



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

# РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

## К КУРСОВОЙ РАБОТЕ

### НА ТЕМУ:

*«Чат-бот для определения возраста и пола по фотографии на основе нейронной сети»*

Студент ИУ7-72Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата) А.Д. Сучков  
(И.О.Фамилия)

Студент ИУ7-72Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата) Н.А. Павлов  
(И.О.Фамилия)

Руководитель курсового проекта

\_\_\_\_\_  
(Подпись, дата) Н.О. Рогозин  
(И.О.Фамилия)

Москва, 2022 г.

**Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

---

УТВЕРЖДАЮ  
Заведующий кафедрой ИУ7  
(Индекс)  
И.В. Рудаков  
(И.О.Фамилия)  
« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

**ЗАДАНИЕ  
на выполнение курсового проекта**

по дисциплине Компьютерные сети

Студент группы ИУ7-72Б

Сучков Александр Дмитриевич  
(Фамилия, имя, отчество)  
Павлов Никита Александрович  
(Фамилия, имя, отчество)

Тема курсового проекта Чат-бот для определения возраста и пола на основе нейронной сети

Направленность КП (учебный, исследовательский, практический, производственный, др.)  
Учебный

Источник тематики (кафедра, предприятие, НИР) кафедра

График выполнения проекта: 25% к 4 нед., 50% к 7 нед., 75% к 11 нед., 100% к 14 нед.

***Задание:***

Разработать чат-бота для определения пола и возраста человека по фотографии на основе нейронной сети. В качестве нейронной сети будет использоваться сверточная нейронная сеть Yolo.

***Оформление курсового проекта:***

Расчетно-пояснительная записка на 20-30 листах формата А4.

Расчетно-пояснительная записка должна содержать постановку введение, аналитическую часть, конструкторскую часть, технологическую часть, экспериментально-исследовательский раздел, заключение, список литературы, приложения.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.): на защиту проекта должна быть представлена презентация, состоящая из 15-20 слайдов. На слайдах должны быть отражены: постановка задачи, использованные методы и алгоритмы, расчетные соотношения, структура комплекса программ, диаграмма классов, интерфейс, характеристики разработанного ПО, результаты проведенных исследований.

Дата выдачи задания « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

**Руководитель курсового проекта**

Н.О. Рогозин  
(Подпись, дата) (И.О.Фамилия)

**Студент**

А.Д. Сучков  
(Подпись, дата) (И.О.Фамилия)

**Студент**

Н.А. Павлов  
(Подпись, дата) (И.О.Фамилия)

# Содержание

|   |    |
|---|----|
| Введение.....   | 4  |
| 1 Аналитическая часть.....                              | 5  |
| 1.1 Формализация задачи.....                            | 5  |
| 1.2 Анализ аналогов .....                               | 5  |
| 1.3 Боты .....  | 5  |
| 1.3.1 Telegram бот .....                                | 5  |
| 1.3.2 WhatsApp бот .....                                | 6  |
| 1.3.3 Viber бот .....                                   | 8  |
| 1.4 Протокол связи HTTP .....                           | 8  |
| 1.5 Сверточные нейронные сети .....                     | 9  |
| 1.6 Сверточная нейронная сеть YOLO .....                | 11 |
| Вывод.....  | 15 |
| 2 Конструкторская часть .....                           | 16 |
| 2.1 Требования к программе.....                         | 16 |
| 2.2 Функциональная модель.....                          | 16 |
| 2.3 Сценарий использования.....                         | 16 |
| 2.4 Структура решения .....                             | 17 |
| 2.5 Сервер .....  | 17 |
| 2.6 Требования к функционалу программного продукта..... | 17 |
| Вывод.....  | 18 |
| 3 Технологическая часть .....                           | 19 |
| 3.1 Выбор технологических средств .....                 | 19 |
| 3.2 Описание функций .....                              | 19 |
| 3.4 Интерфейс программы.....                            | 19 |
| Вывод.....  | 21 |
| Заключение .....  | 22 |
| Список использованных источников .....                  | 23 |

## Введение

В настоящее время множество людей и компаний используют чат-ботов для решения самых разнообразных задач. Их используют, к примеру, в службе поддержки, где клиенты могут задавать часто встречающиеся вопросы по поводу продукции или услуг, или же в качестве автоматических оповещений при наступлении каких-либо событий.

**Целью** данной курсовой работы является создание клиент-серверного приложения, которое определяет пол и возраст человека, изображенного на фотографии, с помощью нейронной сети.

Для достижения поставленной цели, требуется решить следующие **задачи**:

1. рассмотреть существующие типы API мессенджеров;
2. изучить работу нейронных сетей;
3. разработать программное обеспечение, позволяющее осуществить определение возраста и пола по фотографии.

# 1 Аналитическая часть

## 1.1 Формализация задачи

В соответствии с техническим заданием на курсовой проект необходимо разработать чат-бот для мессенджера, который с помощью нейронной сети будет определять возраст и пол человека, изображенного на фотографии.

