# 1 Исследовательский раздел

## 1.1 Понятие эмоции и мимики

Прежде чем начать деятельность в какой-либо предметной области, нужно провести анализ этой предметной области. Поэтому первым делом нужно понять, что такое эмоции и мимика человека.

Эмоция – это особый вид психических процессов, которые выражают переживание человеком его отношения к окружающему миру и самому себе. Опираясь на одну из теорий, за авторством советского физиолога П. К. Анохин, в которой установлено, что способность испытывать эмоции была выработана в результате эволюции живых существ как способ более успешной адаптации к условиям существования в различных условиях.

Эмоции имеют важную роль в человеческой жизни, а в частности социальном взаимодействии между людьми. Они могут выражены разными способами. Примерами могут являться: позы, мимика, движения, голос, дыхание или сердцебиение. Но самой наибольшей выразительностью обладает лицо человека, его мимика. Мимика человека – это способность лица выражать чувства и настроения с помощью движения мышц лица.

Каждый человек может по-разному выражать эмоции, но 70-х годах прошлого века, американским психологом Полом Экманом, было установлено, что определенный ряд эмоций являются универсальными и могут быть поняты разными людьми с абсолютно разными культурами. Этими эмоциями являются: гнев, страх, печаль, отвращение, презрение, удивление и радость. Именно по этим эмоциям реально определить психоэмоциональное состояние человека.

## 1.2 Теория распознавания образов

Следует понимать, что распознавание психоэмоционального состояния человека является подзадачей глобальной задачи распознавания образов. Теория распознавания образа — раздел информатики и смежных дисциплин, развивающий основы и методы классификации и идентификации предметов, явлений, процессов, сигналов, ситуаций и т. п. объектов, которые характеризуются конечным набором некоторых свойств и признаков. Образом называется классификационная группировка в системе классификации, объединяющая определенную группу объектов по некоторым определенным признаком. Образы обладают характерными объективными свойствами в том смысле, что разные люди, обучающиеся на различном материале наблюдений, большей частью одинаково и независимо друг от друга классифицируют одни и те же объекты.

В современном мире, необходимость в распознавании образов может возникнуть в различных предметных областях. Поэтому работа над этой очень важна, так как ее решение может существенно облегчить труд человека, а также повысить надежность и точность различных рабочих процессов за счет отсутствия возможности допущения ошибки из-за человеческого фактора.

## 1.3 Обнаружение лица человека

Задачу, которую предстоит решить в рамках данной работы можно разделить на 2 основные: детектирование лица человека на фотографии и определение (классификация) эмоции человека по его мимике. Ниже я рассмотрю некоторые методы детектирования человеческих лиц.

### 1.3.1 Метод гибкого сравнения на графах

В данном методе лицо человека представляется в виде графа со взвешенными вершинам и ребрами. Суть метода заключается в сопоставлении этих графов. Один из графов является эталонным. Этот граф остается неизменяемым, но в то же время у другого такого графа, каждая вершина смещается на некоторое расстояние в какое-либо направление относительно начального положения вершины. Далее выбирается такая новая позиция вершины, что разница между значениями признаков у графа со смещенными вершинами и эталонным графом была минимальной. Соответственно, граф, вершины которого смещаются называется деформированным, а сам процесс смещения называется деформацией. Процесс деформации будет продолжаться до тех пор, пока мы не найдем минимальное различие между признаками эталонного и деформируемого графов.

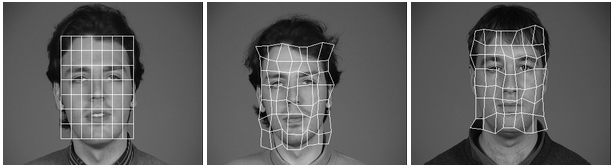


Рисунок 1.1 – Пример эталонного и деформированных лиц

Этот процесс будет применен ко всем лицам, находящимся в базе данных системы. Отсюда можно выделить зависимость между количеством данных в базе, временем выполнения и точностью обнаружения. Чем больше данных, тем дольше обработка, но выше точность. Справедливо и обратное утверждение: чем меньше данных, тем меньше обработка, но ниже точность.

### 1.3.2 Метод Виолы-Джонса

Метод Виолы-Джонса – это алгоритм обнаружения объектов на изображениях в реальном времени, разработанный в 2001 году Полом Виолой и Майклом Джонсом. Несмотря на то, что главной задачей при разработке этого метода было нахождение лиц, этот метод используется и для распознавания других различных предметов. Существует множество реализаций, в том числе в составе библиотеки компьютерного зрения OpenCV.

Данный метод основан на каскадах Хаара, которые представляют собой разбивку некой прямоугольной области на наборы разнотипных прямоугольных подобластей.

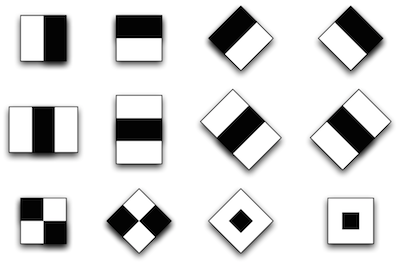


Рисунок 1.2 – Пример каскадов Хаара

Признаки, применяемые алгоритмом, используют суммирование пикселей из прямоугольных регионов. Величина каждого признака вычисляется как сумма пикселей в белых прямоугольниках, из которой вычитается сумма пикселей в черных областях. При стандартном размере признака в 24x24 пикселя, возможно 162 тысячи разных признака и производить расчёт каждого слишком долго. Поэтому в алгоритме Виолы-Джонса используется вариация алгоритма обучения AdaBoost, как для выбора признаков, так и для настройки классификаторов, что позволяет довольно быстро найти лицо. У данного алгоритма низкая вероятность неправильно определить лица или ложно сработать. Также он позволяет обнаружить лицо, которое может быть отклонено под углом в 30º, и при этом точность определения лица может быть свыше 90%.

