



Информатика и
вычислительная техника

Московский институт
электроники и математики

Москва
2022

Разработка веб-приложения CorrectPlan с использованием технологии искусственного интеллекта

Работу выполнила:
Карманова Кристина, БИВ182

Преподаватель курса "Разработка веб-приложений":
Профессор Трубочкина Надежда Константиновна

Постановка задачи

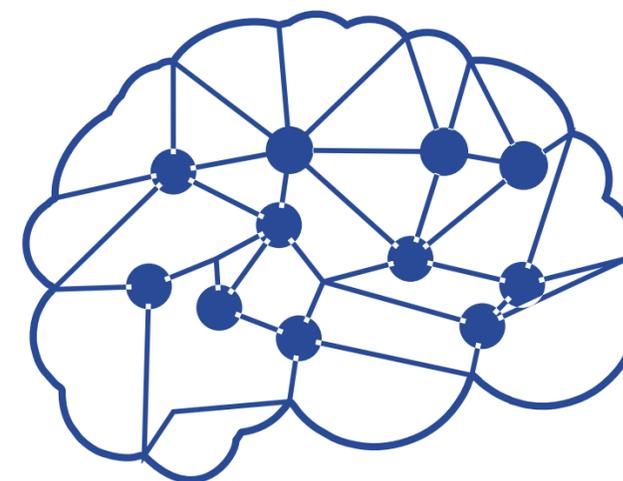
Упростить процессы подготовки и проверки фото и видео материалов на **корректность плана** посредством создания уникального **веб-приложения** с применением **искусственного интеллекта**

Цель

Разработать **приложение** с применением **функций искусственного интеллекта**, позволяющее **узнавать крупность плана** на фото и видео

Область применения

Технология находит широкое применение в **сфере создания подготовки фото и видео материалов**. Оно может быть использовано как профессиональными операторами и монтажёрами, так и обычными пользователями

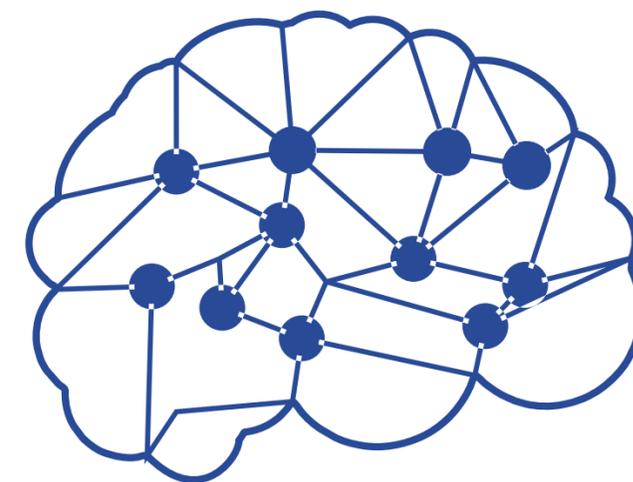


Актуальность

С активным развитием технологий в наш мир пришли современные телефоны, фотоаппараты и камеры, позволяющие запечатлеть практически каждый миг жизни человека. Многие люди стали постоянно снимать фото и видео важных и дорогих сердцу событий, однако не у всех получается сделать "красивый" кадр. Разрабатываемый сервис подскажет человеку, где он использовал неверный план, что поможет быстрее научиться основам фотографии

Новизна

Разрабатываемое приложение является комплексным решением, которое позволит человеку не только узнать, почему важно следовать правилам подбора плана, но и загрузить личное фото и видео для проверки корректности плана и наглядного выделения тела и лица человека