## 1.2 Анализ аналогов

**Eyeofgod** – инструмент по поиску информации о физических и юридических лицах, использующий для поиска открытые и общедоступные банки данных. Ниже приведены достоинства и недостатки.

Достоинства:

- достаточно быстрая обработка и выдача результата;
- подробный отчет

Недостатки:

- нет анализа пола и возраста человека;
- заблокирован по требованию Роскомнадзора;

## 1.3 Боты

С помощью ботов можно упростить задачу коммуникации с пользователями, создав для них умного помощника. Боты понимают текстовые команды и могут обращаться к API вашего вебсайта, сервиса или быть самостоятельным продуктом с уникальными услугами. Обычно, для создания бота требуется сервер, но существует подход, позволяющий обойтись без него. Он идеален при резких колебаниях количества запросов и прекрасно выдерживает пиковые нагрузки.

### 1.3.1 Telegram бот

Telegram — это быстрое и защищённое от прослушивания приложение для обмена сообщениями. Почти как SMS, однако помимо отправки простых

текстовых сообщений, можно также отправлять фото, видео и любые другие файлы. Ещё можно создавать конференции для группового чата.

Также существенным достоинством является поддержка не только мобильных устройств, но и других платформ (Windows/Mac/Linux).

В отличие от WhatsApp, Telegram работает в облаке, и, кроме того, надёжно зашифровывает весь трафик (в том числе и передаваемые файлы). Благодаря этому, можно пользоваться Telegram-ом на любом устройстве и обмениваться неограниченным количеством фотографий, видео и прочих файлов (doc, zip, mp3, и т.д.). Также, благодаря распределенной инфраструктуре и мощным серверам, Telegram гораздо безопаснее и быстрее, чем любое подобное приложение (WhatsApp, Viber, и т. д.).

Другими словами, вы можете пользоваться Telegram-ом не только на вашем мобильном устройстве, но и на домашнем компьютере, и даже онлайн, чего не может сделать Whatsapp.

Bot API – это интерфейс на основе HTTP, созданный для разработчиков, заинтересованных в создании ботов для Telegram. Этот API позволяет подключать ботов к системе. Боты Telegram – это особые учетные записи, для которых не требуется дополнительный номер телефона. Эти учетные записи служат интерфейсом для кода, выполняемого где-то на вашем сервере [2].

Чтобы использовать это, нет необходимости ничего знать о том, как работает протокол шифрования MTProto – промежуточный сервер будет обрабатывать все шифрование и связь с Telegram API за вас. Общение с сервером происходит через простой HTTPS-интерфейс, который предлагает упрощенную версию Telegram API [3].

### **1.3.2 WhatsApp бот**

Чат-боты для популярных мессенджеров стали в последнее время одним из распространенных CRM-каналов. Однако самый популярный мессенджер оставался немного в стороне: ботов делают для Viber, Telegram, Facebook

Messenger и VK, а вот для WhatsApp их существенно меньше. Тем не менее, у него более 27 млн. пользователей в месяц, 59% из которых ежедневно используют мессенджер на смартфоне.

Все существующие мессенджеры делятся на два типа – с открытым и закрытым API. В мессенджерах «ВКонтакте», Telegram, Facebook, Viber открытый способ подключения. Подключиться к ним можно без проблем. В WhatsApp используют закрытый API.

К WhatsApp невозможно присоединиться также, как к другим мессенджерам. Закрытый API – официальная политика компании, направленная на сохранение деловой репутации мессенджера. В WhatsApp не хотят идти по пути Viber, который используют для рассылки спама мошенники и тысячи мелких компаний.

Существует два способа взаимодействия с WhatsApp: незаконный – воровство интерфейса WhatsApp, и законный – подключение к WhatsApp Business API. Чтобы начать общение с клиентами через WhatsApp, компании необходимо получить доступ к API WhatsApp Business, а затем получить подтвержденный (Verified) бизнес-аккаунт. Чтобы пройти проверку, нужно указать имя бренда и отправить аккаунт на модерацию. Регистрация бизнесаккаунта занимает от трех недель до одного месяца. После подтверждения регистрации аккаунту присваивается статус Verified и устанавливается зеленая галочка возле номера телефона, которые видят пользователи вместо имени — остается теперь написать код бота под требуемые задачи.

Чат-боты в WhatsApp решают практически те же самые задачи, что и в других мессенджерах (Viber, Telegram, Facebook Messenger). Бот может отправлять ссылки на продукты компании, эмодзи, аудиофайлы, изображения, геолокацию. У пользователей в свою очередь есть возможность использовать как готовые кнопки ответов, так и общаться на естественном языке. В отличие от других мессенджеров, в WhatsApp нельзя делиться стикерами и «юзерлинками» («юзерлинки» — ссылки в тексте ответа чат-бота, при нажатии на которые идёт

отправка фразы с текстом ссылки так, будто сам пользователь написал и отправил текст этой ссылки).

