



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 мессенджера Telegram с поддержкой
определения возраста и пола по фотографии на
основе нейронной сети***

Студент _____
ИУ7-75Б
(Группа)

(Подпись, дата) А.А. Лаврова
(И.О.Фамилия)

Руководитель курсового проекта

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

2020 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 курсовой работы

по дисциплине Компьютерные сети

Студент группы ИУ7-75Б

Лаврова Анастасия Андреевна
(Фамилия, имя, отчество)

Тема курсового проекта Чат-бот для мессенджера Telegram с поддержкой определения возраста
и пола по фотографии на основе нейронной сети

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

Источник тематики (кафедра, предприятие, НИР) _____

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

Задание Разработать чат-бот для мессенджера Telegram, который с помощью нейронной
сети будет определять возраст и пол человека, изображенного на фотографии. В качестве
нейронной сети будет использоваться сверточная нейронная сеть Yolo.

Оформление курсового проекта:

Расчетно-пояснительная записка на 25-30 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

На защиту проекта должна быть представлена презентация. На слайдах должны быть:
постановка задачи, использованные методы и алгоритмы, расчётные соотношения,
структура комплекса программ, диаграмма классов, интерфейс, характеристики ПО,
результаты исследования.

Дата выдачи задания « ____ » _____ 20__ г.

Руководитель курсового проекта

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

Студент

А. А. Лаврова
(Подпись, дата) (И.О.Фамилия)

СОДЕРЖАНИЕ

РЕФЕРАТ.....	4
ВВЕДЕНИЕ	5
1. Аналитический раздел.....	6
1.1 Формализация задачи.....	6
1.2 Анализ предметной области (клиент-серверное приложение)	6
1.3 Нейронная сеть	7
1.4 Боты	10
1.4.1 Telegram bot	10
1.4.2 WhatsApp bot	11
1.4.3 Viber bot	12
1.5 Вывод.....	13
2. Конструкторский раздел	14
2.1 Функциональная модель	14
2.2 Сценарий использования	14
2.3 Нейросеть YOLO	14
2.4 Вывод.....	18
3. Технологический раздел	19
3.1 Выбор языка программирования	19
3.2 Выбор среды разработки	19
3.3 Описание некоторых моментов реализации	19
3.4 Интерфейс пользователя.....	20
3.5 Вывод.....	23
ЗАКЛЮЧЕНИЕ	24
Список используемых источников	25

РЕФЕРАТ

Данная работа посвящена разработке чат-бота для мессенджера Telegram с поддержкой определения возраста и пола по фотографии на основе нейронной сети.

Расчетно-пояснительная записка содержит 25 стр., 12 рис., 2 листинга, 5 источников.

ВВЕДЕНИЕ

После бурных обсуждений в ИТ-прессе по поводу эффективности чат-ботов, они заняли свою нишу в экосистеме пользователей и компаний. Например, часто проекты внедряют ботов для оповещения о каких-либо событиях, а службы поддержки используют их для того, чтобы быстро отвечать на часто задаваемые вопросы клиентов.

Целью данной курсовой работы является создание клиент-серверного приложения, которое определяет пол и возраст человека по фотографии.

Чтобы достигнуть поставленной цели, требуется решить следующие задачи:

- формализовать задачу;
- рассмотреть существующие типы API социальных сетей;
- изучить работы нейронных сетей;
- разработать программное обеспечение, позволяющее осуществить определение возраста и пола по фотографии с использованием нейронной сети и бота социальной сети.

1. Аналитический раздел

В данном разделе производится постановка задачи и анализ методов решения поставленной задачи.

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 предметной области (клиент-серверное приложение)

«Клиент — сервер» (англ. client-server) — вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами. Фактически клиент и сервер — это программное обеспечение. Обычно эти программы расположены на разных вычислительных машинах и взаимодействуют между собой через вычислительную сеть посредством сетевых протоколов, но они могут быть расположены также и на одной машине. Программы-серверы ожидают от клиентских программ запросы и предоставляют им свои ресурсы в виде данных (например, загрузка файлов посредством HTTP, FTP, BitTorrent, потоковое мультимедиа или работа с базами данных) или в виде сервисных функций (например, работа с электронной почтой, общение посредством систем мгновенного обмена сообщениями или просмотр web-страниц во всемирной паутине). Поскольку одна программа-сервер может выполнять запросы от множества программ-клиентов, её размещают на специально выделенной вычислительной машине, настроенной особым образом, как правило, совместно с другими программами-серверами, поэтому производительность этой машины должна быть высокой. Из-за особой роли такой машины в сети, специфики её оборудования и программного

обеспечения, её также называют сервером, а машины, выполняющие клиентские программы, соответственно, клиентами.

