

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт Математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

01.03.02 Прикладная математика и информатика

(код и наименование направления подготовки, специальности)

Компьютерные технологии и математическое моделирование
(направленность (профиль)/специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Разработка алгоритма распознавания эмоций на изображении

Обучающийся

В.С. Базаева

(Инициалы Фамилия)

(личная подпись)

Руководитель

к.т.н., В.С. Климов

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

к.п.н., Т.С. Якушева

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Аннотация

Тема бакалаврской работы: «Разработка алгоритма распознавания эмоций на изображении».

В данной бакалаврской работе исследуется вопрос программной реализации технологии распознавания эмоций на основе анализа изображений с использованием нейронных сетей.

Объект исследования – алгоритмы распознавания эмоций на изображении.

Предмет исследования – нейросетевая реализации алгоритма распознавания эмоций на изображениях.

Цель исследования – подбор архитектуры и тестирование нейронной сети, обеспечивающей распознавание эмоций на изображении при наименьшем количестве слоев и без поиска ключевых точек лица.

Структура бакалаврской работы представлена введением, тремя разделами, заключением и списком литературы.

Во введении описывается актуальность проводимого исследования, формулируется цель и ставятся задачи, которые необходимо решить.

В первой главе рассматривается сущность задачи распознавания эмоций, а также приводятся наиболее известные существующие аналоги.

Во второй главе осуществляется разработка алгоритма для распознавания эмоций на изображении, описывается обучающая выборка и результаты тестирования алгоритма.

В третьем разделе рассматриваются особенности программной реализации предложенного алгоритма распознавания эмоций на изображении.

В заключении представляются выводы по проделанной работе.

В работе использовано 4 формулы, 33 рисунка, 2 таблицы, список литературы содержит 20 литературных источников. Общий объем выпускной квалификационной работы составляет 51 страницы.

Abstract

The topic of the bachelor's thesis is "Development of an algorithm for emotion recognition on an image".

This bachelor's thesis explores the issue of software implementation of the technology of emotion recognition based on image analysis using neural networks.

The object of the research is algorithms for emotion recognition on an image.

The subject of the research is a neural network implementation of the algorithm for the recognition of emotions in images.

The aim of the study is to select the architecture and test the convolutional neural network that provides emotion recognition in the image with the fewest number of layers and does not require the transformation of the feature space.

The structure of the bachelor's thesis is represented by an introduction, three sections, a conclusion and a list of references.

The introduction describes the relevance of the research, formulates the goal and sets the tasks to be solved.

The first chapter deals with the essence of the problem of emotion recognition, and the most well-known existing analogues are given.

The second chapter develops an algorithm for the recognition of emotions in the image, describes the training sample and the results of testing the algorithm.

The third section discusses the features of the software implementation of the proposed algorithm for recognizing emotions in an image.

The conclusion of the work is presented in the conclusion.

Оглавление

Введение.....	5
Глава 1 Анализ технологий распознавания эмоций на изображении	7
1.1 Введение в задачу распознавания эмоций на изображении	7
1.2 Сравнение алгоритмов распознавания эмоций.....	13
Глава 2 Проектирование алгоритма для анализа эмоций на изображении.....	21
2.1 Алгоритм нейросетевого распознавания эмоций	21
2.2 Источник данных для обучения нейронной сети	28
2.3 Оптимизация структуры нейронной сети и данных для обучения..	32
Глава 3 Программная реализация алгоритма для распознавания эмоций на изображении	36
3.1 Описание разработанного приложения	36
3.2 Особенности программной реализации и примеры работы приложения.....	37
Заключение	48
Список используемой литературы и используемых источников.....	50

Введение

Компьютерное зрение – это научное направление по созданию аппаратно-программных комплексов, способных производить обнаружение, локализацию и распознавание объектов. Технологии компьютерного зрения применяются при решении таких задач, как обеспечение безопасности на массовых мероприятиях, автоматизация охранных функций на различных объектах, верификация клиентов банков по изображению, улучшение качества фотографий [1].

В настоящее время исследователи из области искусственного интеллекта изучают возможности применения технологий компьютерного зрения для распознавания эмоций человека на основе изображения его лица.