### 1.3.3 Сравнение представленных методов

Стоит отметить, что мимика никак практически никак не будет влиять на обнаружения лиц в представленных алгоритмах. Чтобы понять какой из алгоритмов лучше использовать следует построить сравнительную таблицу методов.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Метод | Точность | Влияние мимики | Время обнаружения | Вычислительная сложность | Угол обзора |
| Гибкое сравнение на графах | ~ 90% | Низкое | Высокое  (Зависит от данных) | Высокая | 30º |
| Метод Виолы - Джонса | > 90% | Низкое | Низкая | Низкая | 30º |

Таблица 1.1 – Сравнение методов обнаружения лиц

Из-за высокой точности, быстрой скорости работы, а также из-за простой программной реализации, в частности представление алгоритма в библиотеки OpenCV, мной был выбран именно этот алгоритм обнаружения лиц.

## 1.4 Распознавание эмоции человека

Наиболее простой метод определения эмоции по изображению лица основывается на классификации ключевых лицевых точек. На изображении размечаются от 10 до 70 точек, которые привязываются к расположению носа, глаз, бровей, губ. Такая привязка дает возможность частично захватить мимику лица.

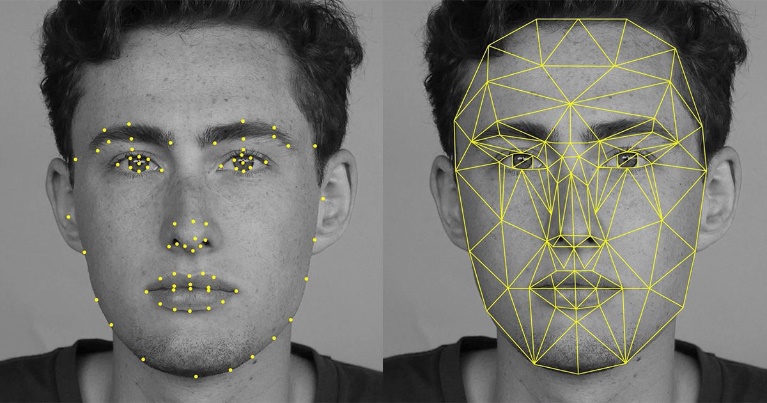


Рисунок 1.3 – Выделение ключевых точек лица

Данные ключевые точки и их координаты можно получить, используя сверточные нейронные сети (англ. convolutional neural network, сокр. CNN).

### 1.4.1 Нейронные сети

Прежде чем перейти к сверточным нейронным сетям, следует определить, что такое нейронная сеть.

Искусственной нейронной сетью называют математическую модель, состоящую из определенного количества простых взаимосвязанных элементов, которые в свою очередь обрабатывают информацию с помощью изменения своих внутренних параметров под влиянием поступающих внешних данных.

Структура искусственной нейронной сети взята из биологии, так как по строению она очень сильно схожа с нейронной сетью мозга человека.

Нейронные сети можно использовать в решении самых различных задач, но свое наибольшее распространение они получили среди задач распознавания образов и прогнозирования различных событий. С помощью нейронных сетей могут анализироваться различные закономерности в больших объемах данных для решения статистических и аналитических задач.

### 1.4.2 Простые нейронные сети

Самым простым видом нейронной сети является персептрон. Основным элементом персептрона является нейрон. Нейрон – это элемент, который может хранить в себе некоторый вес, умеющий принимать значение на вход, выполнять операцию умножения и выдавать результат на выход.

В основном различают три вида нейронов:

* Входные нейроны – нейроны, которые принимают входной сигнал, входную информацию.
* Промежуточные нейроны – нейроны, выполняющие вычислительные операции.
* Выходные нейроны – нейроны, которые выполняют роль выхода из нейронной сети.

Соответственно, персептрон обычно состоит из нескольких слоев, где каждый слой представляет из себя набор нейронов, которые не имеют каких-либо связей между собой. Первый слой состоит из входных нейронов, которые получают входные данные в виде вектора значений. Данные с входного слоя умножаются на вес нейрона, и результаты этого умножения передаются на следующий слой, обычно в каждый из элементов следующего слоя. Таким образом это будет продолжаться до последнего слоя, который по входным результатам сформирует результирующий вектор значений.