1.3 Нейронная сеть

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

Искусственная нейронная сеть обычно обучается с учителем. Это означает наличие обучающего набора (датасета), который содержит примеры с истинными значениями: тегами, классами, показателями.

Например, при попытке создать нейросеть для оценки тональности текста, датасетом будет список предложений с соответствующими каждому эмоциональными оценками. Тональность текста определяют признаки (слова, фразы, структура предложения), которые придают негативную или позитивную окраску. Веса признаков в итоговой оценке тональности текста (позитивный, негативный, нейтральный) зависят от математической функции, которая вычисляется во время обучения нейронной сети.

Раньше генерация признаков производится вручную. Чем больше признаков и точнее подобраны веса, тем точнее ответ. Нейронная сеть автоматизировала этот процесс. Схема нейронной сети изображена на рисунке 1.1 [1].

Искусственная нейронная сеть состоит из трех компонентов:

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

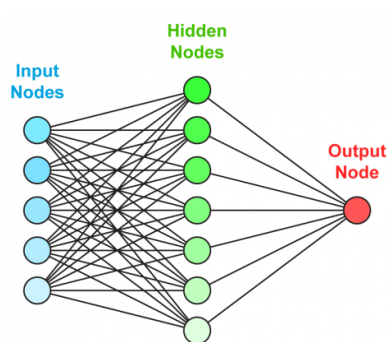


Рисунок 1.1 – схема нейронной сети

Обучение нейросетей происходит в два этапа:

- прямое распространение ошибки;
- обратное распространение ошибки.

Во время прямого распространения ошибки делается предсказание ответа. При обратном распространении ошибка между фактическим ответом и предсказанным минимизируется.