### **1.3.3 Viber бот**

Публичные аккаунты в Вайбере позволяют представителям бизнеса, компаниям и прочим известным личностям взаимодействовать со своей аудиторией. Чаты могут использоваться в широком спектре задач, начиная от маркетинговых рассылок, заканчивая полноценными ботами, позволяющими создавать заказы в магазине.

## **1.4 Протокол связи HTTP**

Выше было решено использовать чат-бота для реализации приложения взаимодействия с нейронной сетью, следовательно в качестве протокола связи выбран HTTP (HyperText Transfer Protocol) – протокол прикладного уровня передачи произвольных данных. Задача, которая традиционно решается с помощью этого протокола — обмен данными между пользовательским приложением, осуществляющим доступ к веб ресурсам и веб-сервером. API многих программных продуктов также подразумевает использование HTTP для передачи данных.

Все ПО для работы с HTTP разделяется на:

- клиенты — отправляют запросы;
- серверы — обрабатывают запросы;
- прокси — посредники для выполнения транспортных служб.

Каждое HTTP сообщение состоит из трех частей:

- стартовая строка, определяющая тип сообщения;
- заголовки, характеризующие тело сообщения и прочие сведения;
- тело сообщения — данные.



## 1.5 Сверточные нейронные сети

Нейронная сеть — попытка с помощью математических моделей воспроизвести работу человеческого мозга для создания машин, обладающих искусственным интеллектом.

Схема нейронной сети изображена на рисунке 1.1 [1].

Искусственная нейронная сеть состоит из трех компонентов:

- входной слой;
- скрытые (вычислительные) слои;
- выходной слой.

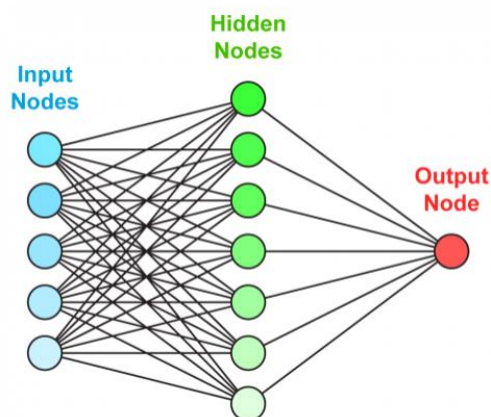


Рисунок 1.1 – схема нейронной сети

Сверточная нейронная сеть (Convolutional Neural Network — ConvNet/CNN) — это Deep Learning-алгоритм, который может принимать входное изображение, присваивать важность (усваиваемые веса и смещения) различным областям/объектам в изображении и может отличать одно от другого.

Глубокое обучение (deep learning) – это класс алгоритмов машинного обучения, которые учатся глубже (более абстрактно) понимать данные. Популярные алгоритмы нейронных сетей глубокого обучения представлены на рисунке 1.2 [1].