Ежегодно проводятся соревнования разработчиков алгоритмов распознавания эмоций, цель в которых – добиться максимальной точности распознавания на заданном наборе данных. Анализ результатов соревнований AFEW-VA, EmotioNet challenge, OMG-Emotion challenge, 6th Emotion Recognition in the Wild Challenge (EmotiW) и EmoReact, AffectNet позволил выявить следующие особенности реализации алгоритмов. При реализации алгоритмов используется связка двух методов:

- метод поиска ключевых точек лица на изображении,
- метод классификации, определяющий класс эмоции на основе взаимного расположения ключевых точек лица.

Конечная точность работы алгоритма распознавания эмоций зависит от подбора удачного сочетания этих двух методов [5].

В данной работе предлагается отказаться от использования связки «метод поиска ключевых точек лица + метод классификации» и вместо этого использовать сверточную нейронную сеть. Сверточные нейронные сети предназначены для работы с изображениями и умеют самостоятельно определять необходимые признаки изображений для их классификации. Предполагается, что сверточная нейронная сеть с малым количеством слоев

сможет показать результаты по точности сопоставимые со связкой «метод поиска ключевых точек лица + метод классификации».

Таким образом, цель работы – подбор архитектуры и тестирование нейронной сети, обеспечивающей распознавание эмоций на изображении при наименьшем количестве слоев и без поиска ключевых точек лица.

Для достижения цели решаются следующие задачи:

- проанализировать предметную область и существующих подходов по распознаванию эмоций на изображениях;
- разработать нейросетевой алгоритм распознавания эмоций на изображении и выбрать структуру нейронной сети;
- разработать программное обеспечение и протестировать эффективность алгоритма.

Практическая значимость бакалаврской работы заключается в разработке программного обеспечения для нейросетевого распознавания эмоций.

Данная работа состоит из введения, трех глав, заключения, списка используемой литературы и приложений.

В первой главе рассматривается сущность задачи распознавания эмоций, а также анализируются существующие подходы для решения данной задачи.

Во второй главе осуществляется разработка алгоритма для распознавания эмоций на изображении, описывается обучающая выборка и результаты тестирования алгоритма.

В третьем разделе рассматриваются особенности программной реализации предложенного алгоритма распознавания эмоций на изображении.

В заключении представляются выводы по проделанной работе.

Бакалаврская работа состоит из 51 страницы текста, 33 рисунков, 2 таблиц, 4 формул и 20 источников.

Глава 1 Анализ технологий распознавания эмоций на изображении

1.1 Введение в задачу распознавания эмоций на изображении

Алгоритмы машинного обучения являются одной из ключевых технологий искусственного интеллекта, обеспечивающей ему способность учиться решать задачи, задаваемые пользователем. Изначально алгоритмы машинного обучения применялись для анализа данных, которые можно представить в виде простых таблиц с небольшим количеством столбцов и строк. Например, в виде таких таблиц можно представить данные пациентов медицинского учреждения или студентов, обучающихся в университете. В этом случае анализ таких данных с помощью алгоритмов машинного обучения позволяет автоматизировано искать закономерности в имеющихся параметрах и строить на их основе различные прогнозные модели. На данных пациентов больницы – можно построить модель, определяющую по симптомам и результатам медицинских анализов вероятность наличия определенных заболеваний, а для студентов учебного заведения, на основе предыдущих оценок по дисциплинам можно прогнозировать вероятность успешного завершения обучения в вузе [11].

С течением времени алгоритмы машинного обучения стали применяться также в анализе текстовой информации, что сформировало направление в искусственном интеллекте под названием «Обработка естественного языка». А применение машинного обучения в задачах анализа данных видео потока и отдельных кадров привело к возникновению направления в искусственном интеллекте под названием «Компьютерное зрение».

Компьютерное зрение – это научное направление по созданию аппаратно-программных комплексов, способных производить обнаружение, локализацию и распознавание объектов.

Технологии компьютерного зрения применяются для решения задач из разных сфер деятельности. В качестве примеров можно привести следующие направления применения технологий компьютерного зрения:

- обеспечение безопасности на массовых мероприятиях;
- автоматизация охранных функций на различных объектах;
- верификация клиентов банков по изображению;
- автоматическое улучшение качества фотографий;
- поиск информации на основе изображений;
- установление связей людей по фотографиям из социальных сетей.