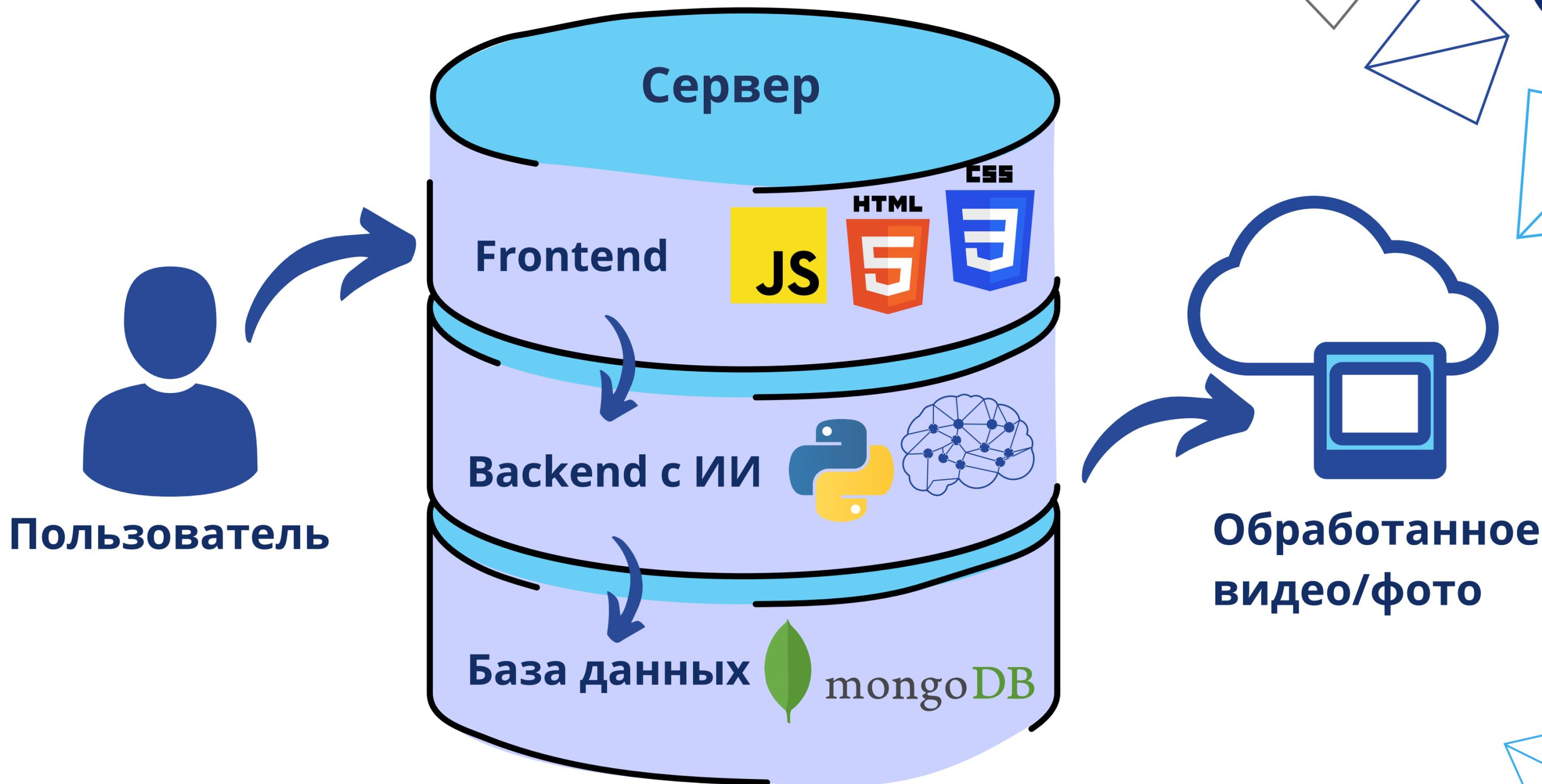


Обзор аналогов

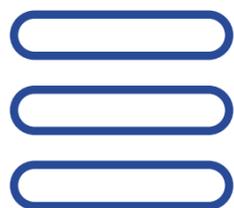
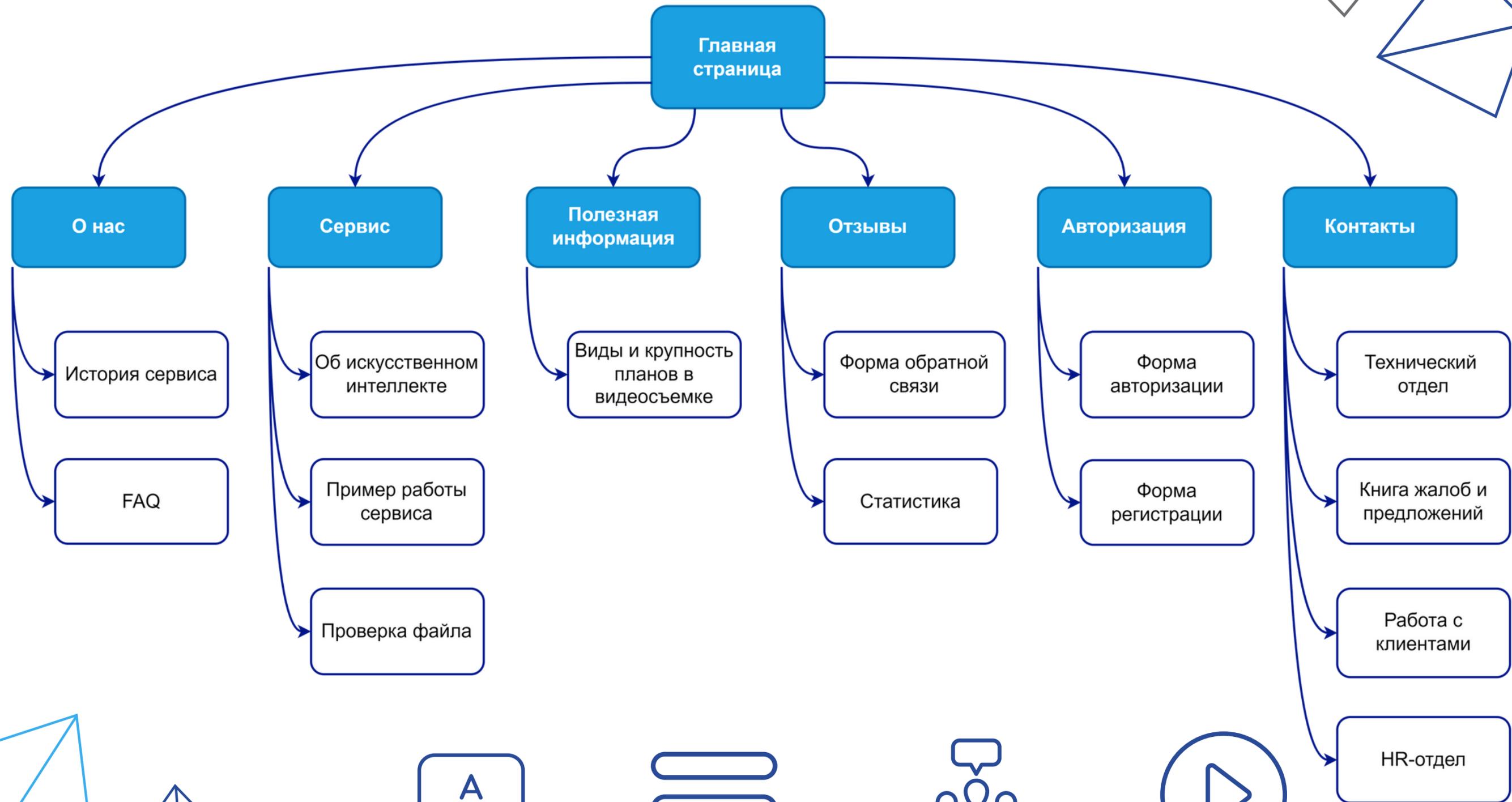


Название аналога	Достоинства	Недостатки
Fantastic Factory	<ul style="list-style-type: none">• Множество обучающих статей об основах фотографии	<ul style="list-style-type: none">• Нет возможности проверить полученные навыки• Нет возможности загрузить свои фото и видео материалы
CRAFTKINO	<ul style="list-style-type: none">• Множество справочной информации о том, как снять короткометражный фильм, и разбор основ кинематографа	<ul style="list-style-type: none">• Нет возможности проверить полученные навыки• Нет возможности загрузить свои фото и видео материалы
Science Hunter	<ul style="list-style-type: none">• Программа позволяет распознать лица по доступным картинкам из загруженной базы	<ul style="list-style-type: none">• Нет справочной информации о фото и видео съемке• Нет возможности загрузить видео• Программа не распознает тела• Программа не определяет крупность плана
NTECH LAB	<ul style="list-style-type: none">• Программа распознает лица и силуэты людей	<ul style="list-style-type: none">• Нет справочной информации о фото и видео съемке• Нет возможности определить крупность плана

Структура веб-приложения



Структура приложения



Макет страницы

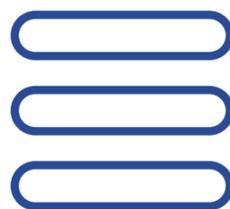


Логотип и
название

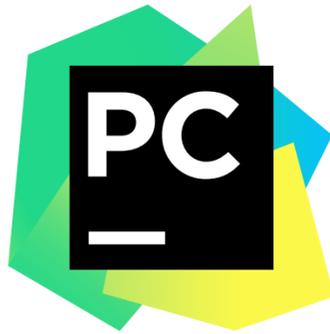
Меню

Контентная часть

Footer с копирайтом



Выбор средств разработки



PyCharm - интегрированная среда разработки для языка программирования Python. Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживает веб-разработку на Django. PyCharm разработана компанией JetBrains на основе IntelliJ IDEA.

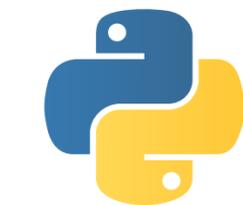
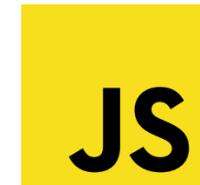


Caffe - среда для глубинного обучения. Caffe является открытым программным обеспечением, распространяемым под лицензией BSD license. Написано на языке C++, и поддерживает интерфейс на языке Python.

Название Caffe произошло от сокращения «Convolution Architecture For Feature Extraction» (Свёрточная архитектура для извлечения признаков)



MongoDB - документоориентированная система управления базами данных, не требующая описания схемы таблиц. Считается одним из классических примеров NoSQL-систем, использует JSON-подобные документы и схему базы данных.



Структура БД



```
LocalBase localhost:27017
db.getCollection('login')
```

- LocalBase (7)
 - System
 - config
 - recognition
 - Collections (3)
 - login
 - statistic
 - status

login 0.015 sec.

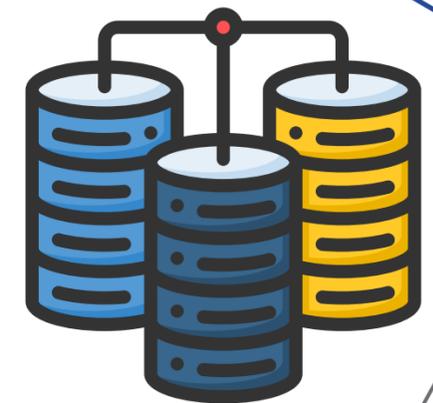
_id	login	password
1	admin	admin
2	Ivan	123456789
3	Kris	123
4	123	123

status 0.001 sec.

_id	file_name	status	progress
1	e4d8...	done	100
2	eda1...	fail	98
3	1193...	done	100
4	c35b...	done	100
5	f983...	done	100
6	04ca...	done	100
7	4ff8d...	done	100
8	2353...	done	100
9	5e64...	done	100
10	d457...	done	100

statistic 0.001 sec.

_id	AgainRadios_no	FriendRadios_no	HelpRadios_no	LikeRadios_yes	PayRadios_no	WorkRadios_yes	AgainRadios_yes	FriendRadios_yes	HelpRadios_yes	LikeRadios_no	PayRadios_yes	WorkRadios_no
1	3	5	3	5	5	3	4	2	4	4	2	4



О нас

О НАС

Здесь вы узнаете о том, как появилась наша компания и как она развивалась на протяжении своего существования.

ПРОБЛЕМА



Один из наших основателей имеет хобби - снимать своих друзей и родных на камеру.

Однако, однажды он столкнулся с проблемой, что для того, чтобы видеоматериал получался красивым и качественным, необходимо учитывать множество разных нюансов.

Он перечитал множество различной литературы и фото и видеосъемки и задается вопросом "А как же можно всё это упростить? Или, хотя бы, более точно понимать ошибки?"



Тогда у него возникла идея сделать такой сервис, который мог бы помочь хотя бы с незначительной частью обработки и проверки видеоматериала.

Он подумал, что можно сделать некую программу-помощник для определения крупности плана человека на фото видео.

Такая система позволила бы не только проверить видео, но и обучаться, поскольку программа может не только сказать, в чем ошибка, но и показать.

РЕАЛИЗАЦИЯ



Он собрал команду и создал программу, которая с помощью нейронной сети определяет крупность плана.

Более того, для повышения точности распознавания было решено использовать две нейросети - одну, которая находит тело человека и другую, которая находит его лицо. Далее алгоритм вычисляет отношение лица к телу, что позволяет определить крупность плана человека, ведь пропорции у всех людей практически одинаковы.

ОТВЕТЫ НА ЧАСТЫЕ ВОПРОСЫ

Здесь вы можете прочитать ответы на самые часто задаваемые вопросы

ДЛЯ КАКОГО СОЗДАН СЕРВИС?

Сервис нацелен, в основном, на видеоператоров и монтажеров. Но также им могут пользоваться и обычный люди.

ОН БЕСПЛАТНЫЙ?

Да, наш сервис абсолютно бесплатный.

КАК ИМ ПОЛЬЗОВАТЬСЯ?

Чтобы воспользоваться сервисом, вам необходимо загрузить видео во вкладке "Сервис" и дождаться завершения обработки.

КАК ОН РАБОТАЕТ?

Видео обрабатывается нейросетью, которая определяет крупность плана человека на видео.

В ЧЕМ ЕГО ПОЛЕЗНОСТЬ?

Сервис полезен тем людям, кому необходимо часто контролировать или снимать видео, чтобы определить, является ли композиция в их видео верной.

СКОЛЬКО РАЗ ЕГО МОЖНО ИСПОЛЬЗОВАТЬ?

Вы можете воспользоваться сервисом столько раз, сколько необходимо. Он абсолютно бесплатный.

МОИ ВИДЕО В БЕЗОПАСНОСТИ?

Да, они обрабатываются на сервере, после чего удаляются. Мы не храним ваши файлы.

КАК МОЖНО С ВАМИ СВЯЗАТЬСЯ?

Чтобы связаться с разработчиками сервиса, перейдите во вкладку "Контакты".

О НАС

Сервис для проверки крупности плана. Работаем с 2000 года.