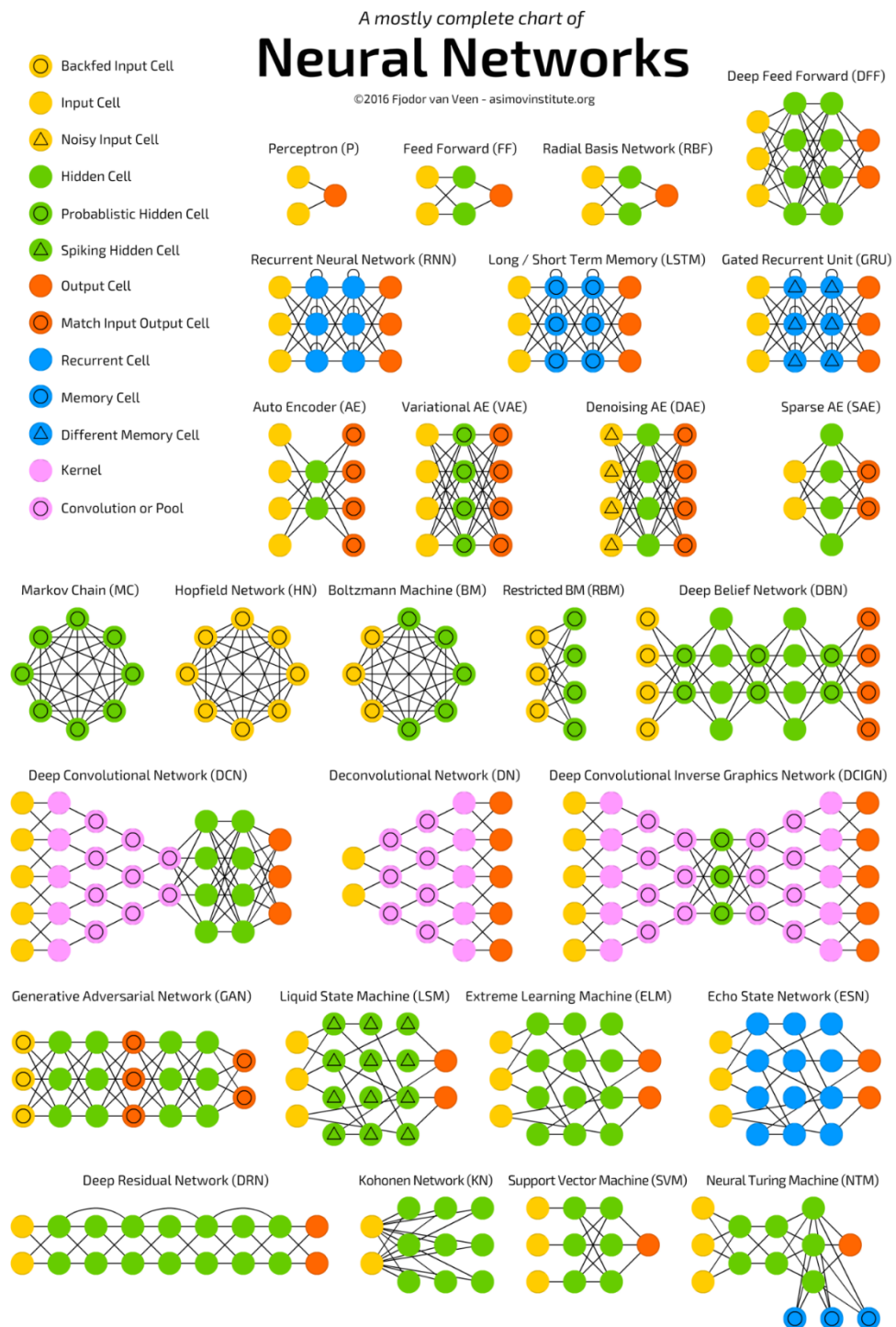


Рисунок 1.2 – схемы нейронных сетей глубокого обучения

Более формально в deep learning:

- используется каскад (пайплайн, как последовательно передаваемый поток) из множества обрабатывающих слоев (нелинейных) для извлечения и преобразования признаков;

- основывается на изучении признаков (представлении информации) в данных без обучения с учителем. Функции более высокого уровня 10 (которые находятся в последних слоях) получаются из функций нижнего уровня (которые находятся в слоях начальных слоев);
- изучает многоуровневые представления, которые соответствуют разным уровням абстракции; уровни образуют иерархию представления.

Сверточный вид нейронных сетей наиболее удобен для обработки изображений, поэтому решено использовать именно их для достижения поставленной цели, а именно нейронную сеть YOLO.

## 1.6 Сверточная нейронная сеть YOLO

YOLO (You only look once) – это система обнаружения объектов, предназначенная для обработки в реальном времени.

YOLO делит входное изображение на сетку  $S \times S$ . Каждая ячейка сетки предсказывает фиксированное количество граничных прямоугольников. Из рисунка 2.3 видно, что желтая ячейка сетки делает два прогноза для граничного прямоугольника (синие прямоугольники), чтобы определить местонахождение человека [4].

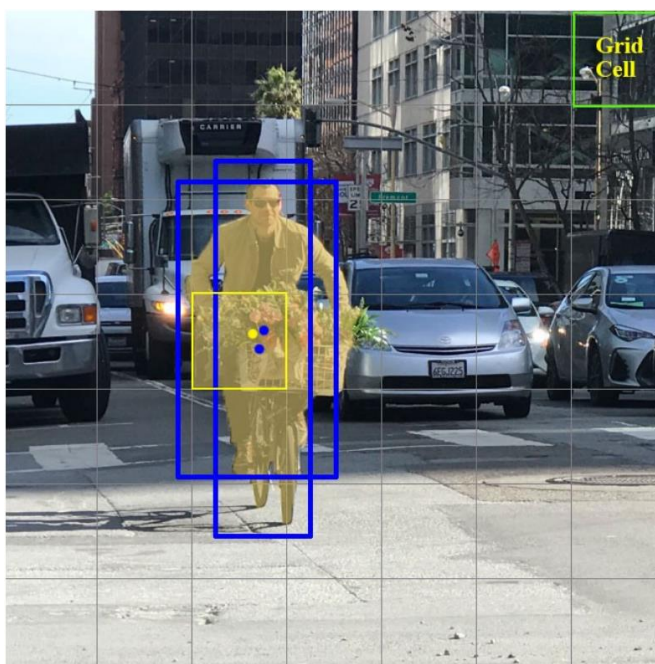


Рисунок 2.3 – ячейки YOLO

Для каждой ячейки сетки:

- прогнозирует  $B$  граничных прямоугольников, и каждый прямоугольник имеет один балл достоверности;
- обнаруживает только один объект вне зависимости от количества прямоугольников  $B$ ;
- предсказывает вероятности условного класса  $C$  (по одной на класс для вероятности класса объекта);
- для оценки PASCAL VOC в YOLO используются сетки  $7 \times 7$  ( $S \times S$ ), 2 граничных прямоугольника ( $B$ ) и 20 классов ( $C$ ).