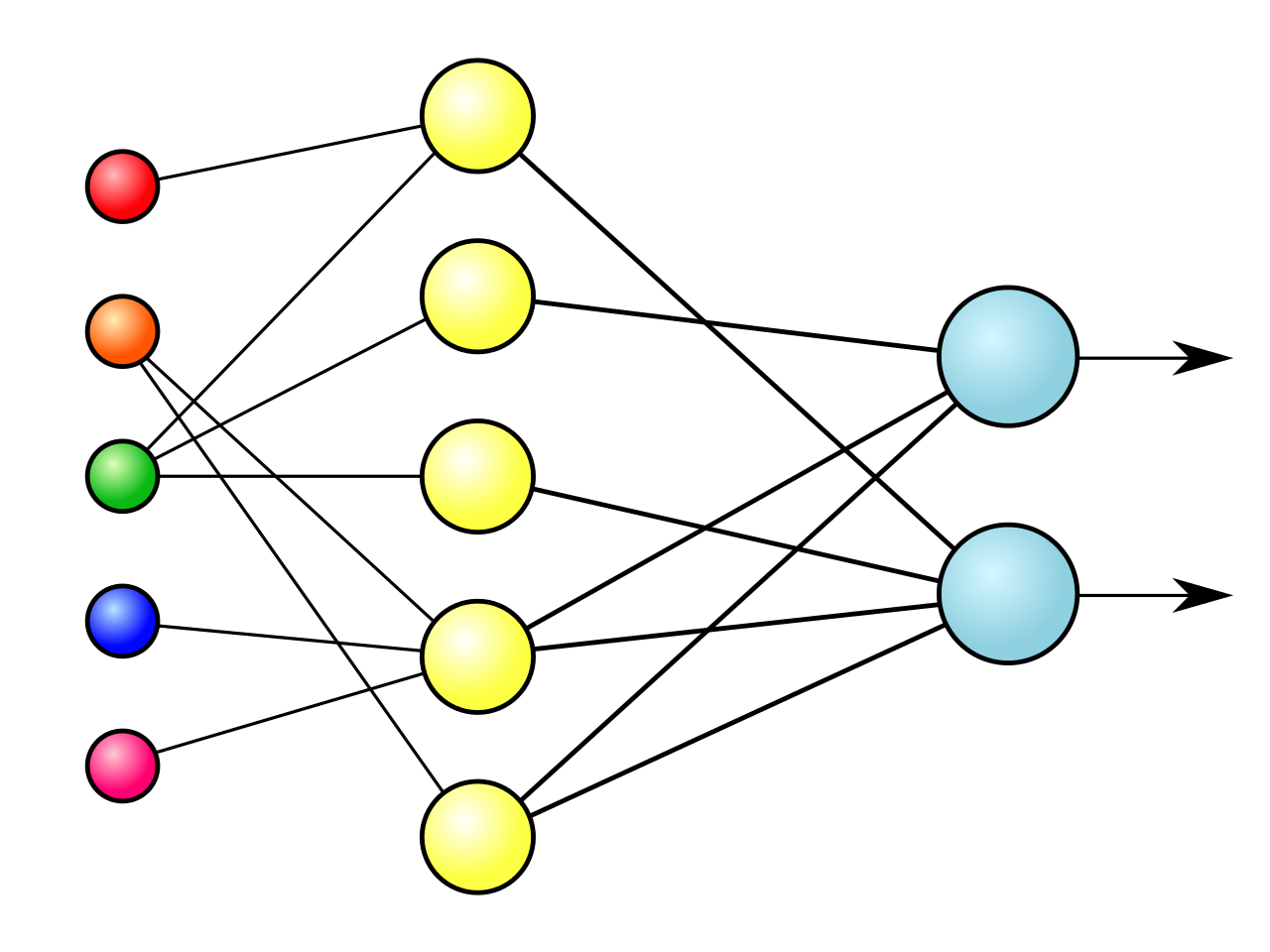


Рисунок 1.4 – Пример обычного персептрона

### 1.4.3 Сверточные нейронные сети

Сверточная нейронная сеть – это специальная архитектура искусственных нейронных сетей, которую предложил французский ученый в области информатики Ян Лекун в 1988 году. Эта архитектура была создана с целью эффективного распознавания образов и сейчас очень сильно распространена среди технологий глубинного обучения.

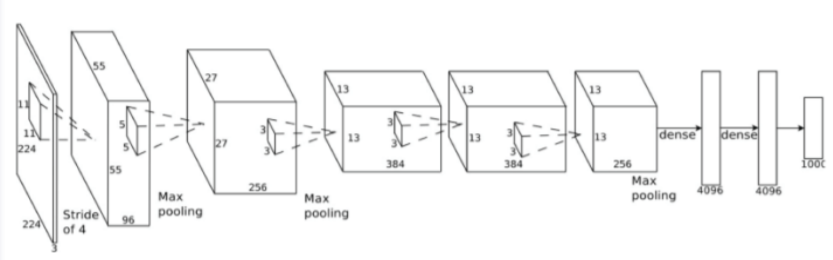


Рисунок 1.5 – Архитектура CNN

Главной особенностью сверточных нейронных сетей является наличие сверточного (англ. convolutional) и субдискретизирующих слоев (англ. pooling). Во время построения модели, эти 2 слоя чередуются между собой. Сверточный слой взаимодействуют с входным изображением путем навешивания на это изображение набора фильтров (ядер). Этот процесс называется сверткой. После сверточного слоя идет субдискретизирующий (англ. pooling) слой, задача которого уменьшить размерность получившегося выхода сверточного слоя. Выход со второго слоя дает в результате карты признаков. Карта признаков должна показать насколько входное изображение похоже на то или иное ядро свертки.

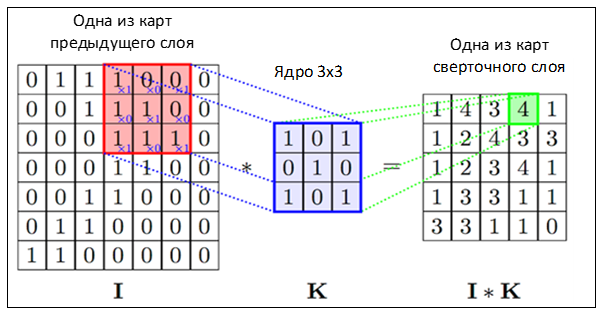


Рисунок 1.6 – Пример свертки

После нескольких сверточных блоков обычно идет полносвязный слой, роль которого представляют из себя многослойные персептроны. На практике это последний выходной слой, суть которого вывести результирующий вектор.