МЕНЮ

О нас
Полезная информация
Отзывы
Сервис
Авторизация
Контакты

ПОДПИШИТЕСЬ НА НОВОСТНУЮ РАССЫЛКУ

Введите e-mail Подписаться

СОЦИАЛЬНЫЕ СЕТИ

f v o in

Мобильная версия



ПК

О НАС

Здесь вы узнаете о том, как появилась наша компания и как она развивалась на протяжении своего существования.

ПРОБЛЕМА



Один из наших основателей имеет хобби - снимать. Однако, однажды он столкнулся с проблемой, что для того, чтобы видеоматериал получался красивым и качественным, необходимо учитывать множество разных нюансов.

Он перечитал множество различной литературы и фото и видеосъемки и задается вопросом "А как же можно всё это упростить? Или, хотя бы, более точно понимать ошибки?"



Тогда у него возникла идея сделать такой сервис, который мог бы помочь хотя бы с незначительной частью обработки и проверки видеоматериала.

Он подумал, что можно сделать некую программу-помощник для определения крупности плана человека на фото видео.

Такая система позволила бы не только проверить видео, но и обучаться, поскольку программа может не только сказать, в чем ошибка, но и показать.

РЕАЛИЗАЦИЯ



Он собрал команду и создал программу, которая с помощью нейронной сети определяет крупность плана.

Более того, для повышения точности распознавания было решено использовать две нейросети - одну, которая находит тело человека и другую, которая находит его лицо. Далее алгоритм вычисляет отношение лица к телу, что позволяет определить крупность плана человека, ведь пропорции у всех людей практически одинаковы.

ОТВЕТЫ НА ЧАСТЫЕ ВОПРОСЫ

Здесь вы можете прочитать ответы на самые часто задаваемые вопросы

ДЛЯ КАКОГО СОЗДАН СЕРВИС?

Сервис нацелен, в основном, на видеоператоров и монтажеров. Но также им могут пользоваться и обычный люди.

ОН БЕСПЛАТНЫЙ?

Да, наш сервис абсолютно бесплатный.

КАК ИМ ПОЛЬЗОВАТЬСЯ?

Чтобы воспользоваться сервисом, вам необходимо загрузить видео во вкладку "Сервис" и дождаться завершения обработки.

КАК ОН РАБОТАЕТ?

Видео обрабатывается нейросетью, которая определяет крупность плана человека на видео.

В ЧЕМ ЕГО ПОЛЕЗНОСТЬ?

Сервис полезен тем людям, кому необходимо часто контролировать или снимать видео, чтобы определить, является ли композиция в их видео верной.

СКОЛЬКО РАЗ ЕГО МОЖНО ИСПОЛЬЗОВАТЬ?

Вы можете воспользоваться сервисом столько раз, сколько необходимо. Он абсолютно бесплатный.

МОИ ВИДЕО В БЕЗОПАСНОСТИ?

Да, они обрабатываются на сервере, после чего удаляются. Мы не храним ваши файлы.

КАК МОЖНО С ВАМИ СВЯЗАТЬСЯ?

Чтобы связаться с разработчиками сервиса, перейдите во вкладку "Контакты".

О НАС

Сервис для проверки крупности плана. Работаем с 2000 года.

МЕНЮ

О нас
Полезная информация
Отзывы
Сервис
Авторизация
Контакты

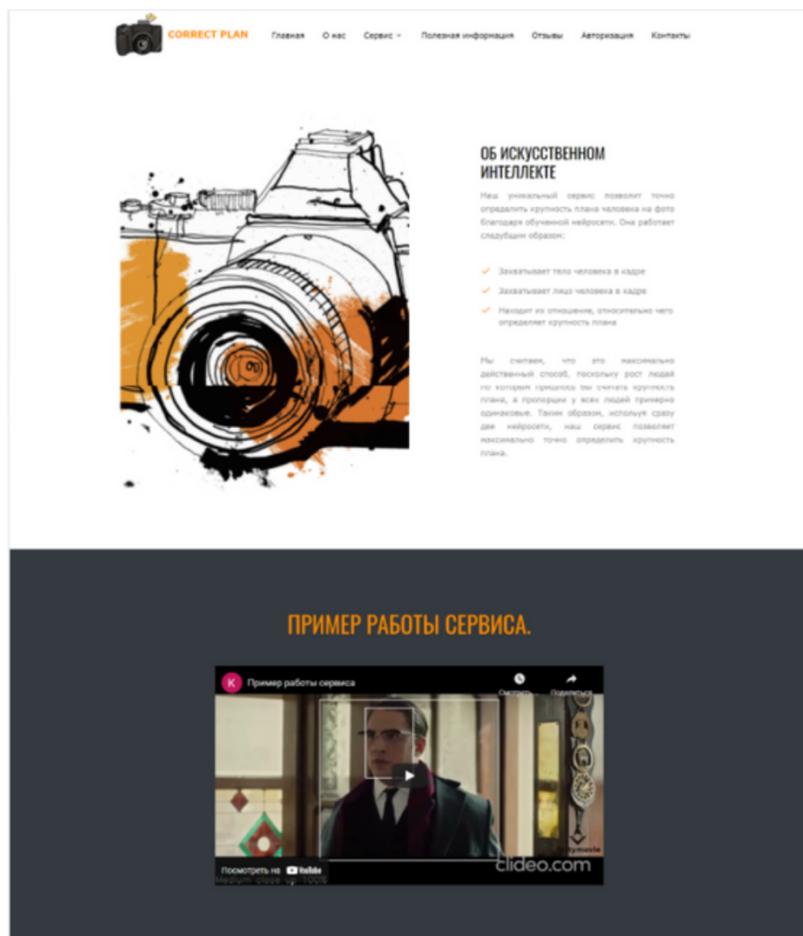
ПОДПИШИТЕСЬ НА НОВОСТНУЮ РАССЫЛКУ

Введите e-mail Подписаться

СОЦИАЛЬНЫЕ СЕТИ

f v o in





Мобильная версия

ПК



ОБ ИСКУССТВЕННОМ ИНТЕЛЛЕКТЕ

Наш уникальный сервис позволит точно определить крупность плана человека на фото благодаря обученной нейросети. Она работает следующим образом:

- ✓ Захватывает тело человека в кадре
- ✓ Захватывает лицо человека в кадре
- ✓ Находит их отношение, относительно чего определяет крупность плана

Мы считаем, что это максимально действенный способ, поскольку рост людей может отличаться, что влияет на поазатели, по которым пришлось бы считать крупность плана, а пропорции у всех людей примерно одинаковые. Таким образом, используя сразу две нейросети, наш сервис позволяет максимально точно определить крупность плана.

1		Extreme Close Up Деталь	ECU	Деталь
2		Big Close Up Предельно крупный	BCU	
3		Close Up Крупный	CU	Крупный
4		Medium Close Up Полукрупный	MCU	
5		Medium Shot Средний	MS	Средний
6		Medium Long Shot Полонный	MLS	2/3
7		Long Shot Полный	LS	Общий
8		Very Long Shot Общий	VLS	
9		Extreme Long Shot Дальний	XLS	Дальний

ПРОВЕРИТЬ ФАЙЛ
ИНСТРУКЦИЯ ПО РАБОТЕ С СЕРВИСОМ:

ПРОВЕРИТЬ ФАЙЛ
ИНСТРУКЦИЯ ПО РАБОТЕ С СЕРВИСОМ:

- ✓ Выберите файл с вашего устройства с помощью кнопки "Выберите файл"
- ✓ Загрузите выбранный для проверки файл с помощью кнопки "Загрузить", расположенной ниже
- ✓ Дождитесь окончания обработки и загрузки готового файла (шкала под кнопкой покажет вам текущий процент выполнения)
- ✓ После окончания обработки, вам станет доступна кнопка "Скачать", нажмите на неё
- ✓ При нажатии кнопки "Скачать", в новом окне откроется ссылка для скачивания вашего итогового файла после обработки

ВЫБЕРИТЕ И ЗАГРУЗИТЕ ФАЙЛ

Выберите файл | Файл не выбран



ПРОЦЕНТ ВЫПОЛНЕНИЯ ОБРАБОТКИ



О НАС

Сервис для проверки крупности плана. Работаем с 2000 года.