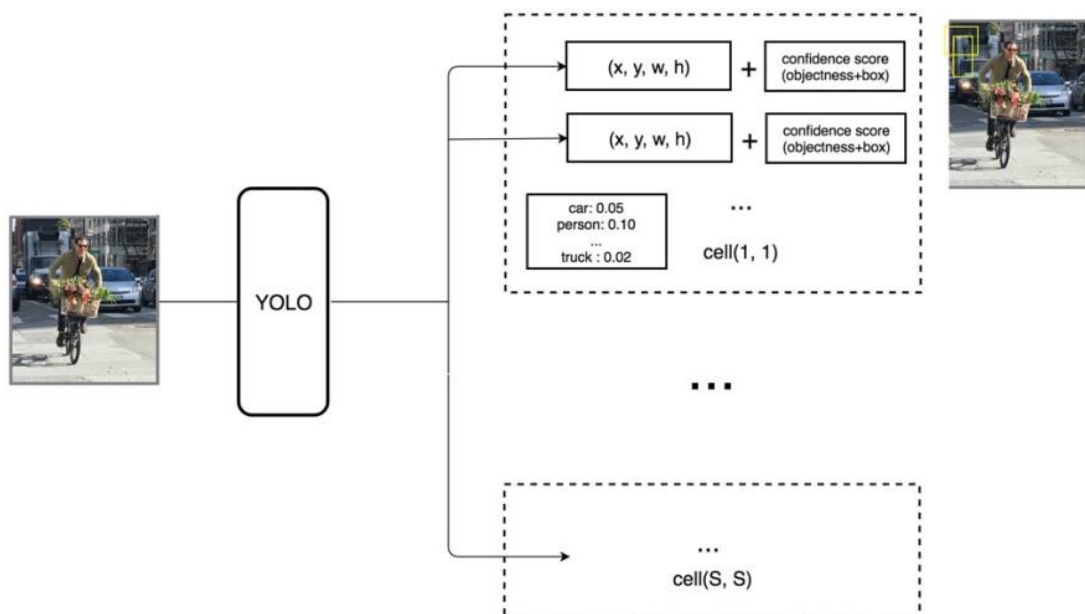


Рисунок 2.4 – YOLO делает прогнозы  $S \times S$  с граничными блоками  $B$

Каждый ограничивающий прямоугольник содержит 5 элементов:  $(x, y, w, h)$  и показатель достоверности прямоугольника. Оценка достоверности отражает, насколько вероятно, что блок содержит объект (объектность) и насколько точен ограничивающий блок. Мы нормализуем ширину ограничивающего прямоугольника  $w$  и высоту  $h$  на ширину и высоту изображения. При этом  $x$  и  $y$  - смещения к соответствующей ячейке. Следовательно,  $x, y, w$  и  $h$  находятся в диапазоне от 0 до 1. Каждая ячейка имеет 20 условных вероятностей класса. Вероятность условного класса – это вероятность того, что обнаруженный объект принадлежит определенному классу

(одна вероятность на категорию для каждой ячейки). Итак, прогноз YOLO имеет форму  $(S, S, B \times 5 + C) = (7, 7, 2 \times 5 + 20) = (7, 7, 30)$  (пример изображен на рисунке 2.5).

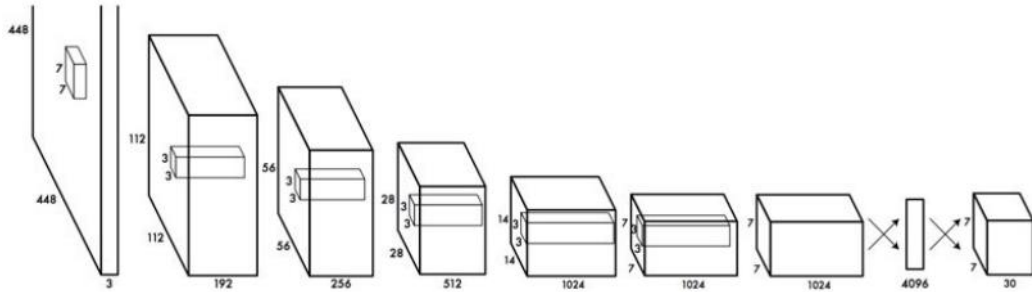


Рисунок 2.5 – YOLO делает прогнозы  $S \times S$  с граничными блоками  $B$

Оценка достоверности класса для каждого блока прогнозов рассчитывается как произведение оценки достоверности прямоугольника и условной вероятности класса.

YOLO прогнозирует несколько ограничивающих рамок для каждой ячейки сетки. Чтобы вычислить потери для истинного положительного результата, мы хотим, чтобы только один из них отвечал за объект. Каждое предсказание становится лучше при предсказании определенных размеров и соотношений сторон. YOLO использует сумму квадратов ошибок между предсказаниями и истиной для расчета потерь [4].