### 1.4.4 Преимущества и недостатки CNN

Сверточная нейронная сеть имеет следующий ряд преимуществ:

* На практике доказано, что это один из лучших алгоритмов по классификации и распознаванию изображений.
* По сравнению с многослойным персептроном, в сверточных нейронных сетях намного меньше весов, которые нужно настраивать. Это связано с единым ядром весов, которое используется для всего изображения. Поэтому такая нейронная сеть будет обобщать полученную информацию, а не просто запоминать пиксели и менять для каждого весовые коэффициенты, как это делает полносвязная нейронная сеть.
* Вычисления можно распараллелить, а значит, что можно ускорить обучение, используя графические процессоры (GPU).
* Есть устойчивость к сдвигу или повороту изображения. А используя методы генерации повернутых или сдвинутых изображений можно значительно повысить устойчивость.
* При обучении используется метод обратного распространения ошибки.

Недостатком является большое количество параметров сети, которые можно по-разному варьировать. Это может быть размерность ядра свертки, шаг сдвига ядра, количество слоев и т. д. Каждый из параметров может существенно повлиять на результат, поэтому настройка этих параметров производится экспериментальным путем. Поэтому для каждой новой задачи нужно проектировать новую модель.

Исходя из выше перечисленных преимуществ, для распознавания эмоций в рамках данной работы я буду использовать сверточные нейронные сети.

## 1.5 Выбор программных инструментов и языка программирования

### 1.5.1 Язык программирования Python

Языком программирования для реализации системы распознавания психоэмоционального состояния человека я выбрал Python. Это высокоуровневый язык программирования, ориентированный на повышение производительности разработчика и читаемости кода.

Python является одним из самых популярных языков в мире. В первую очередь за счет своей простоты и универсальности. Под универсальностью подразумевается возможность разрабатывать различные программные средства за счет различных библиотек дополнительных библиотек, которые создает сообщество по всему миру. Особую популярность язык получил в трех областях: разработка автоматических тестов, веб-разработка и машинное обучение.

Так как язык очень хорошо применим к машинному обучению, то мой выбор пал именно на этот язык.

### 1.5.2 Выбор фреймворка машинного обучения

В Python есть два крупных фреймворка для машинного обучения: TensorFlow и PyTorch.

TensorFlow – это фреймворк для глубинного обучения, который разработан компанией Google. Сильными ее сторонами являются визуализация и огромное количество опций для разработки высокоуровневых моделей машинного обучения.

PyTorch — это фреймворк созданный под влиянием фреймворка Torch, написанного на языке программирования Lua. Разработан данный фреймворк компанией Facebook. Он хорош для исследований, так как хорошо читается и отлаживается, но у него нет хорошо написанной документации и некоторых опций, которые есть в TensorFlow. Этот фреймворк довольно-таки новый, поэтому постоянно дорабатывается, но пока не дотягивает для уровня разработки компании Google.

Из-за хорошей документации, удобных инструментов визуализации и простого описания слоев нейронных сетей мой выбор пал на TensorFlow.

### 1.5.3 Библиотека OpenCV

Библиотека OpenCV – это библиотека компьютерного зрения и машинного обучения с открытым исходным кодом. В неё входят более 2500 алгоритмов, в которых есть как классические, так и современные алгоритмы для компьютерного зрения и машинного обучения. Использование этой библиотеки обусловлено наличием алгоритм Виолы-Джонса, который будет использован для обнаружения лиц.

# 1.6 Постановка задачи

Главной задачей этой работы является разработка системы распознавания психоэмоционального состояния человека, с последующей интеграцией данной системы в бот мессенджера Telegram. Выбор такого способа интеграции системы был обусловлен широким распространением различных систем мгновенного обмена сообщениями, что в свою очередь делает пользование разработкой максимально удобной для конечного пользователя, а выбор Telegram обусловлен хорошим API и возможностью воплощать в жизнь различные программные идеи в ботах.

Краткая блок-схема работы алгоритма системы представлена ниже:

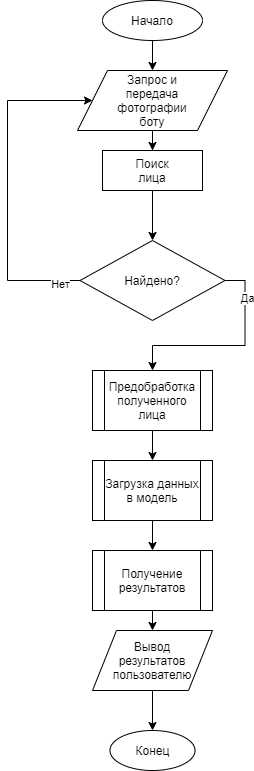


Схема 1.1 – Блок-схема алгоритма

Более подробная блок-схема алгоритма будет представлена в приложении, а сам алгоритм и его работа будут описаны в второй главе работы.

# 2 Специальный раздел

## 2.1 Требования к программной разработке

**Требования к функциональным характеристикам**

Программа должна выполнять следующие функции:

1. Чтение и сохранение в файловой системе полученной исходной фотографии.
2. Обнаружение лиц на фотографии.
3. Получение и сохранение обнаруженных лиц в файловой системе.
4. Предобработка фотографии лица.
5. Получение результата классификации эмоции лица, переданной системе.

**Требования к данным для обучения сверточной нейронной сети**

1. Данные находятся в формате CSV.
2. CSV – файл должен содержать 3 поля: pixels (лицо в пикселях) и emotion (эмоция соответствующего лица), usage.