Одной из современных задач компьютерного зрения является распознавание эмоций человека на основе анализа изображения с его лица. Технология распознавания эмоций может применяться для автоматизированной оценки реакции человека на внешние раздражители. Сфер применения у данной технологии большое множество. Например, в образовании, по мимике лица можно контролировать уровень вовлеченности студентов в образовательный процесс. В различных социальных сетях распознавание эмоций может быть использовано для персонализированных рекомендаций контента (по эмоциям пользователя можно определять какой контент ему нравится, а какой нет). Также распознавание эмоций может применяться как дополнительный инструмент при прохождении тестирования на полиграфе.

Для поддержания интереса к проблеме распознавания эмоций на изображении ежегодно проводятся соревнования по разработке и тестированию алгоритмов.

В таблице 1 представлены последние известные соревнования по разработке и тестированию алгоритмов для распознавания эмоций на изображениях.

Таблица 1 – Соревнования по разработке алгоритмов для распознавания эмоций на изображениях

Название соревнования	Тип анализируемых данных	Количество классов эмоций	Последний год проведения
AFEW-VA	Видео	2 класса	2022
EmotioNet challenge	Фотографии	16 классов	2020
OMG-Emotion challenge	Аудио и видео	7 классов	2018
6th Emotion Recognition in the Wild Challenge (EmotiW)	Аудио и видео	6 классов	2018
EmoReact	Аудио и видео	17 классов	2018
AffectNet	Фотографии	7 классов	2017

Как видно из таблицы 1, весь спектр эмоций можно разделить на разное количество классов. Общий подход по представлению и классификации эмоций человека описал в своей работе Дж. Рассел. В ней он выделил два базовый параметра любой эмоции – «Интенсивность / arousal» и «Знак / valance». На окружности, центр которой расположен на пересечении осей arousal и valance расположено названия эмоций. Графическое представление спектра эмоций по модели Дж. Рассела показано на рисунке 1. Примером положительной эмоцией (большое положительное значение параметра valance) с низкой интенсивностью (близкое к нулю значение arousal) является «удовлетворенность».

Модель представления эмоций Дж. Рассела часто используется при разметке обучающей выборки данных. Все наборы данных, представленные в соревнованиях, описанных в таблице 1, размечены с использованием этой модели.