Функция потерь состоит из:

- потеря классификации;
- потеря локализации;
- потеря уверенности;

Потеря классификации:

Если объект обнаружен, потеря классификации в каждой ячейке является квадратом ошибки условных вероятностей класса для каждого класса:

$$\sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

### Потеря локализации:

Потеря локализации измеряет ошибки в предсказанных местоположениях и размерах граничных рамок [5]. Мы считаем только рамку, отвечающую за обнаружение объекта.

$$\begin{aligned} & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[ \left( \sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left( \sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \end{aligned}$$

### Потеря уверенности:

Если объект обнаружен в рамке, потеря уверенности (измерение объектности рамки) составляет:

$$\sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2$$

Если объект не обнаружен в рамке, потеря уверенности составляет:

$$\lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2$$

Окончательная потеря добавляет вместе потери локализации, достоверности и классификации:

$$\begin{aligned} & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[ \left( \sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left( \sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\ & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \\ & + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \end{aligned}$$

## **Вывод**

В данном разделе была проанализирована поставленная задача и рассмотрены существующие решения. Рассмотрены достоинства и недостатки самых популярных ботов в мессенджерах. Также в ходе сравнительного анализа представленных инструментов был выбран Telegram бот в связи со всеми вышеперечисленными преимуществами. В качестве протокола связи выбран протокол HTTP. Рассмотрены сверточные нейронные сети и нейронная сеть YOLO.

## 2 Конструкторская часть

### 2.1 Требования к программе

Приложение должно предоставлять следующие возможности:

- команда для начальной инициализации пользователя;
- команда для вывода инструкции;
- обращение к серверу для обработки фотографии;
- отображение результатов анализа, полученных от сервера.

### 2.2 Функциональная модель

На рисунке 2.1 изображена функциональная модель IDEF0 нулевого уровня.

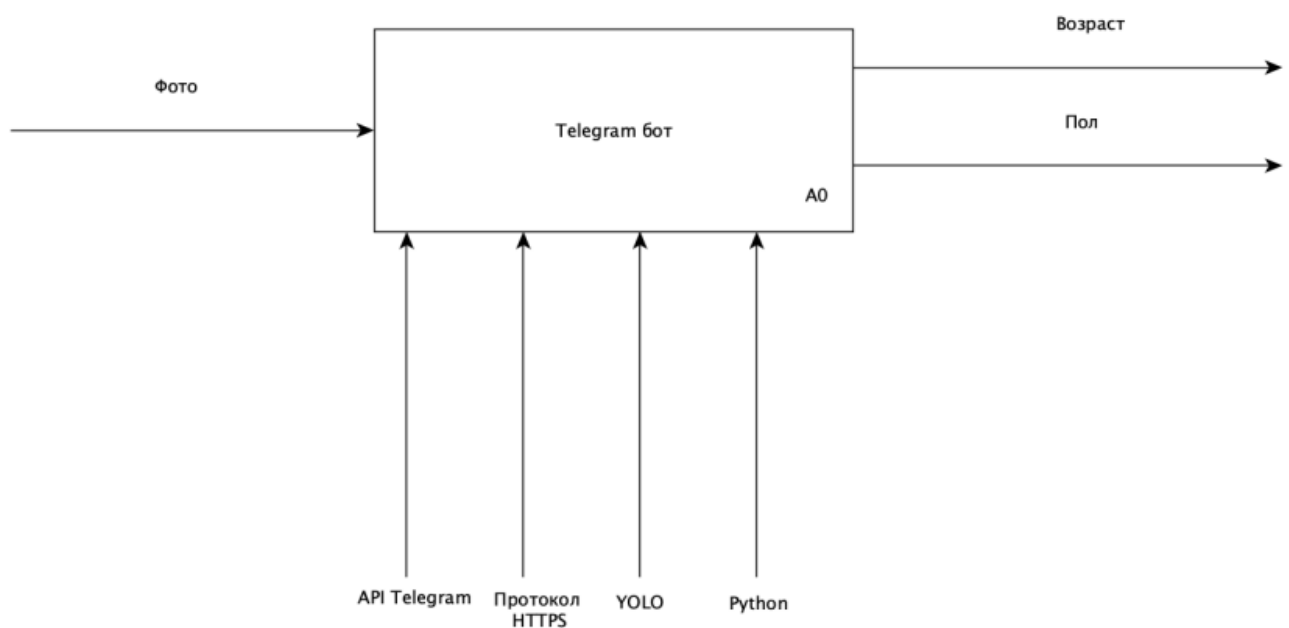


Рисунок 2.1 – IDEF0 нулевого уровня

### 2.3 Сценарий использования

На рисунке 2.2 представлена диаграмма взаимодействия пользователя с приложением.





Рисунок 2.2 – use-case диаграмма

## 2.4 Структура решения

Предполагается использование архитектуры «Клиент — сервер» — вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между так называемыми серверами, и заказчиками услуг — клиентами.