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

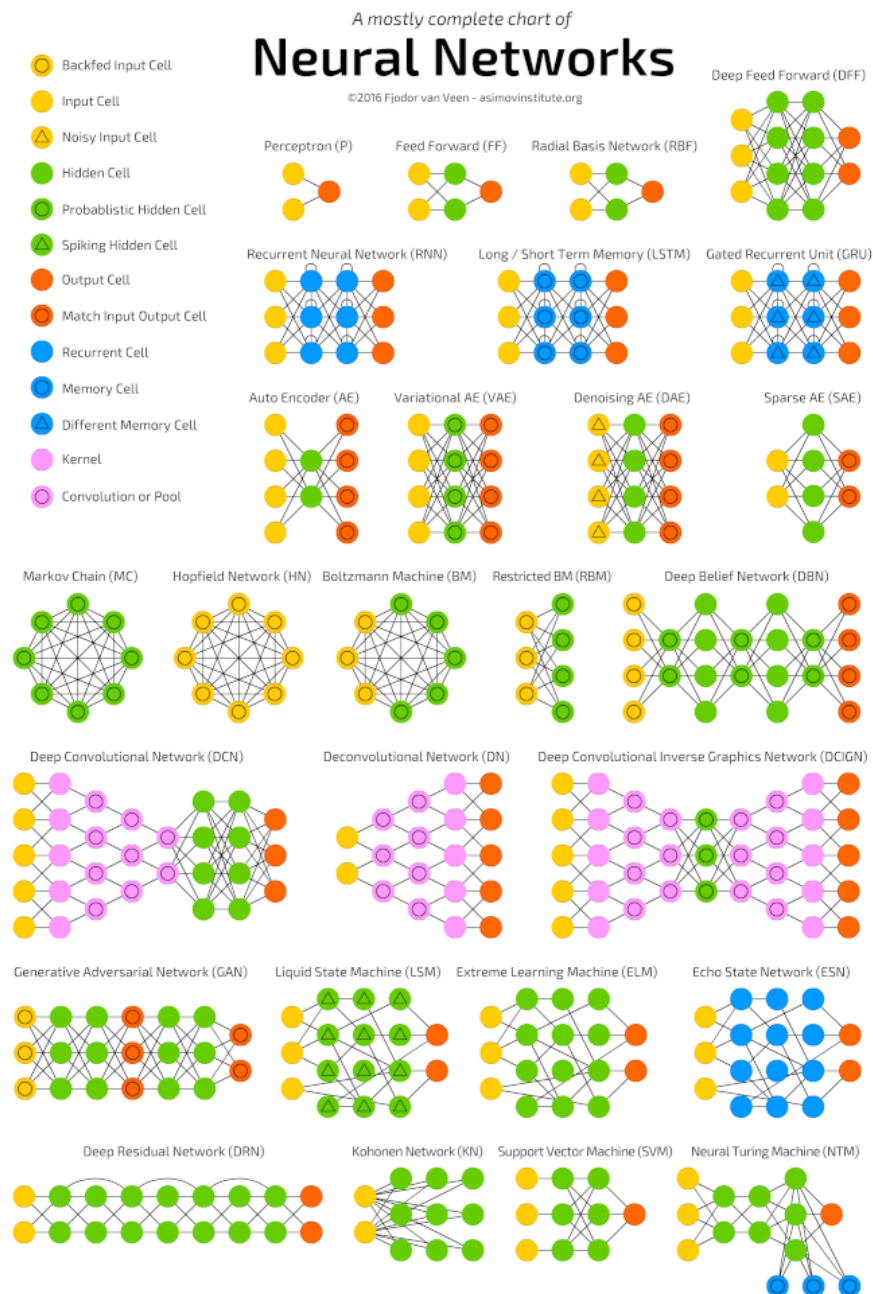


Рисунок 1.2 – схемы нейронных сетей глубокого обучения

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

(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.

1.4 Боты

С помощью ботов можно упростить себе 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.

1.4.1 Telegram bot

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].

1.4.2 WhatsApp bot

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

1.4.3 Viber bot

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.

1.5 Вывод

В данном разделе были изучены принципы работы нейронной сети и 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.

2. Конструкторский раздел

В данном разделе рассматривается процесс проектирования структуры программного обеспечения.

2.1 Функциональная модель

На рисунке 2.1 изображена функциональная модель IDEF0 нулевого уровня.

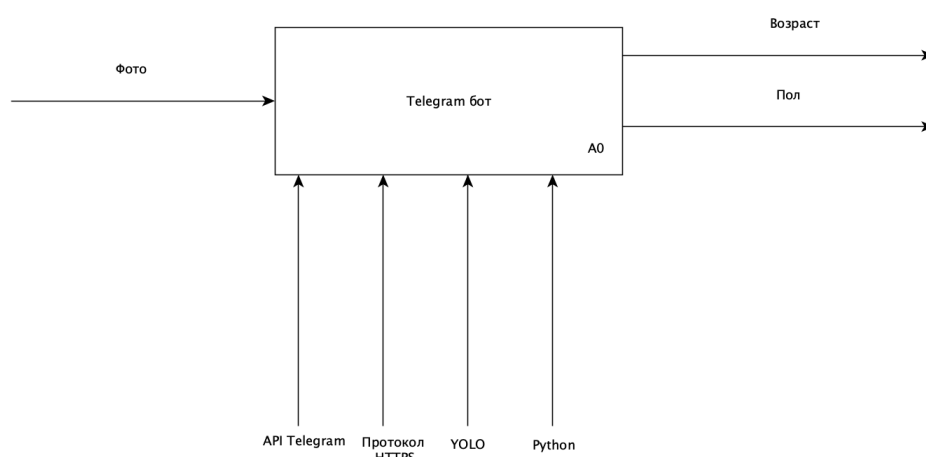


Рисунок 2.1 - IDEF0 нулевого уровня

2.2 Сценарий использования

На рисунке 2.2 представлена диаграмма взаимодействия пользователя с приложением.



Рисунок 2.2 – Use-case диаграмма

2.3 Нейросеть 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. В этом примере желтая ячейка сетки делает два прогноза для граничного прямоугольника (синие прямоугольники), чтобы определить местонахождение человека (рисунок 2.3) [4].