**Требования к входным файлам**

1. Цветные или черно-белые изображения.
2. Формат изображений JPG, JPEG, PNG.
3. Размер и формат изображения не имеют значения

**Требования к выходным данным**

1. Лицо, которое было получено системой в ходе обработки
2. Гистограмма распределения вероятностей классов
3. Эмоция, которую распознает система

**Дополнительные требования**

Так как разработанная система будет интегрироваться в бот мессенджера Telegram, пользователю требуется иметь устройство, которое поддерживает данный мессенджер. Так как Telegram является кроссплатформенной разработкой, то достаточно иметь устройство под управлением следующих операционных систем: Windows Phone, iOS, Android, Microsoft Windows, Chrome OS, Mac OS, Linux или macOS.

## 2.2 Проектирование программной разработки

Современные системы распознавания образов оперируют методами классификациями и идентификации предметов, явлений, процессов и других объектов которые характеризуются набором некоторых свойств и признаков. Все такие системы строятся на основе использования машинного обучения и нейронных сетей. Поэтому в первую очередь, нужно найти такой набор данных, на основе которого модель нейронной сети сможет обучиться, и в последствии выдать высокий процент на точности на тестовых случаях. В рамках данной работы мне потребовался датасет с лицами людей, выражающих какие-либо эмоции, а также классы этих эмоций.

Следующем этапом разработки является предобработка этих данных. Обычно датасетом является файл с расширением \*.csv (Comma-Separated Values). Это текстовый формат для представления табличных данных. Каждая строка такого файла является одной строкой таблицы. Данные в строке разбиты на блоки с помощью запятых. Следовательно, такие файлы следует правильно обработать, синтаксически проанализировать (распарсить) для дальнейшей работы. От того в каком формате данные будут лежать в наборе, зависит то, как и к какому формату нужно привести входные данные. Причина необходимости в этом кроется в модели нейронной сети, так как она обучается на формате данных, приведенных в датасете.

Третьим этапом становится разработка функций и процедур, которые смогут привести входные данные к нужному формату. В моем случае входными данными являются обычные фотографии людей. В этих фотографиях, с помощью алгоритмов компьютерного зрения, я должен определить есть ли на этих фотографиях лица. Эти лица позже пойдут на вход нейронной сети, которая классифицирует эмоцию на фотографии.

Четвертый этап – это создание модели сверточной нейронной сети (CNN). На данном этапе нужно составить комбинацию сверточных, пулиноговых и полносвязных слоев для классификации эмоции. Передать такие параметры и выбрать такую функцию активации, чтобы результат точности на тестовых данных был как можно выше. На этом же этапе мы обучаем и сохраняем модель CNN для дальнейшего использования в системе распознавания психоэмоционального состояния человека.

Пятым этапом проектирования является, создание бота в мессенджере Telegram. Диалог с конечным пользователем будет проходить непосредственно с помощью бота. Предполагается, что пользователь будет отправлять фотографию с лицом на машину, на которой работает система. В результате, пользователь получает на выходе вероятную эмоцию, представленную на фотографии с лицом.

Последним этапом будет являться тестирование разработки и интеграция модели CNN в Telegram-бота. Это тестирование должно установить работоспособность нейронной сети и бота в целом.

Исходя из всего выше перечисленного список поставленных задач выглядит следующим образом:

1. Поиск набора данных для обучения нейронной сети.
2. Обработка и анализ данных набора.
3. Создание модели сверточной нейронной сети.
4. Создание процедур и функций для преобразования входных данных.
5. Написание бота для Telegram.
6. Тестирование и подключение модели к боту.

## 2.3 Программная реализация

### 2.3.1 Поиск и обработка набора данных

Как уже было написано в части проектирования программной разработки, первое с чего следует начать – это поиск набора данных на которых будет обучаться нейронная сеть. Конечно же можно сделать этот набор данных самостоятельно, но это будет очень времязатратно. Поэтому чтобы не тратить время составления такого набора, есть множество ресурсов, на которых можно найти нужный датасет. Одним из таких ресурсов является платформа для соревнований по машинному обучению Kaggle. Соревнование, из которого я взял набор данных называется «Challenges in Representation Learning: Facial Expression Recognition Challenge». А сам файл с данными представлен в виде файла с форматом \*.csv, и называется icml\_face\_data.csv.

Данные в этом датасете представлены согласно описанным требованиям – это поля таблицы, разделенные запятыми. Имеется 3 столбца с данными: emotion, usage и pixels. Столбец pixel – столбец с фотографиями. Здесь они представлены в виде одномерного массива, из которого будет формироваться черно-белая фотография с лицом размером 48х48 пикселей. Столбец emotion – это столбец с целевыми переменными. Именно в нем указано какая эмоция изображена на фотографии. Всего 7 классов эмоций: грусть, удивление, отвращение, страх, радость, нейтральность, гнев. Распределение эмоций этого набора данных представлена в таблице 2.1.

|  |  |
| --- | --- |
| Злость | 4953 |
| Отвращение | 547 |
| Страх | 5121 |
| Радость | 8989 |
| Грусть | 6077 |
| Удивление | 4002 |
| Нейтрально | 6198 |

Таблица 2.1 – Распределение эмоций на наборе данных

Последний столбец – это столбец который указывает на каких фотографиях должна обучаться нейронная сеть, а на каких она должна проходить тесты в рамках состязания на Kaggle. Его в рамках работы использовать не буду. На рисунке 2.1 показано как выглядят первые 5 строк необработанного набора данных.

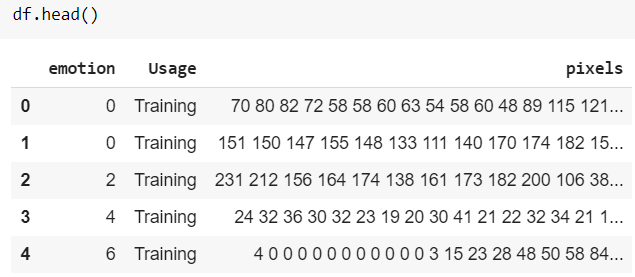


Рисунок 2.1 – Пример данных из набора

### 2.3.2 Обработка данных датасета

Далее происходит переход к следующей поставленной задаче. Это обработка полученных данных. Первое, что следует сделать, это посмотреть на данные и убедиться в том, что это действительно лица людей. Код, представленный на рисунке 2.2, преобразует одномерный массив пикселей в двумерный, из которого я формирую двухканальные изображения черно-белого цвета.