МЕНЮ

- О нас
- Полезная информация
- Отзывы
- Сервис
- Авторизация
- Контакты

ПОДПИСЫВАЙТЕСЬ НА НОВОСТНУЮ РАССЫЛКУ

Введите e-mail

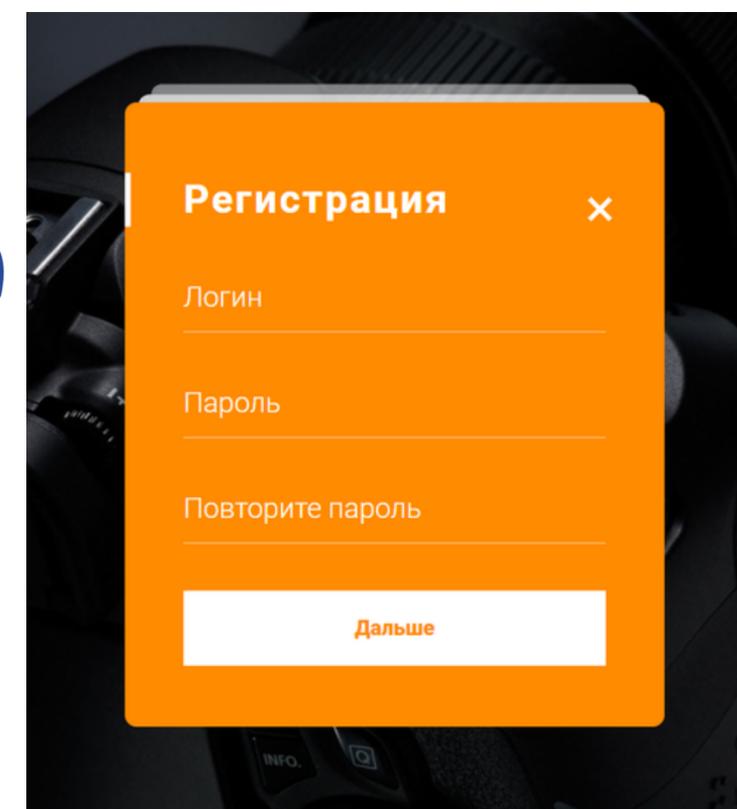
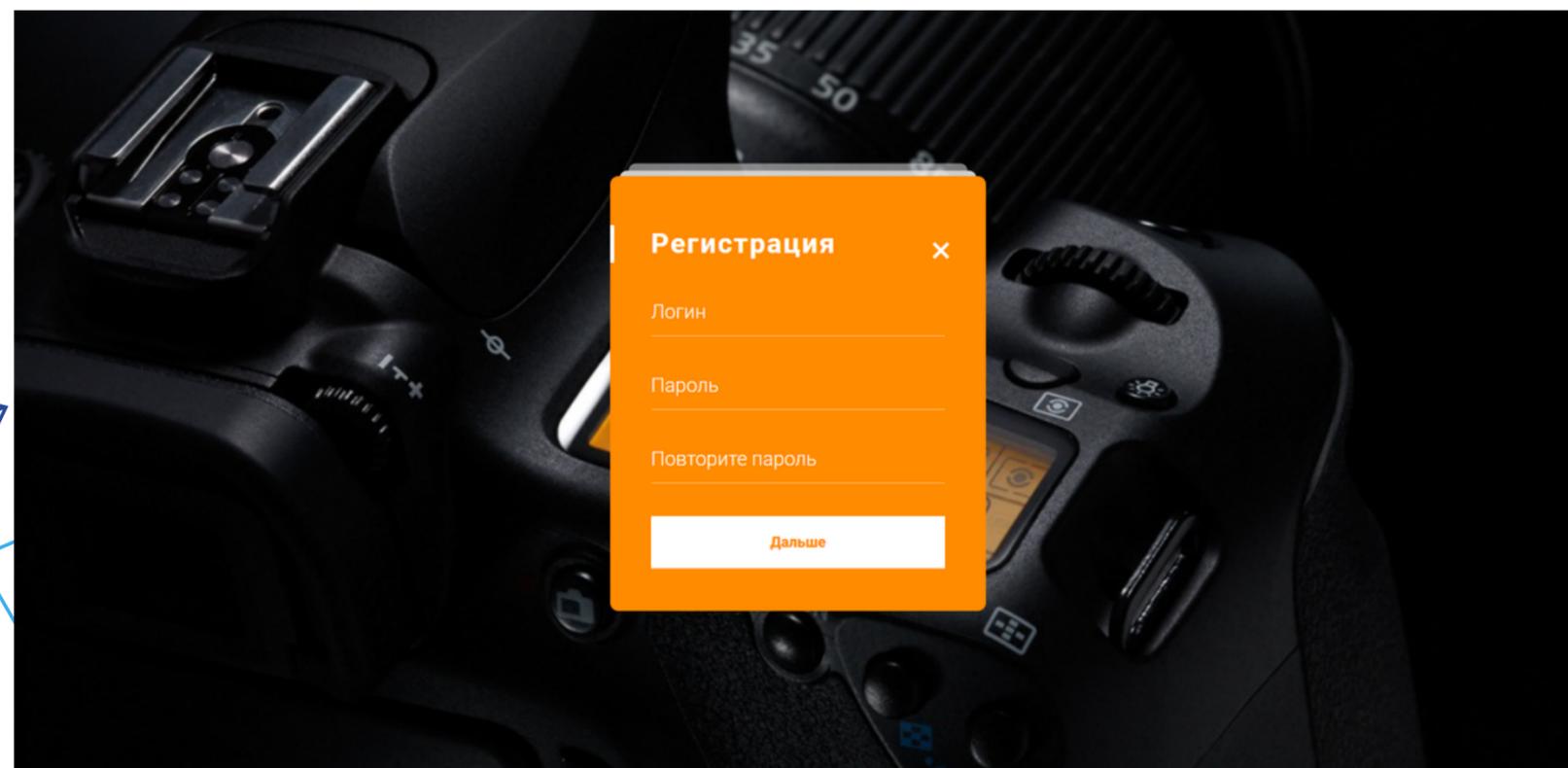
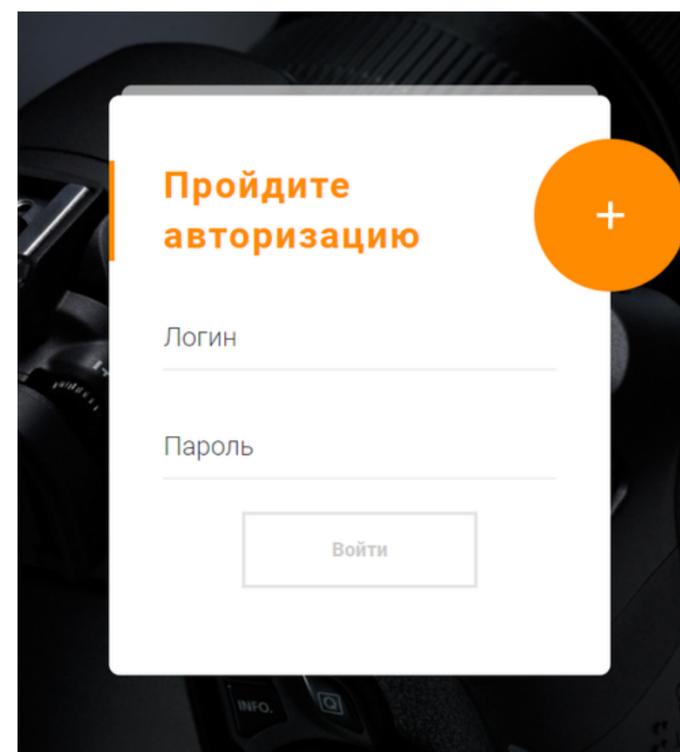
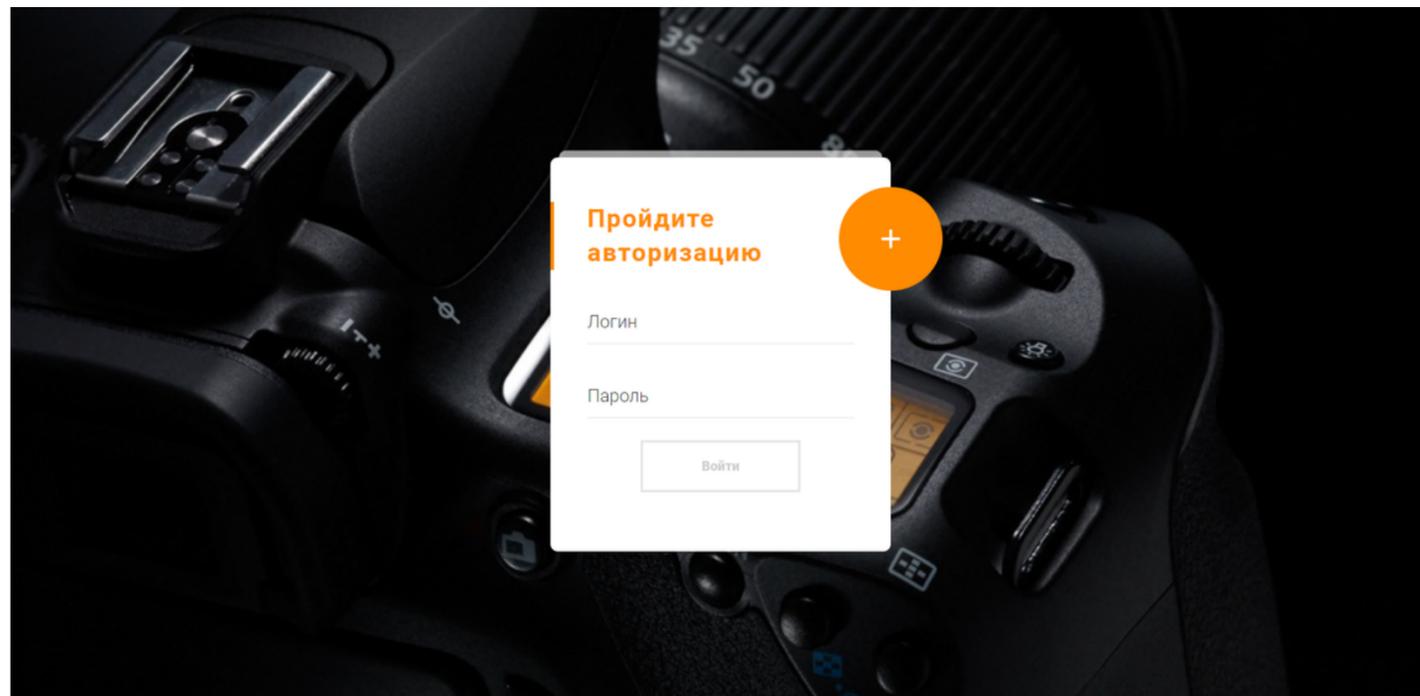
СОЦИАЛЬНЫЕ СЕТИ