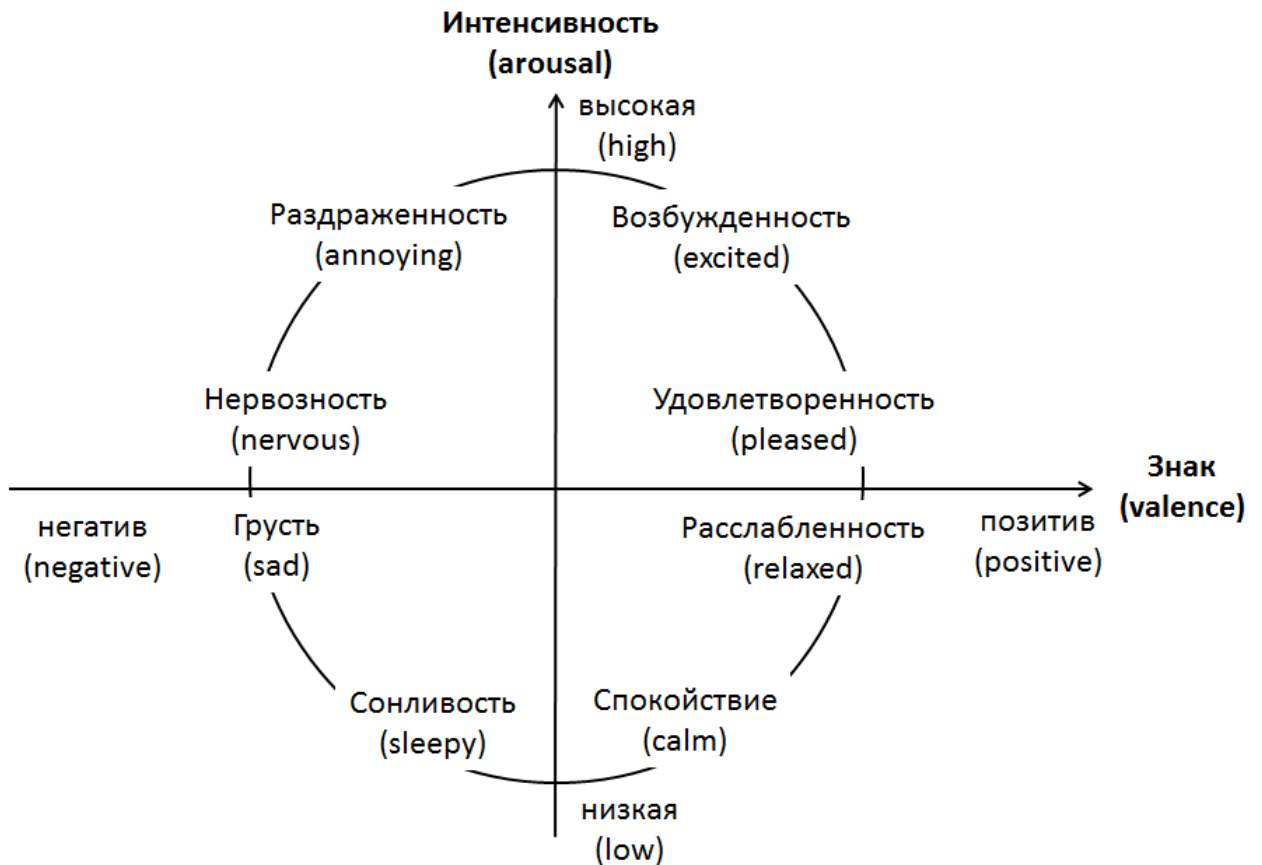


Рисунок 1 – спектр эмоций по модели Дж. Рассела

Обучающая выборка данных необходима для настройки алгоритмов распознавания эмоций. Для формирования обучающей выборки данных берутся статичные изображения, аудио и видео данные, на которых показаны различные эмоция, и размечаются вручную. В ходе ручной разметки данных специалист должен определить, какая эмоция показана на объекте из выборки данных, а также записать соответствующую метку класса эмоции в систему учета.

Разметка данных для формирования обучающей выборки данных является трудоемким и долгим процессом. Одни и те же изображения можно разметить по-разному в зависимости от выбранного количества категорий эмоций. Например, можно при разметке объектов обучающей выборки использовать только 4 категории эмоций, разделив все эмоциональные оттенки на группы так, как это показано в верхней части рисунка 2.