Рисунок 2.2 – Преобразование данных в изображения лиц

Преобразование данных в изображения было сделано в целях проверки данных. Нейронная сеть должна будет получить на вход массив пикселей размером 48х48, поэтому мы просто будем использовать те самые двумерные массивы из которых получили изображения.

После получения массивов, данные разбиваются на 2 части: на обучающую и на валидационную выборки данных. Общее количество изображений на выборках представлено в таблице 2.2.

|  |  |  |
| --- | --- | --- |
| Face\_data | Обучающая выборка | Валидационная выборка |
| Количество изображений | 28710 | 3590 |

Таблица 2.2 – Распределение изображений по выборкам

На обучающей выборке сверточная нейронная сеть будет обучаться, а валидационный набор данных будет предназначена для оценки работы модели. Оценка будет осуществляться путем перекрестной проверки (cross-validation). С помощью такой проверки, появляется возможность проанализировать поведение модели на неизвестных и независимых данных.

Далее некоторые изображения из обучающей выборки будут переработаны с помощью функции ImageDataGenerator (). Суть этой функции заключается в небольшом изменении исходных изображений. Изменениями могут быть наклоны, перевороты изображения или обычное обычный поворот изображения зеркально.



а) б)

Рисунок 2.3 – Пример изменения изображения

(a) – исходное изображение, (б) – результат работы функции

Данное преобразование нужно для повышения устойчивости распознавания эмоции нейронной сетью. Благодаря нему, сеть будет более устойчива к изменению ракурсов или положения лица.

### 2.3.3 Обучение сверточной нейронной сети

Как уже было сказано ранее, архитектура сверточной нейронной сети наиболее удобная и более результативная для работы с различными изображениями. В исследовательском разделе уже упоминалось про главный недостаток таких сетей – это большое количество различных параметров, начиная с формата входных данных, заканчивая количеством сверточных и субдискретизирующих слоев. Такие параметры устанавливаются эмпирическим путем, путем исследования и экспериментов с моделью нейронной сети. Так же следует понимать, что модель нужно проектировать так, чтобы CNN не переобучалась и машинально не запоминала расположение пикселей у той или иной эмоции. Итоговая модель полученной нейронной сети выглядит следующим образом:

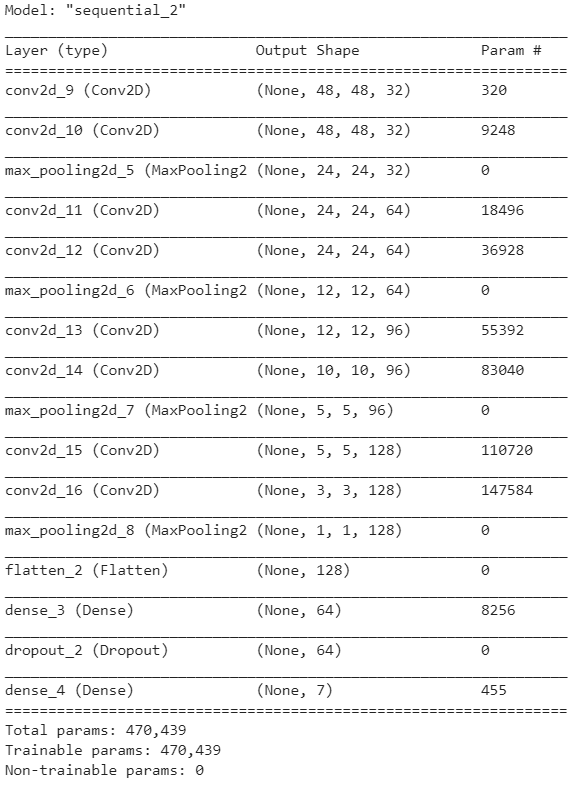


Рисунок 2.4 – Модель свертночной нейронной сети

Как видно из модели я использую по два сверточных слоя перед слоем субдискретизации. Свертка будет происходить с отступом по краям, чтобы ядро свертки прошло по всем пикселям. Общая формула свертки выглядит следующим образом:

,

где I – исходное изображение,

K – ядро свертки размером h\*w.

В конце каждого слоя свертки данные попадают в функцию активации ReLU (rectified linear unit). График функции ReLU представлен на рисунке 2.5 имеет следующую формулу:

Данная функция активации возвращает значение х, если х положительно, и 0 если х отрицательно.