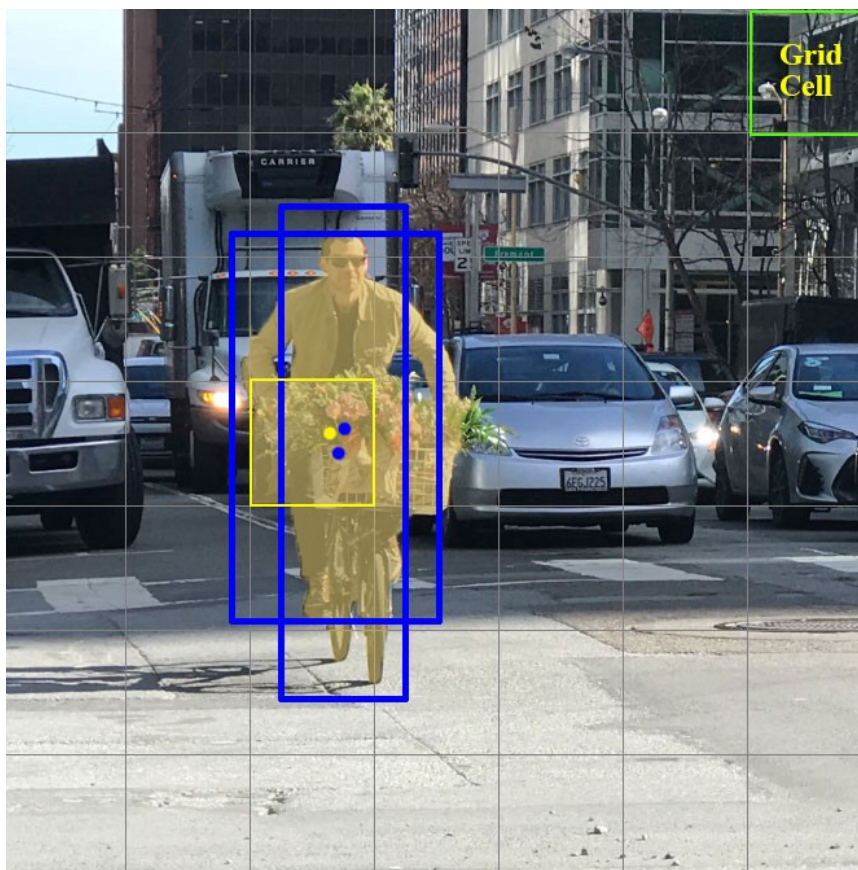


Рисунок 2.3 – ячейки YOLO

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 7×7 ($S \times S$), 2 граничных прямоугольника (B) и 20 классов (C).

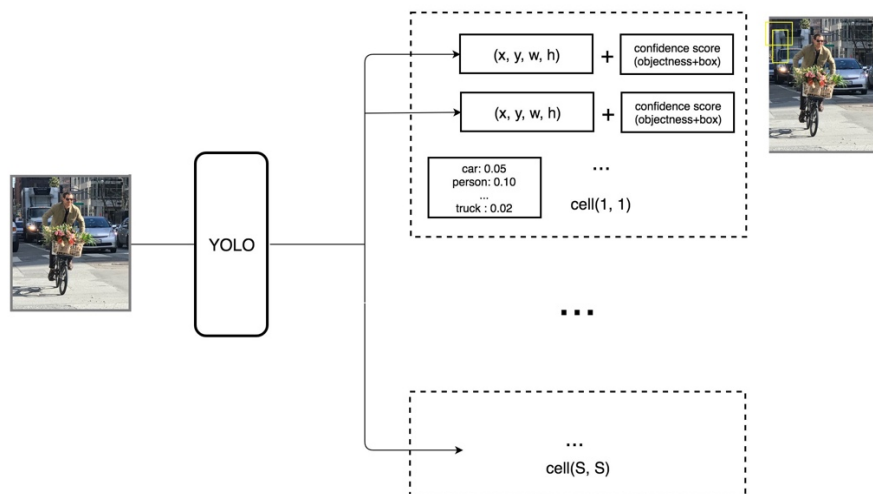


Рисунок 2.4 - YOLO делает прогнозы $S \times S$ с граничными блоками B

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).

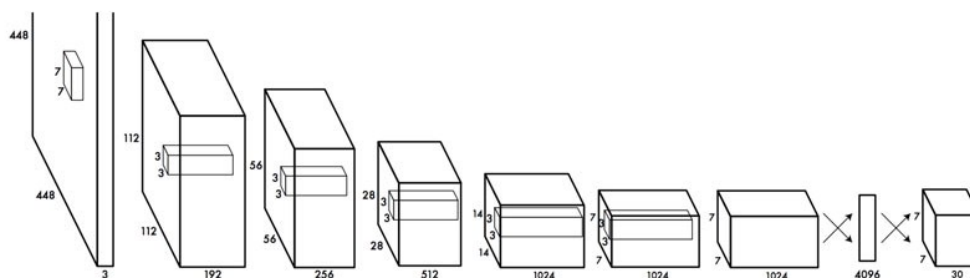


Рисунок 2.5 – пример прямоугольников

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:

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

2.4 Вывод

В данном разделе были приведены функциональная модель, сценарий использования приложений, рассмотрена и изучена свёрточная нейронная сеть Yolo.