Программы-серверы ожидают от клиентских программ запросы и предоставляют им свои ресурсы в виде данных (например, загрузка файлов посредством HTTP, FTP) или в виде сервисных функций.

## 2.5 Сервер

Сервер должен удовлетворять следующим требованиям:

- обрабатывать, последовательно полученные запросы;
- наличие обработчика запроса на обработку фотографии;
- предоставлять результаты обработки и анализа фотографии.

## 2.6 Требования к функционалу программного продукта

Для взаимодействия с нейронной сетью используется следующая спецификация протокола http:

- инициализация пользователя (реализуется командой /start);
- просмотр инструкции (реализуется командой /help);
- отправка фотографии на обработку и вывод результатов (реализуется командой /photo).

## **Вывод**

В данном разделе были проанализированы требования к программе. Приведены функциональная модель, сценарий использования приложений. Выбрана клиент-серверная архитектура и сформулированы требования к функционалу программного продукта.

## 3 Технологическая часть

### 3.1 Выбор технологических средств

В качестве языка программирования для реализации данного курсового проекта был выбран язык Python по причине совмещения нескольких парадигм программирования, а также из-за того, что на языке Python представлена библиотека telebot для взаимодействия с API Telegram.

В качестве среды разработки использован Visual Studio Code, так как с его помощью удобно работать с python файлами. Среда обладает кроссплатформенностью и большим выбором настроек проекта.

### 3.2 Описание функций

В приложении А приведены листинги реализаций основных методов.

Кратко опишем функции, выполняемые реализованным чат-ботом:

- **getStartCommand** – метод служит для перехвата команды “/start” от пользователя, после чего происходит приветствие;
- **getHelpCommand** – метод предназначен для перехвата команды “/help”, пользователю выводится инструкция по использованию;
- **getPhotoMessage** – метод служит для обработки отправленных пользователем фотографий, где далее происходит основная логика запуска нейронной сети и дальнейшая обработка результатов;
- **getTextMessage** – метод отвечает за перехват текстовых сообщений с последующими ответами на них.

### 3.4 Интерфейс программы

На рисунках 3.1 – 3.4 представлен пользовательский интерфейс.

Для того, чтобы начать использование бота, необходимо нажать на кнопку или прописать команду “/start”, после чего произойдёт приветствие. Далее, пользователь может загрузить изображение.

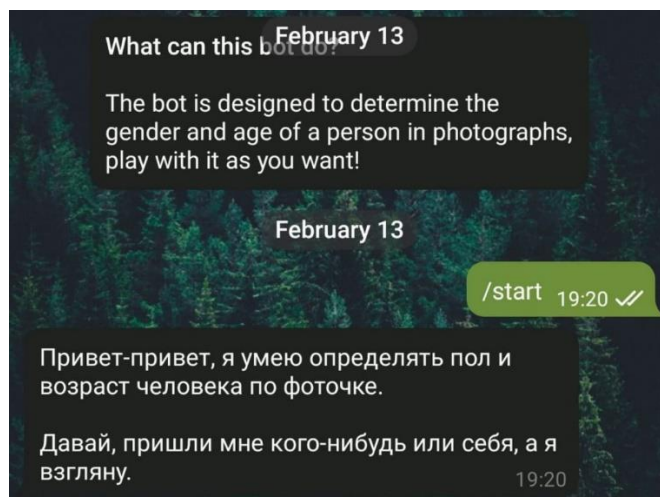


Рисунок 3.1 – пример работы команды “/start”