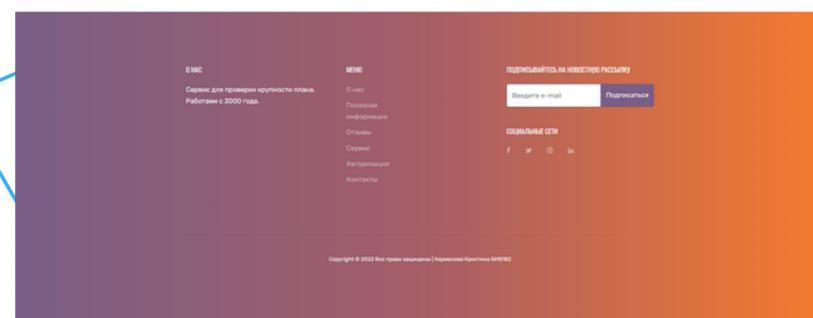
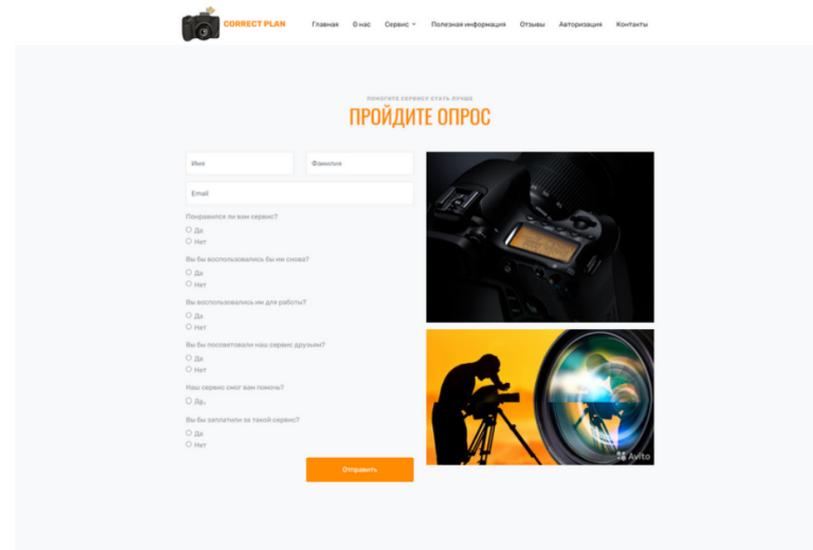
Copyright © 2022 Все права защищены | Карпанова Кристина 618182



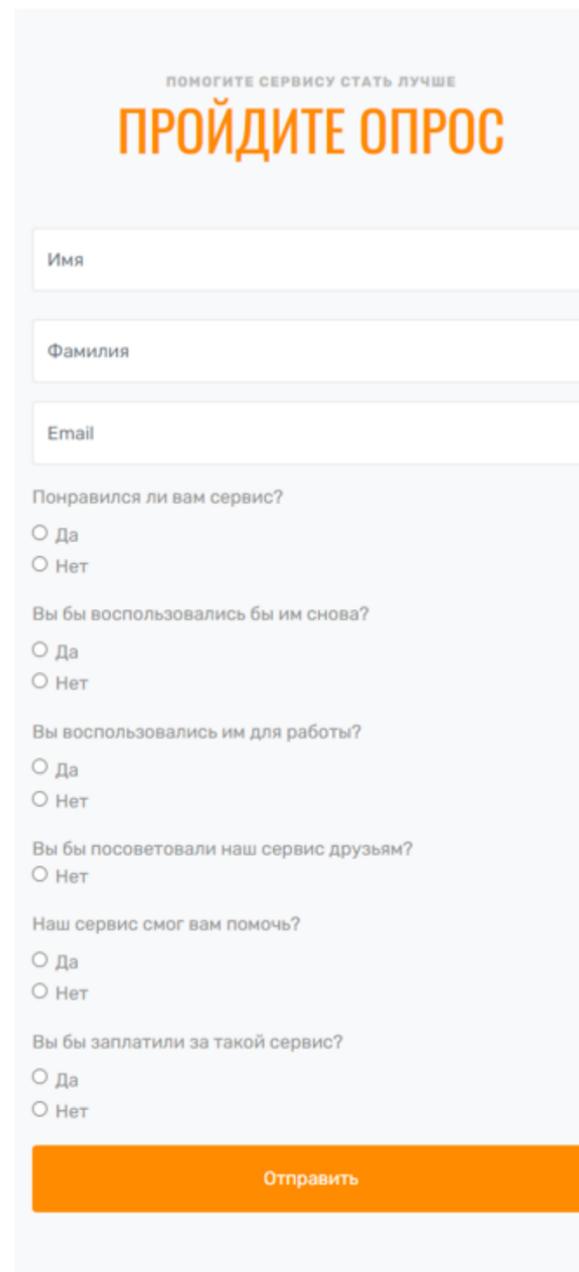
Авторизация и регистрация



Отзывы и статистика



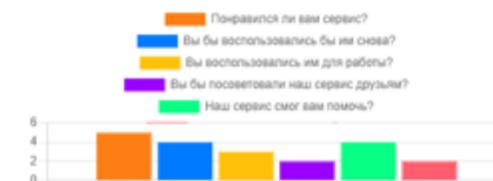
ПК



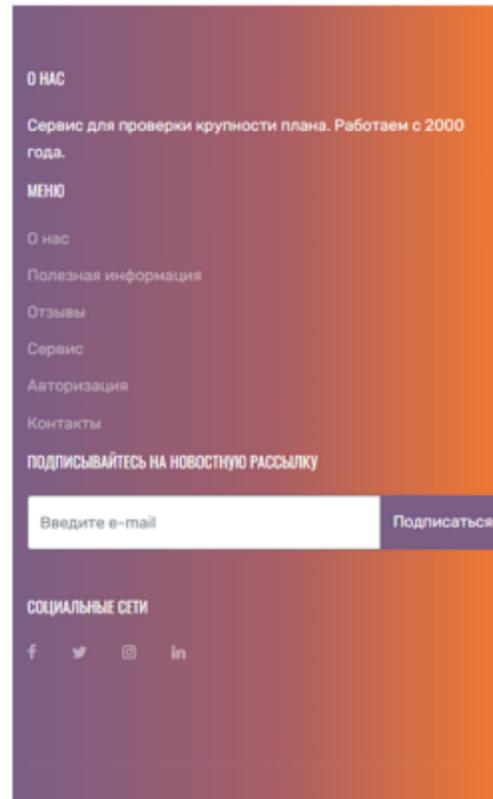
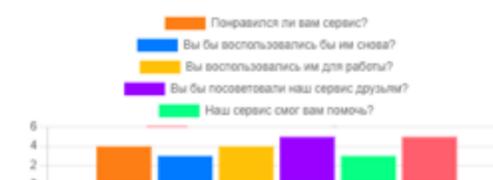
Мобильная версия

ОТЗЫВЫ И СТАТИСТИКА

КОЛИЧЕСТВО ОТВЕТОВ ДА



КОЛИЧЕСТВО ОТВЕТОВ НЕТ



Используемые нейросети



Нейросеть для распознавания лица человека



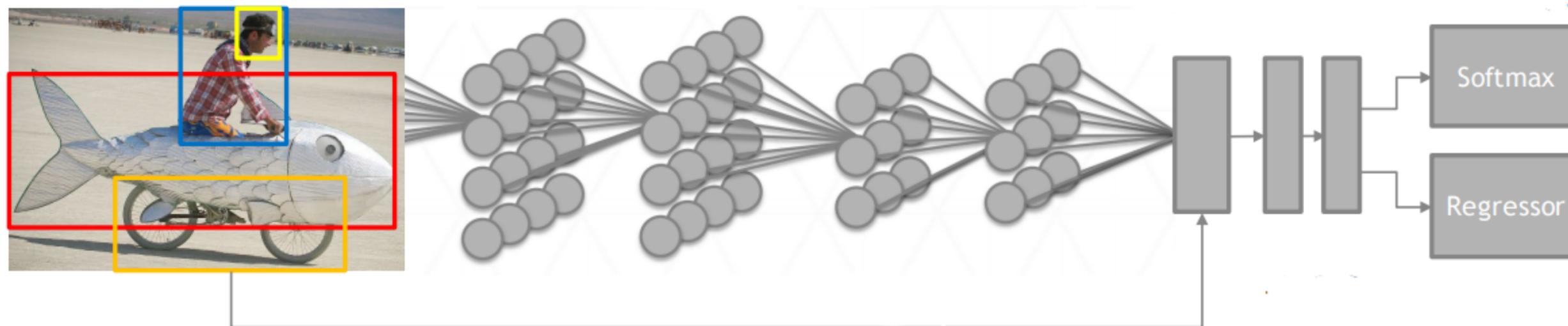
Нейросеть для распознавания тела человека

Структура нейросети

Обе нейросети имеют являются **сверточными нейросетями**. Это специальная архитектура искусственных нейронных сетей, нацеленная на **эффективное распознавание образов**, входит в состав технологий **глубокого обучения**.

Использует некоторые особенности зрительной коры, в которой были открыты так называемые **простые клетки**, реагирующие на **прямые линии** под разными углами, и **сложные клетки**, реакция которых связана с активацией определённого набора простых клеток.