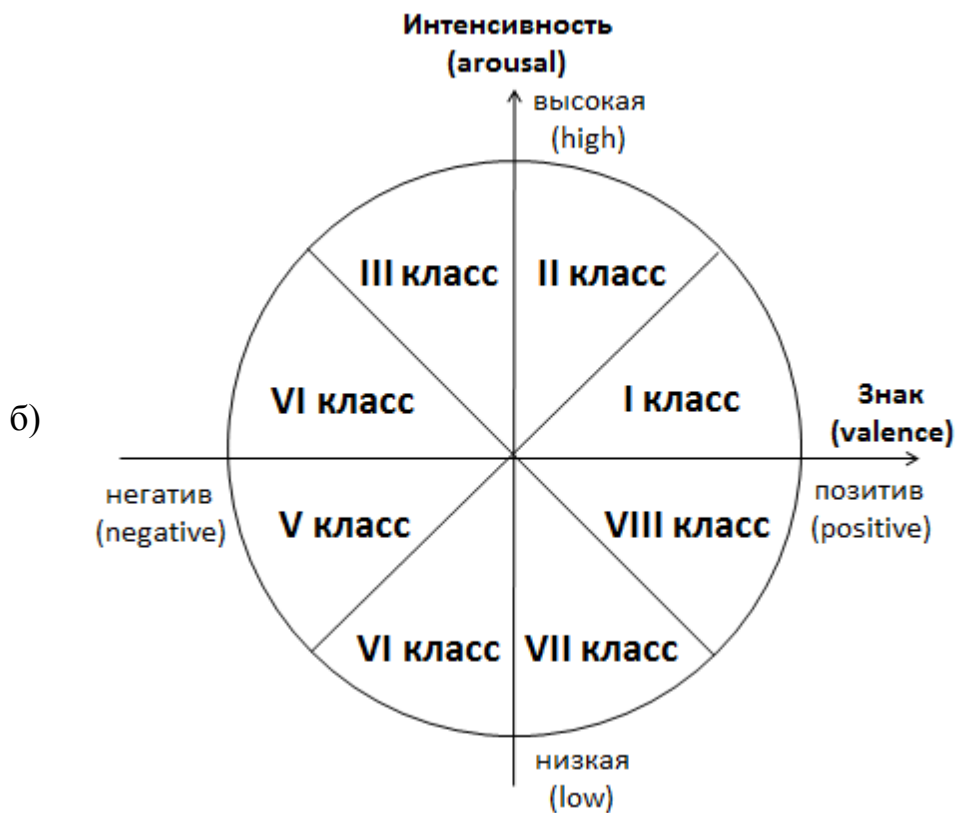
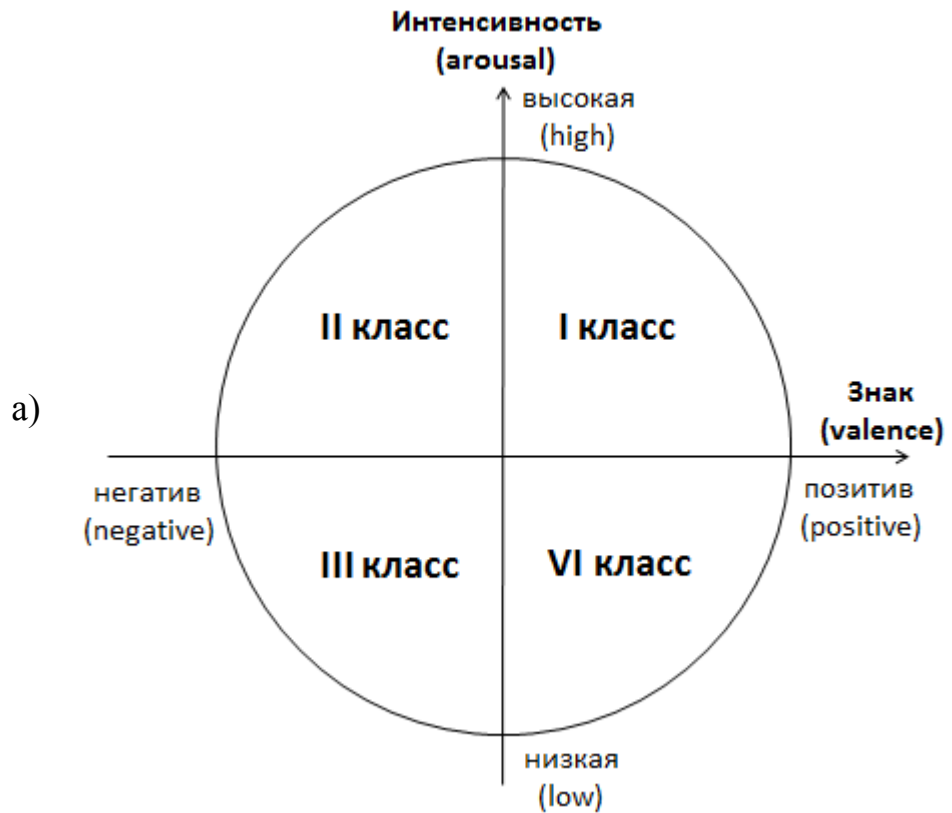


Рисунок 2 – Распознавание эмоций как задача классификации при разном количестве классов в исходных данных: а – при 4 классах, б – при 8 классах

В некоторых случаях может потребоваться более точная классификация эмоций. В этом случае требуется увеличить количество классов, например, до 8 (нижняя часть рисунка 2). Это увеличит трудоемкость разметки обучающей выборки, но, в конечном счете, алгоритм распознавания эмоций, обучающийся на этих данных, сможет различать больше эмоциональных оттенков.

Во всех соревнованиях алгоритмов распознавания эмоций, показанных в таблице 1, применялись уникальные обучающие выборки, размеченные с использованием различного количества классов (от 2 до 17).

Стоит также отметить, что распознавание эмоций может быть представлено в виде регрессионной задачи. В этом случае требуется на основе анализа изображения лица определить числовые значения компонентов *valence* и *arousal*. На основе этих значений можно точно определить оттенок эмоции, воспользовавшись моделью Дж. Рассела (рисунок 3).

