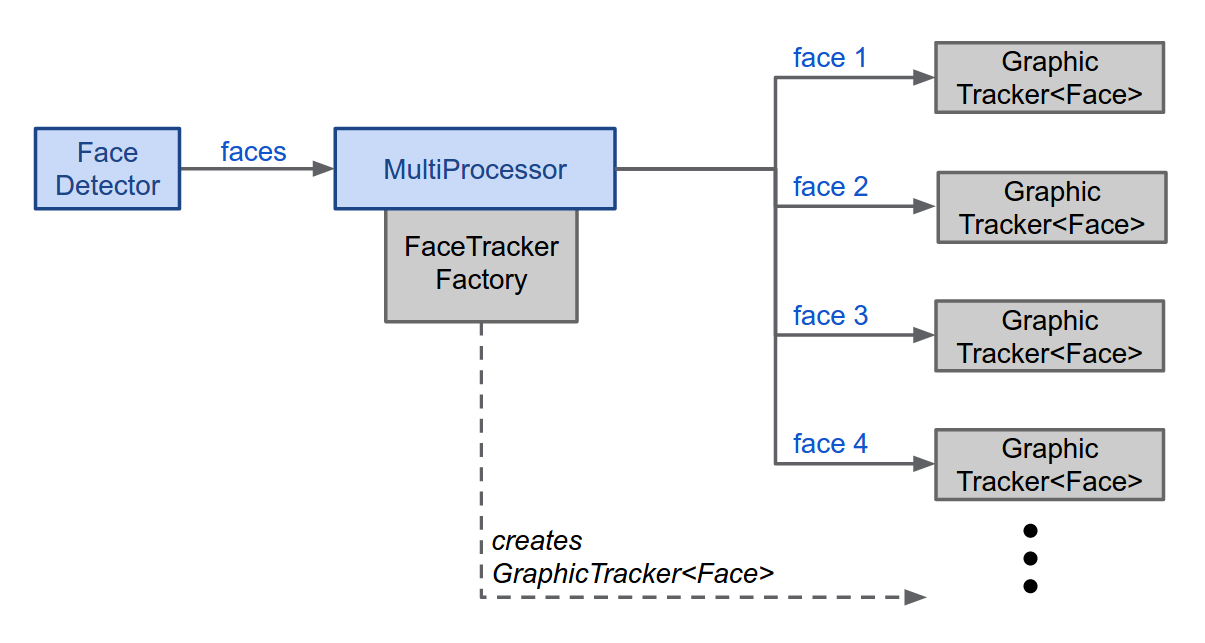
Целью данной работы являлась разработка приложения, позволяющего любому человеку получить в своё распоряжение систему видеонаблюдения с функцией распознавания лиц и возможностью подключения к внешним камерам.