Таким образом, идея свёрточных нейронных сетей заключается в **чередовании свёрточных слоёв и субдискретизирующих слоёв**. Структура сети — **однонаправленная (без обратных связей)**, принципиально многослойная.



Фрагменты описания слоев нейросетей

```
    lr_mult: 0.0
  }
  param {
    lr_mult: 0.0
  }
}
layer {
  name: "data_scale"
  type: "Scale"
  bottom: "data_bn"
  top: "data_bn"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
}
```

**Нейросеть для распознавания
тела человека**

```
layer {
  name: "conv0"
  type: "Convolution"
  bottom: "data"
  top: "conv0"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  param {
    lr_mult: 2.0
    decay_mult: 0.0
  }
}
```

**Нейросеть для распознавания
лица человека**

```
def __init__(self, stream):
    print("INFO: loading model...")
    weights_prototxt_net1 = os.path.join(SRC_DIR, 'backend/api/utils/weights/body_deploy.prototxt.txt')
    weights_caffemodel_net1 = os.path.join(SRC_DIR, 'backend/api/utils/weights/body_deploy.caffemodel')
    self.net1 = cv2.dnn.readNetFromCaffe(weights_prototxt_net1, weights_caffemodel_net1)
    weights_prototxt_net2 = os.path.join(SRC_DIR, 'backend/api/utils/weights/face_deploy.prototxt.txt')
    weights_caffemodel_net2 = os.path.join(SRC_DIR, 'backend/api/utils/weights/face_deploy.caffemodel')
    self.net2 = cv2.dnn.readNetFromCaffe(weights_prototxt_net2, weights_caffemodel_net2)
```

**Использование
моделей с
помощью
OpenCV**

Сбор данных

Чтобы не собирать данные самостоятельно, было принято решение воспользоваться сайтом <https://www.kaggle.com> для поиска подходящих датасетов. Там были найдены данные для обучения нейросети распознавать тела и лица.



```
73 train/392.jpg 0
74 train/393.jpg 0
75 train/394.jpg 0
76 train/395.jpg 0
77 train/396.jpg 0
78 train/397.jpg 0
79 train/398.jpg 0
80 train/399.jpg 0
81 train/420.jpg 1
82 train/421.jpg 1
83 train/422.jpg 1
84 train/423.jpg 1
85 train/424.jpg 1
86 train/425.jpg 1
87 train/426.jpg 1
88 train/427.jpg 1
```

Обучение нейронной сети

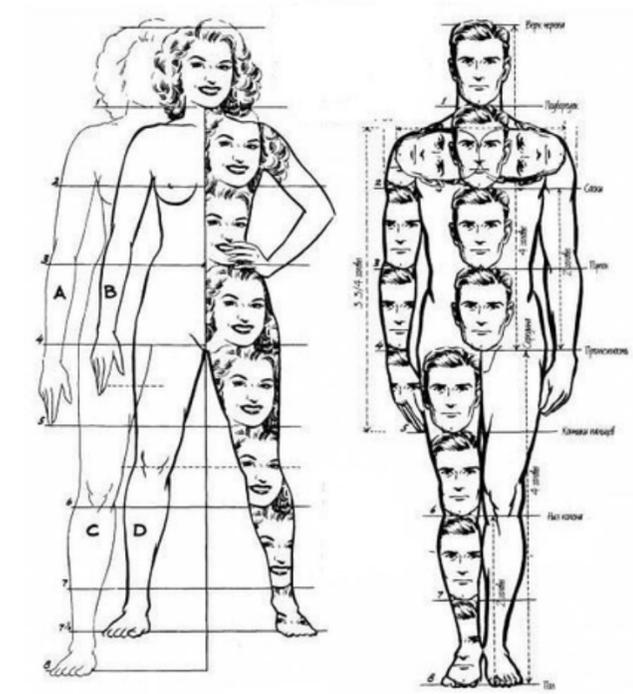
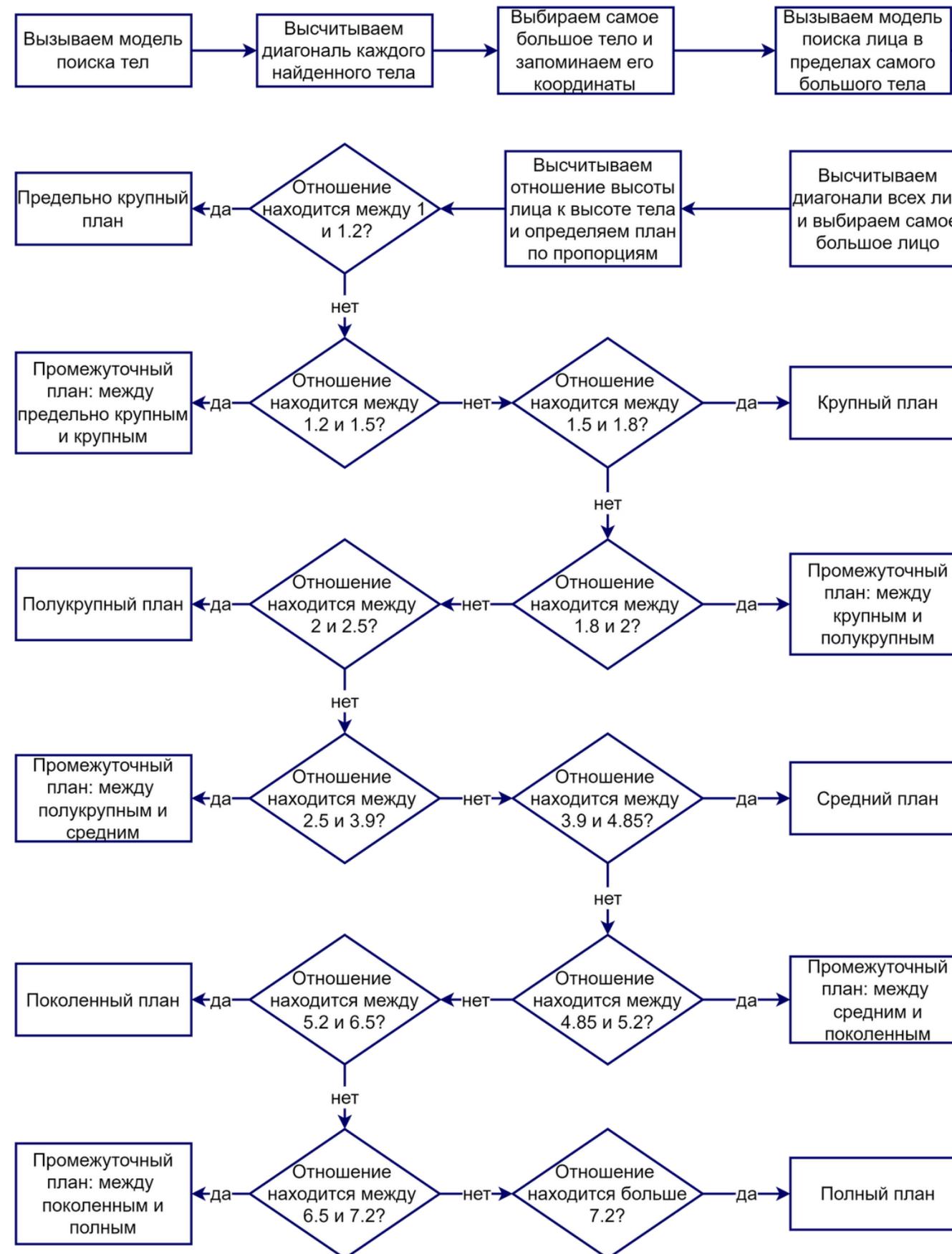
Обучение нейронной сети происходило с помощью caffe.

В текстовом файле были собраны пути до изображений, на которых обучалась сеть.

Каждые 5000 итераций сеть сохранялась, нашим сетям хватило 5000 итераций для того, чтобы получить неплохой результат. Результатом обучения стали файлы моделей с расширением `.caffemodel`.

Алгоритм

Суть алгоритма состоит в следующем: он получает данные обо всех найденных в кадре телах, выбирает самое большое, в его пределах выбирает самое большое лицо и считает отношение высот лица и тела. Далее, используя это отношение и сведения о пропорциях тела человека, программа определяет крупность плана.