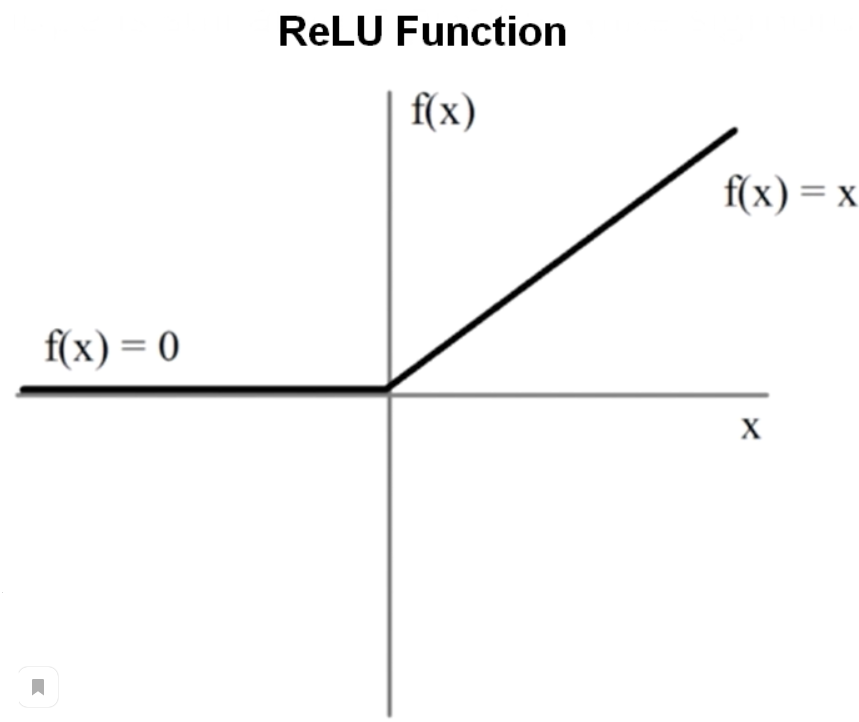


Рисунок 2.5 – График функции ReLU

Применение ReLU существенно повышает скорость сходимости стохастического градиентного спуска по сравнению с другими функциями активации. Это обусловлено линейным характером и отсутствием насыщения данной функции.

После последнего слоя субдискретизации результат поступает на слой выравнивания (flatten). Этот слой произведет преобразование двумерного массива в одномерный. После чего этот одномерный массив будет передан в полносвязный слой, для дальнейшей классификации.

Чтобы нейронная сеть не переобучалась, я пользуюсь двумя методами избежать этого переобучения. Первым методом является метод отсева (dropout). Суть этого метода заключается в том, что на каждой эпохе обучения некоторые нейроны из полносвязного слоя, будут отключаться с определённой вероятностью во время обучения. В результате более обученные нейроны получают в сети больший вес. Такой приём значительно увеличивает скорость обучения, качество обучения на тренировочных данных, а также повышает качество предсказаний модели на новых тестовых данных.

Второй метод – умышленное уменьшение скорости обучения нейронной сети. Когда значение точности не будет иметь сильного прироста, скорость обучение будет уменьшаться. Модель часто получает выигрыш от такого замедления и уменьшает шанс переобучиться.

Выходной функцией активации последнего полносвязного слоя будет функция softmax. Такая функция хороша, когда нужно классифицировать большое объектов. Формула этой функции выглядит следующим образом:

Последний слой нейронной сети на выходе выдает 7 вероятностей (матрица весовых коэффициентов), которые описывают возможную принадлежность к той или иной эмоции.

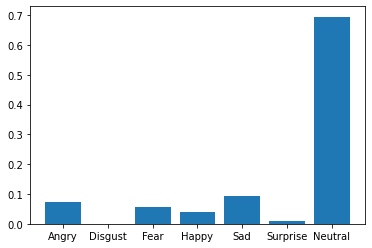


Рисунок 2.6 – Гистограмма распределения эмоций

Обучение проводилось на CPU и GPU. В ходе обучения был сделан вывод что обучение сверточной нейронной сети происходит примерно в 2000 раз быстрее. Обучение одной эпохи данной модели на CPU (Intel Core i7 (2.8 Ghz)) заняло 7 часов, в тоже время обучение одной эпохи на GPU (Nvidia GeForce GTX 1050 Ti) заняло 25 секунд. Поэтому появляется необходимость в наличии сервера с мощными GPU. Платформа Google Colab предоставляет ресурсы для работы с данными и машинным обучением, и дает возможность пользоваться их серверами для работы с моделями машинного обучения.

Полученная модель сохраняется либо для дальнейшего дообучения, либо для использования модели в прикладных приложениях. Как правило TensorFlow сохраняет модель с разрешением \*.h5.

### 2.3.4 Дополнительные функции

Далее рассмотрим основные функции, которые используются в рамках данной работы.

1. viewImage() – функция, которая отрисовывает изображение.
2. cropFace() – эта функция с помощью алгоритма Виолы-Джонса обнаруживает лицо на входном изображении. После чего происходит обрезание изображения в области лица, тем самым получая само лицо для дальнейшей работы системы.
3. save\_face() – выполняет сохранение полученного лица в файловую систему.
4. scale\_image() – функция преобразует изображения с лицом к формату данных нейронной сети. Производит сжатие фотографии к размеру 48х48 пикселей. Пример такого преобразования представлен на рисунке 2.7.

 https://sun9-40.userapi.com/c206820/v206820922/12e4c1/6xWhnYmqeNM.jpg

(а) (б)

Рисунок 2.7 – Пример работы функции сжатия изображения

(Изображение б) специально увеличено)

1. grays() – преобразует цветное изображение к черно-белому.

### 2.3.5 Использование API Telegram

Для работы с мессенджером будет использовано API Telegram, специально созданное для работы с ботами. В main - функции будет используется бесконечный цикл, который ждет получения входных данных. Как только данные будут переданы, цикл начнет прогонять полученные данные через функции преобразования. Внутри цикла так же подгружается полученная модель обучения. На основе результатов работы нейронной сети, строится диаграмма вероятностей и выделяется более вероятная эмоция. Эти результаты и найденное лицо, будут отправляться пользователю в ответ с помощью методов send\_photo() и send\_message().

### 2.3.6 Тестирование и отладка

На этапе интеграции модели и после обучения модели проводится тестирование точности модели. Первый тест проходит на валидационной выборки. Так как обучение происходит по эпохам, то после каждой эпохи мы проверяем точность CNN.