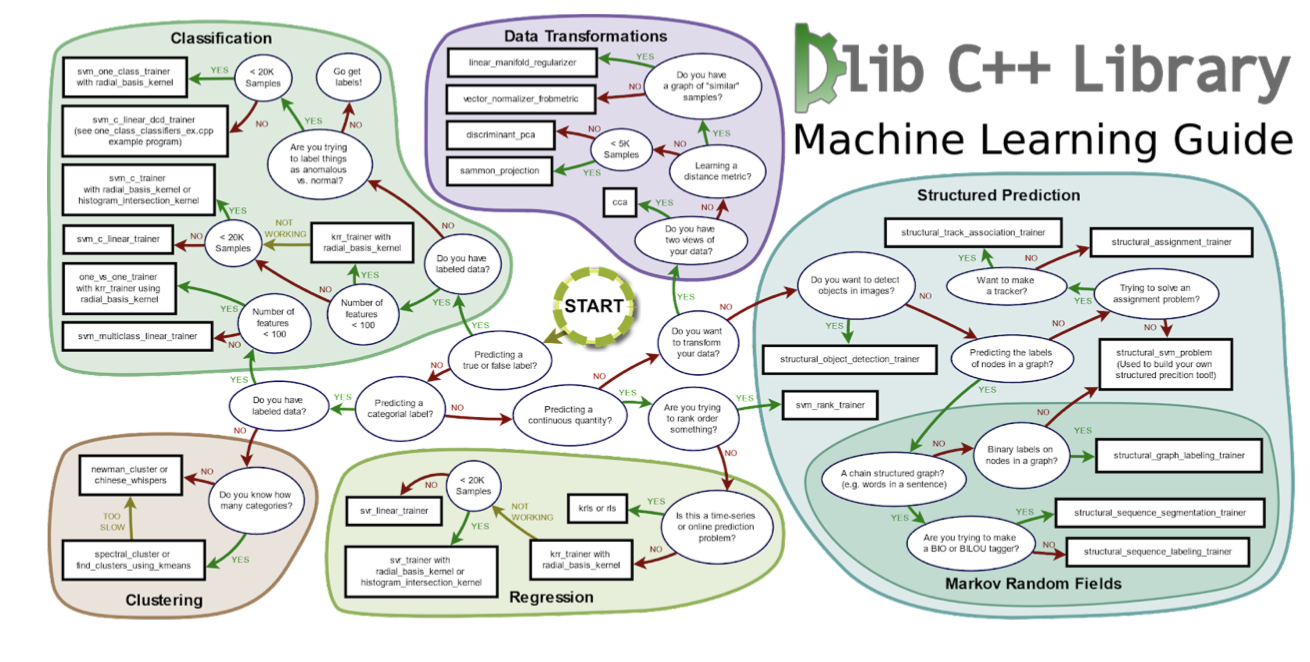
Исходя из поставленной цели, в работе были определены следующие задачи:

* Провести исследование совокупности методов машинного обучения и математических моделей
* Проанализировать существующие решения поставленной задачи
* Разработать приложение с функцией распознавания лиц

Таким образом, все цели и задачи, поставленные вначале написания данной проектной работы, были полностью выполнены и освещены.

ПРИЛОЖЕНИЕ





СПИСОК ЛИТЕРАТУРЫ

1. Dlib: [Подробнее о распознавании лиц в dlib](http://blog.dlib.net/2017/02/high-quality-face-recognition-with-deep.html)
2. Исследование на модель нейронной сети: [ResNet: остаточная CNN для классификации изображений (neurohive.io)](https://arxiv.org/pdf/1512.03385.pdf)
3. Бэнчмарк: [A\_Set\_of\_Neural\_Network\_Benchmark\_Problems\_and\_Benchmarking\_Rules](https://www.researchgate.net/publication/2738156_PROBEN1_-_A_Set_of_Neural_Network_Benchmark_Problems_and_Benchmarking_Rules)
4. Deploy(развертывание) нейронной сети на сервере:
5. [Проектирование RESTful API с помощью Python и Flask / Хабр (habr.com)](https://habr.com/ru/post/246699/?)
6. [How to integrate Flutter app with Python code - Stack Overflow](https://stackoverflow.com/questions/64853113/how-to-integrate-flutter-app-with-python-code)
7. [Build a REST API with Python](https://towardsdatascience.com/the-right-way-to-build-an-api-with-python-cd08ab285f8f)
8. [HTTP Requests](https://www.tutorialspoint.com/http/http_requests.htm)
9. [Front-End Back-End : Introduction](https://www.clariontech.com/blog/cloud-computing-architecture-what-is-front-end-and-back-end#:%7E:text=The%20cloud%20computing%20architecture%20comprises,visible%20to%20the%20end%2Duser.)
10. Flask: [Flask против Django (2019) Какой web-framework выбрать? (python-scripts.com)](https://python-scripts.com/flask-or-django?)
11. Макет: [Prototype in Figma](https://www.figma.com/file/1QMHxGpNLAF8CuYpaC1SKK/MasterEye?node-id=0%3A1)