Алгоритм



```
def recognize_body(self, img):
    self.x_body, self.y_body, self.x1_body, self.y1_body = [None, None, None, None]
    self.confidence_body = 0
    (h, w) = img.shape[:2]
    blob1 = cv2.dnn.blobFromImage(image_resize(image=img, width=300, height=300),
                                  0.007843, (300, 300), (127.5))

    self.net1.setInput(blob1)
    body = self.net1.forward()

    np_body = np.arange(0, body.shape[2])
    for i in np_body:
        confidence = body[0, 0, i, 2]
        idx = int(body[0, 0, i, 1])
        if confidence > 0.4 and idx == 15:
            box = body[0, 0, i, 3:7] * np.array([w, h, w, h])
            (x_b, y_b, x1_b, y1_b) = box.astype("int")
            x_b, y_b, x1_b, y1_b = abs(x_b), abs(y_b), abs(x1_b), abs(y1_b)
            if x1_b > self.w_vs:
                x1_b = self.w_vs
            if y1_b > self.h_vs:
                y1_b = self.h_vs
            diagonal = len_(x_b, x1_b, y_b, y1_b)
            if self.max_diagonal < diagonal:
                self.max_diagonal = diagonal
                self.confidence_body = confidence
                self.x_body, self.y_body, self.x1_body, self.y1_body = x_b, y_b, x1_b, y1_b

    self.max_diagonal = 0
    if self.x_body:
```

```
        self.confidence_body = confidence
        self.x_body, self.y_body, self.x1_body, self.y1_body = x_b, y_b, x1_b, y1_b

    self.max_diagonal = 0
    if self.x_body:
        self.arr_b.append(self.y1_body - self.y_body)
        cv2.rectangle(img, (self.x_body, self.y_body), (self.x1_body, self.y1_body), (255, 255, 255), 2)
    return self.x_body, self.y_body, self.x1_body, self.y1_body

def recognize_face(self, img):
    self.x_faces, self.y_faces, self.x1_faces, self.y1_faces = [None, None, None, None]
    self.confidence_face = 0
    if self.x_body:
        crop_img = img[abs(self.y_body):self.y_body + (self.y1_body - self.y_body),
                       abs(self.x_body):self.x_body + (self.x1_body - self.x_body)]
    else:
        crop_img = img
    (h, w) = crop_img.shape[:2]
    blob2 = cv2.dnn.blobFromImage(image_resize(image=crop_img, width=300, height=300),
                                  1.0, (300, 300), (104.0, 117.0, 123.0))
```

```
    self.net2.setInput(blob2)
    faces = self.net2.forward()
    np_faces = np.arange(0, faces.shape[2])
    for i in np_faces:
        confidence = faces[0, 0, i, 2]
        if confidence > 0.5:
            box = faces[0, 0, i, 3:7] * np.array([w, h, w, h])
            (x_f, y_f, x1_f, y1_f) = box.astype("int")
            x_f, y_f, x1_f, y1_f = abs(x_f), abs(y_f), abs(x1_f), abs(y1_f)
            if x1_f > self.w_vs:
```

```
                faces = self.net2.forward()
                np_faces = np.arange(0, faces.shape[2])
                for i in np_faces:
                    confidence = faces[0, 0, i, 2]
                    if confidence > 0.5:
                        box = faces[0, 0, i, 3:7] * np.array([w, h, w, h])
                        (x_f, y_f, x1_f, y1_f) = box.astype("int")
                        x_f, y_f, x1_f, y1_f = abs(x_f), abs(y_f), abs(x1_f), abs(y1_f)
                        if x1_f > self.w_vs:
                            x1_f = self.w_vs
                        if y1_f > self.h_vs:
                            y1_f = self.h_vs
                        diagonal = len_(x_f, x1_f, y_f, y1_f)

                        if self.max_diagonal < diagonal:
                            self.max_diagonal = diagonal
                            self.confidence_face = confidence
                            self.x_faces, self.y_faces, self.x1_faces, self.y1_faces = x_f, y_f, x1_f, y1_f
```

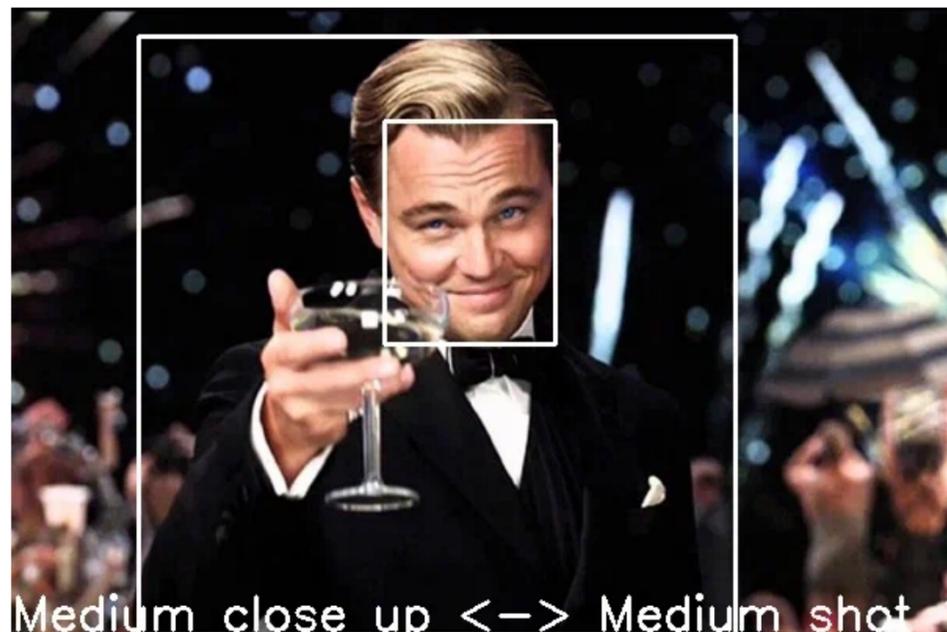
```
    self.max_diagonal = 0
    if self.x_faces:
        self.arr_f.append(self.y1_faces - self.y_faces)
    if self.x_body and self.x_faces:
        cv2.rectangle(img, (self.x_faces + self.x_body, self.y_body + self.y_faces),
                      (self.x1_faces + self.x_body, self.y_body + self.y1_faces), (255, 255, 255), 2)
    elif self.x_faces:
        cv2.rectangle(img, (self.x_faces, self.y_faces), (self.x1_faces, self.y1_faces), (255, 255, 255), 2)
    return self.x_faces, self.y_faces, self.x1_faces, self.y1_faces
```