3. Технологический раздел

В данном разделе выбирается язык программирования, на котором будет реализована поставленная задача, производится выбор среды разработки и рассматривается интерфейс пользователя.

3.1 Выбор языка программирования

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.

3.2 Выбор среды разработки

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

3.3 Описание некоторых моментов реализации

На листинге 3.1 представлена реализация функции с приветственным сообщением и переходом на другую функцию.

Листинг 3.1 – функция с приветственным сообщением

```
@bot.message_handler(commands=['start'])
def send_welcome(message):
    bot.reply_to(message, "Привет! Как тебя зовут?")
    bot.register_next_step_handler(message, get_name)
```

На листинге 3.2 представлена реализация загрузки фотографий из клиента:

Листинг 3.2 – реализация загрузки фотографий из клиента

```
@bot.message_handler(content_types=['photo'])
def photo(message):
    print ('message.photo =', message.photo)
    fileID = message.photo[-1].file_id
    print ('fileID =', fileID)
    file_info = bot.get_file(fileID)
```

```
print ('file.file_path =', file_info.file_path)
downloaded_file = bot.download_file(file_info.file_path)

with open("image.jpg", 'wb') as new_file:
    new_file.write(downloaded_file)
```

3.4 Интерфейс пользователя

На рисунках 3.3 – 3.7 показан интерфейс и пример работы с ботом.