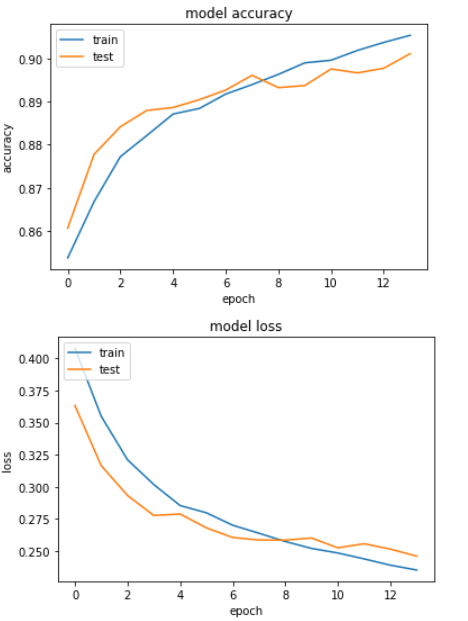


Рисунок 2.8 – Графики точности и потерь

Далее проверяется работоспособность бота. На вход подаются различные фотографии, и просматриваются различные ответы бота. При нахождении каких-либо ошибок, код немедленно отлаживался.

## 2.4 Демонстрация результата

Результат работы разработки показан на рисунке 2.9. На нем показан результат распознавания эмоции «радость».

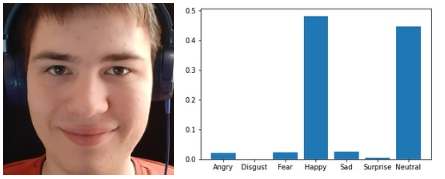


Рисунок 2.9 – Результат работы системы

# 3 Методический раздел

## 3.1 Расчет затрат на разработку

Величину затрат на разработку () программного продукта можно найти по формуле (3.1):

+++++, (3.1)

где – затраты на используемые материалы;

– заработная плата исполнителей;

– дополнительная заработная плата исполнителей:

– отчисления от заработной платы исполнителей;

– накладные расходы.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Затраты | Количество | Стоимость, руб. | Сумма,  руб. |
|  | Бумага, уп. | 2 | 140 | 280 |
|  | Письменные принадлежности, шт. | 2 | 100 | 200 |
|  | Картридж для принтера, шт. | 1 | 1660 | 1660 |
|  | Справочная литература, кн. | 3 | 500 | 1500 |
| Итого: | | | | 3 640 |

Таблица 3.1 – Расчёт материальных затрат

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Оборудование | Мощность, кВт. | Стоимость  1 кВт, руб. | Время работы, час. | Сумма, руб. |
| 1 | Компьютер | 0,4 | 3,38 | 320 | 433 |
| 2 | Принтер | 0,05 | 3,38 | 80 | 14 |
| Итого: | | | | | 547 |

Таблица 3.2 – Расчёт затрат на электроэнергию

### 3.1.1 Основная заработная плата исполнителей

К статье «Заработная плата» относится заработная плата научных, инженерно-технических и других работников, непосредственно участвующих в разработке ПО. Расчет ведется по формуле (3.2):

, (3.2)

где - заработная плата исполнителей (руб.);

– средняя дневная тарифная ставка работника организации (руб./чел.дни);

Т – трудоёмкость разработки программного средства (чел.дни).

определяется по формуле (3.3):

= С/, (3.3)

где:  
 С – месячная зарплата работника (руб./мес.);

– среднее количество рабочих дней в месяце (22дн.).

|  |  |  |  |
| --- | --- | --- | --- |
| Исполнители | Оклад, руб/мес | Продолжительность работы, календ. дн. | Сумма, руб |
| Руководитель проекта | 100000 | 14 | 50000 |
| Программист | 70000 | 14 | 35000 |
| Тестировщик | 40000 | 14 | 20000 |
| Итого | | | 105000 |

Таблица 3.3 – Расчёт фонда заработной платы (ФЗП)

Дополнительная заработная плата на период разработки ПС рассчитывается относительно основной и составляет 10% от её величины (3.4):

= (3.4)

= 0,1 \* 105000= 10500 (руб).

### 3.1.2 Расчет страховых взносов во внебюджетные фонды.

Отчисления на социальное страхование рассчитываются относительно выплаченной заработной платы и составляют 30% (3.5):

=(+)\*0.3. (3.5)

= (10500 + 105000) \* 0.3 = 34650

### 3.1.3 Накладные расходы

Величина накладных расходов принимается равной 60% от основной зарплаты исполнителей (3.6):

, (3.6)

где – накладные расходы (руб.);

– основная заработная плата исполнителей (руб.);

К – коэффициент учёта накладных расходов (К=0,6).

= 105000\*0.6 = 63000 руб

### 3.1.4 Общие затраты

Исходя из формулы, представленной ранее (3.1), сумма, затраченная на разработку, составит 206837 рублей.

## 3.2 Оценка экономической эффективности проекта

Годовой эффект от использования результатов разработки определяется по формуле:

Эг = P-Z (3.7)

где, P – результат, достигнутый в течении года, благодаря использованию проекта;  
 Z – полные затраты на разработку.  
 Результат, достигнутый благодаря использованию разработанной программы, определяется доход от спрогнозированного спроса.  
 Доход от спрогнозированного спроса оценивается условно:

P =300000 (руб).

Таким образом, годовой эффект (Эг) от использования разработки равен:

Эг = P-Z = 300000 – 206837 = 93163‬ (руб).

Разработанная программа позволяет заработать 93163‬ рублей за год, за счет прогнозирования спроса.