Эксперименты

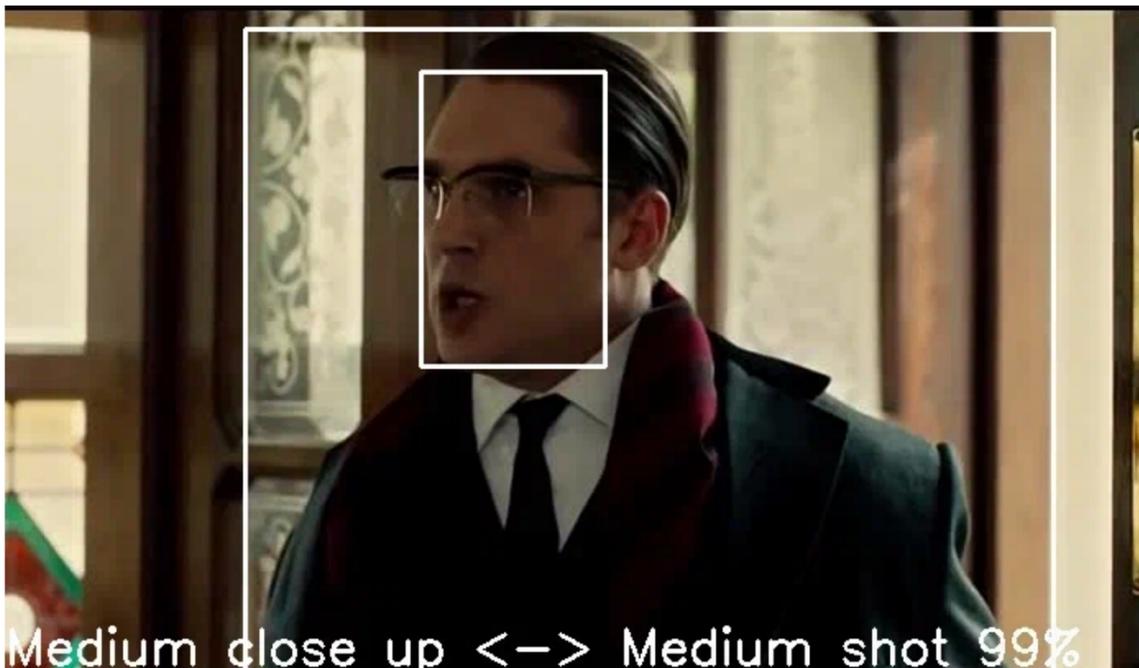
Эксперимент 1



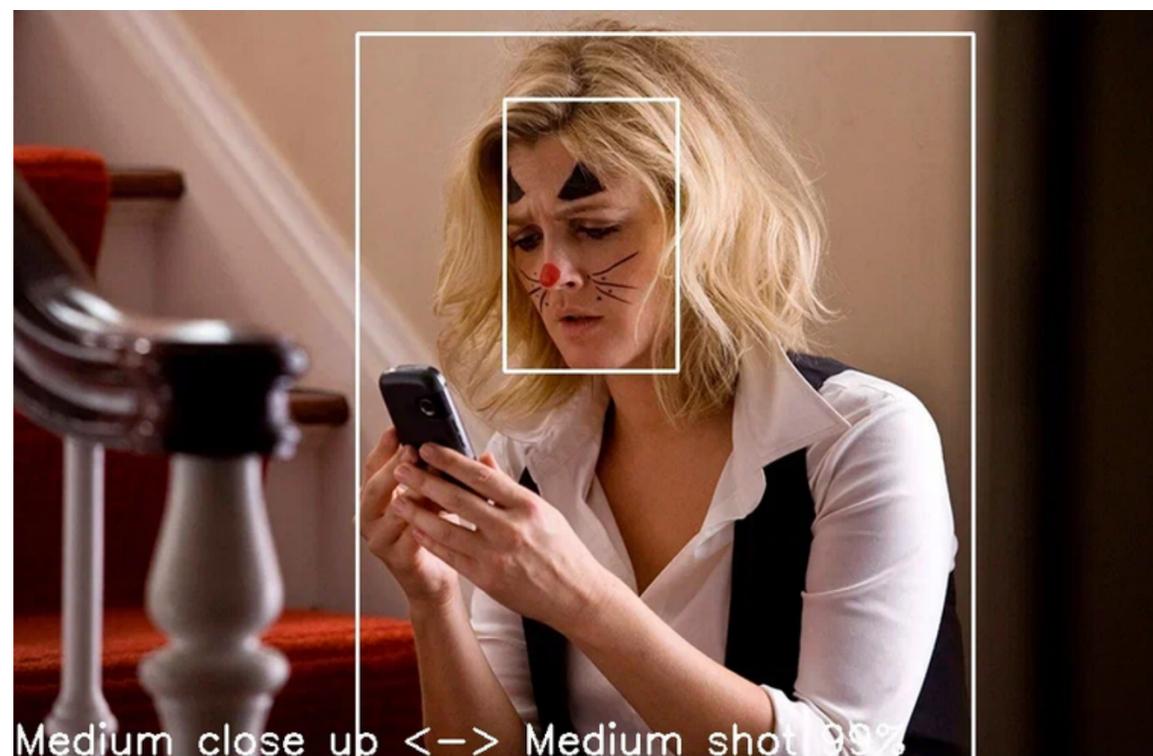
Точность распознавания 100%

	Medium close up Полукрупный / Молочный
	Medium shot Средний / Пасхальный

**Средняя точность
распознавания 99%**



Точность распознавания 99%



Точность распознавания 99%



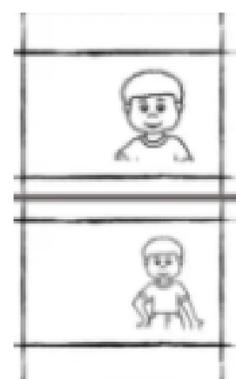


Эксперименты

Эксперимент 2



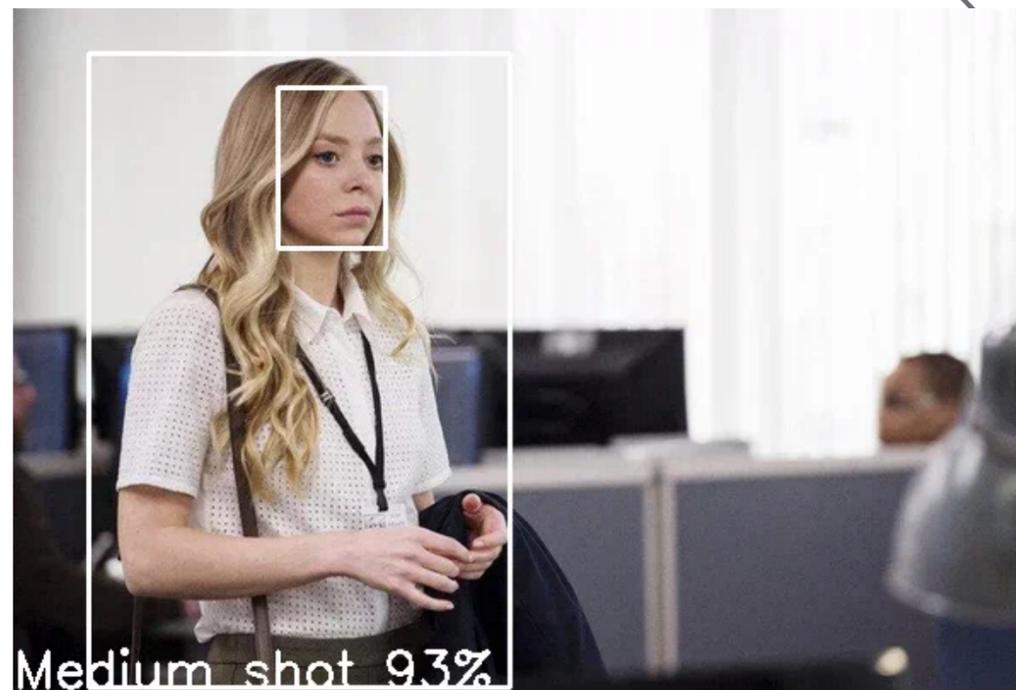
Точность распознавания 98%



Medium close up
Полукрупный / Молочный

Medium shot
Средний / Пасхальный

Средняя точность распознавания 97%



Точность распознавания 93%

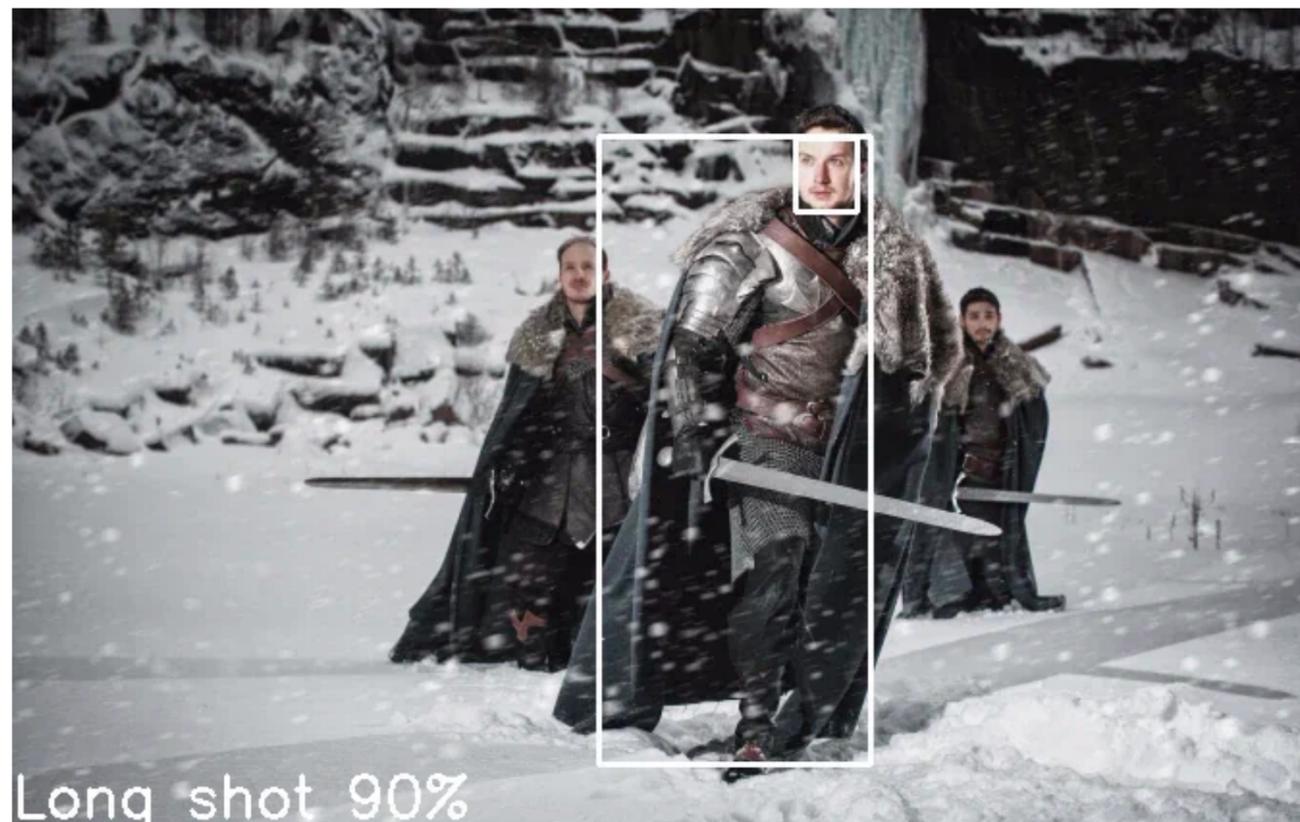


Точность распознавания 100%



Эксперименты

Эксперимент 3



Long shot 90%

Точность распознавания 90%



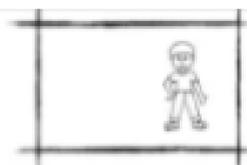
Long shot 100%

Точность распознавания 100%



Long shot 81%

Точность распознавания 81%



Long shot
Полный

Средняя точность распознавания 90%



Эксперименты

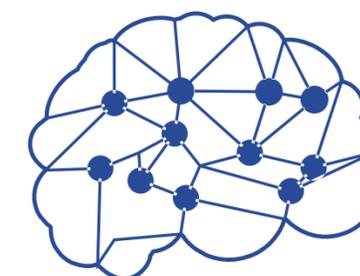
Результаты



▪ Точность распознавания



На основании проведённых экспериментов можно сделать вывод, что нейросеть более точно определяет тела и лица на более крупных планах. Однако даже на полном плане точность распознавания составляет 95%, что является отличным результатом.



Средняя точность распознавания 95%

Дальнейшее развитие



Разработанное приложение может быть доработано:

- Во-первых, можно повысить точность определения тела и лица человека.
- Во-вторых, можно увеличить количество анализируемых лиц и тел.

Итог

В ходе работы выполнено следующее:

- разработано приложение с применением функций искусственного интеллекта, позволяющее определить крупность плана на фотографии и видео
- реализована функция загрузки собственного фото и видео материала в приложение
- проведен ряд экспериментов, на основе которых установлено, что точность работы ИИ составляет порядка 95%
- разработан адаптивный интерфейс веб-страницы, способной обрабатывать запросы пользователей



Спасибо за внимание!

**Теперь перейдём к демонстрации работы
веб-приложения**