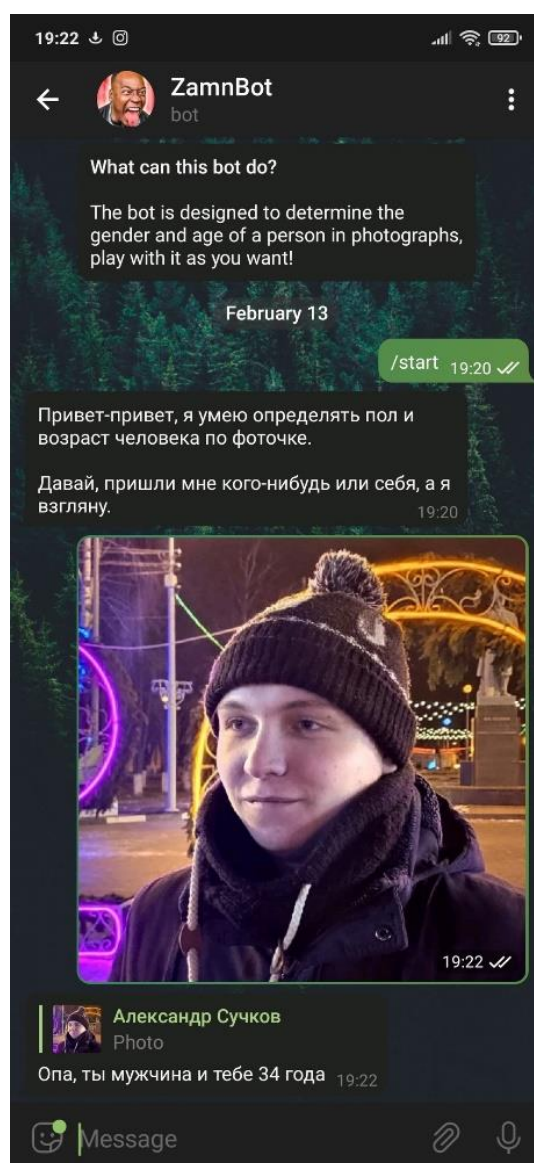


Рисунок 3.2 – пример обработки фотографии с человеческим лицом

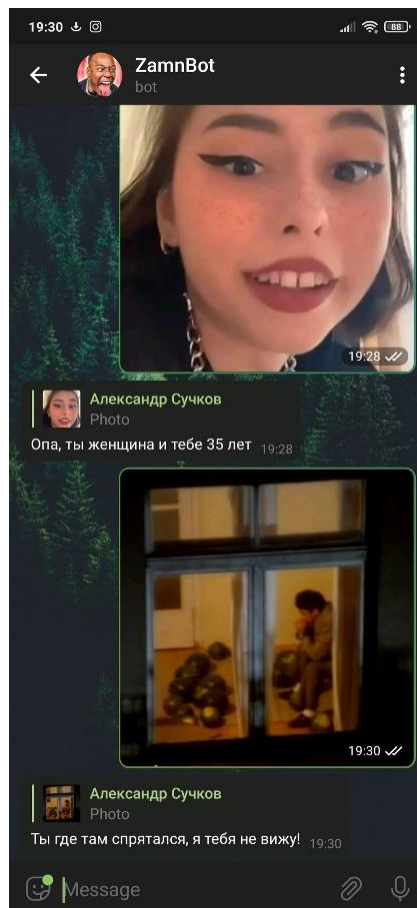


Рисунок 3.3 – пример обработки фотографии без человеческого лица

В случае, если пользователь хочет ознакомиться с функционалом, предусмотрена команда “/help” для вывода инструкции к использованию (рисунок 3.4).

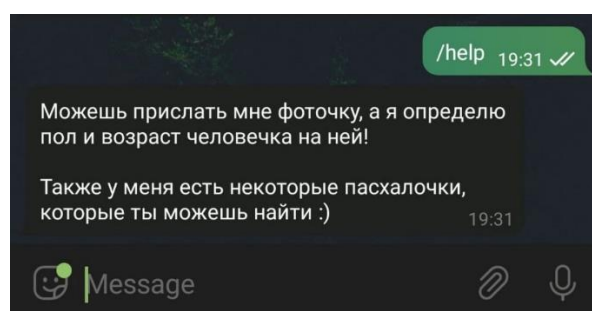


Рисунок 3.4 – пример вывода инструкции к использованию

## Вывод

В данном разделе был выбран язык Python в качестве языка программирования, на котором реализована поставленная задача, выбран Visual Studio Code в качестве среды разработки, а также приведен интерфейс для взаимодействия с пользователем.

## **Заключение**

В ходе выполнения данной курсовой работы была достигнута цель, рассмотрены существующие аналоги, а также выбран вид и протокол взаимодействия с нейронной сетью. Был разработан Telegram чат-бот, предоставляющий пользователю интерфейс для работы с нейронной сетью.

Программа реализована таким образом, что пользователь может задавать команды для инициализации, вывода инструкции к использованию, а также для обработки фотографий нейронной сетью и получения пола и возраста изображенного на ней человека.

В ходе выполнения поставленной задачи были изучены возможности Telegram чат-ботов и сверточной нейронной сети YOLO. Получен опыт работы с протоколами связи и знания в области компьютерных сетей.

В качестве улучшения проекта предполагается дальнейшее обучение модели для получения более точных результатов, а также добавление распознавания эмоций на лице человека.

## **Список использованных источников**

1. Как работает нейронная сеть: алгоритмы, обучение, функции активации и потери. [Электронный ресурс] – Режим доступа: <https://neurohive.io/ru/osnovy-data-science/osnovy-nejronnyh-setej-algoritmy-obuchenie-funkcii-aktivacii-i-poteri/> (дата обращения: 05.02.2022).
2. Telegram API Bot. [Электронный ресурс] – Режим доступа: <https://core.telegram.org/bots/api> (дата обращения: 10.02.2022).
3. MTProto Mobile Protocol. [Электронный ресурс] – Режим доступа: <https://core.telegram.org/mtproto> (дата обращения: 10.02.2022).
4. Real-time Object Detection with YOLO, YOLOv2 and now YOLOv3. [Электронный ресурс] – Режим доступа: <https://jonathan-hui.medium.com/real-time-object-detection-with-yolo-yolov2-28b1b93e2088> (дата обращения: 12.02.2020).
5. Нейронные сети для начинающих. Часть 1. [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/312450/> (дата обращения: 10.02.2022).