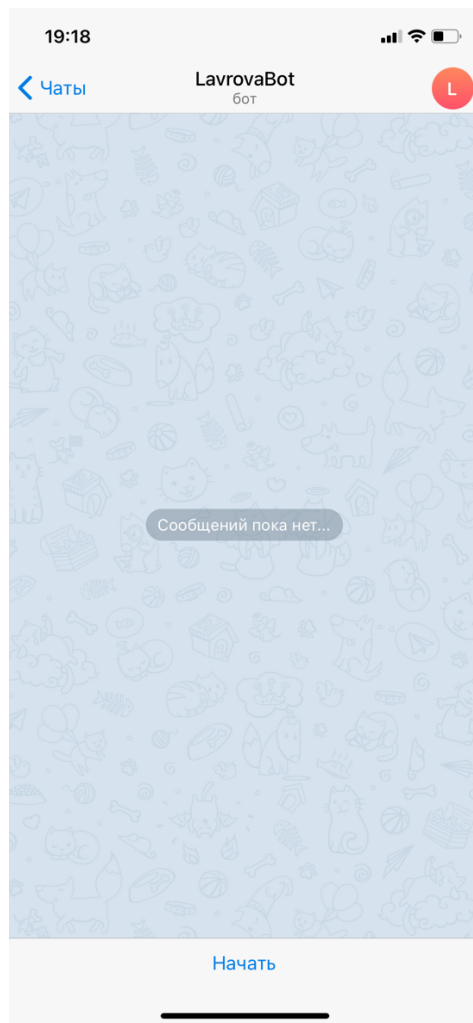


Рисунок 3.3 – пример работы с ботом

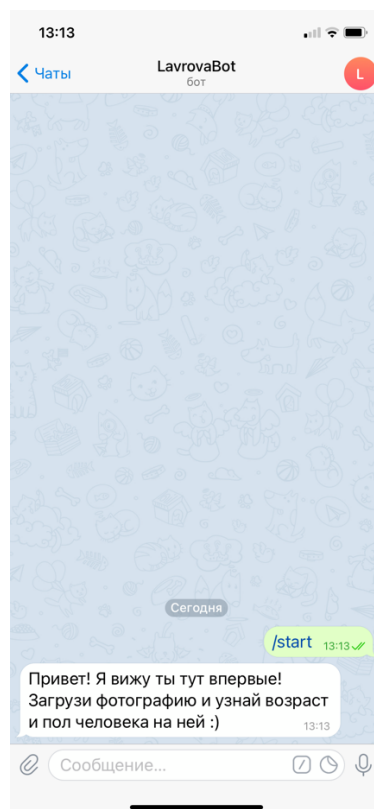


Рисунок 3.4 – пример работы с ботом

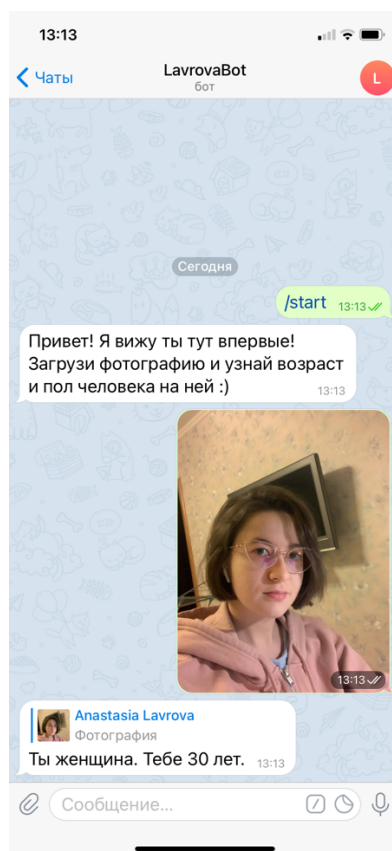


Рисунок 3.5 – пример работы с ботом

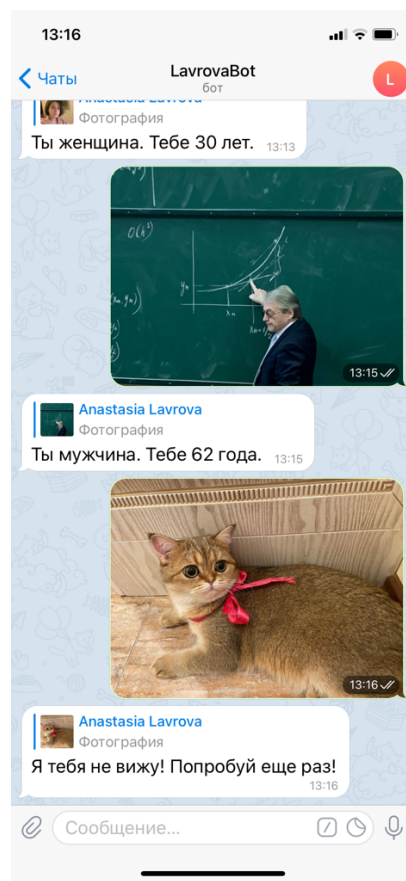


Рисунок 3.6 – пример работы с ботом

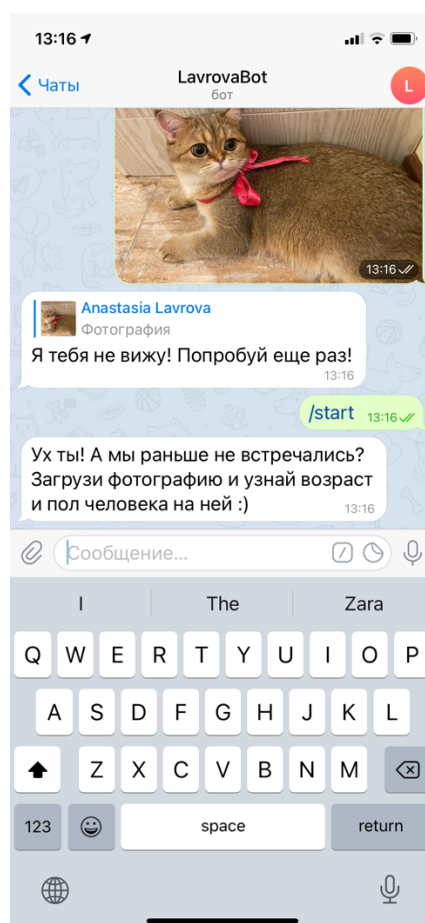


Рисунок 3.7 – пример работы с ботом

3.5 Вывод

В данном разделе был выбран язык Python в качестве языка программирования, на котором была реализована поставленная задача, был выбран Pycharm в качестве среды разработки, а также был рассмотрен интерфейс для взаимодействия с пользователем.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной курсовой работы были выполнены следующие задачи:

- была формализована задача;
- рассмотрены существующие API социальных сетей;
- изучена работа социальных сетей;
- разработано программное обеспечение, позволяющее осуществить определение возраста и пола по фотографии с использованием нейронной сети и бота социальной сети.

В качестве улучшения проекта предполагается дальнейшее обучение модели для получения более точных результатов.

Список используемых 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: 05.12.2020).
2. Telegram API Bot. [Электронный ресурс] – Режим доступа: <https://core.telegram.org/bots/api> (дата обращения: 10.12.2020).
3. MTProto Mobile Protocol. [Электронный ресурс] – Режим доступа: <https://core.telegram.org/mtproto> (дата обращения: 18.12.2020).
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: 15.12.2020).
5. Нейронные сети для начинающих. Часть 1. [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/312450/> (дата обращения: 18.12.2020).