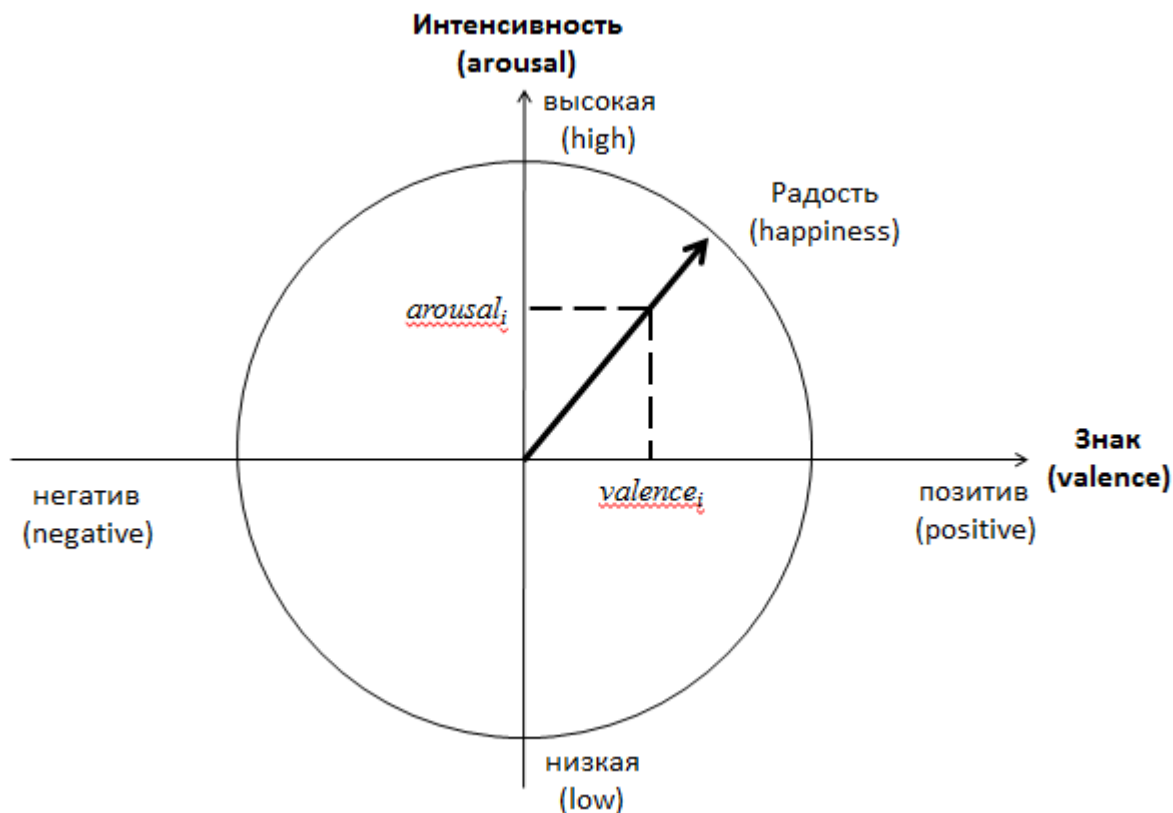


Рисунок 3 – Распознавание эмоций как задача регрессионного анализа

1.2 Сравнение алгоритмов распознавания эмоций

Анализ лучших решений, представленных на соревнованиях AFEW-VA, EmotioNet challenge, OMG-Emotion challenge, 6th Emotion Recognition in the Wild Challenge (EmotiW) и EmoReact, AffectNet по разработке алгоритмов распознавания эмоций показал следующие особенности по их реализации. Работа большей части представленных алгоритмов состоит из выполнения 3 этапов: загрузка изображения, поиск на изображении ключевых точек лица и анализ взаимного расположения ключевых точек лица для определения класса эмоций [7,8].

На первом этапе загружаемое изображение преобразуется в трехмерный числовой массив. Размерность массива зависит от высоты и ширины изображения, а также от цветовой модели изображения. В используемых на соревнованиях наборах данных все изображения уже приведены к единому размеру и формату.

На втором этапе производится анализ числового массива данных с целью обнаружения ключевых точек лица на изображении. Под ключевыми точками понимаются – контур лица (в некоторых случаях - контур скул), контур бровей, контур носа, контур границ глаз, контур границ губ. Параметрами ключевых точек лица, являются: координаты в пространстве (3 числовых значения), типы точек в зависимости от элемента лица с которым они связаны (1 числовое значение), номер точки из последовательности (1 числовое значение). Пример описания ключевых точек для контура левой брови (массив `left_eyebrow`) показан ниже (1):

$$\text{left_eyebrow} = \begin{pmatrix} 155 & 203 & 201 & 2 & 1 \\ 160 & 206 & 202 & 2 & 2 \\ 164 & 207 & 202 & 2 & 3 \\ 169 & 206 & 203 & 2 & 4 \\ 171 & 204 & 202 & 2 & 5 \end{pmatrix} \quad (1)$$

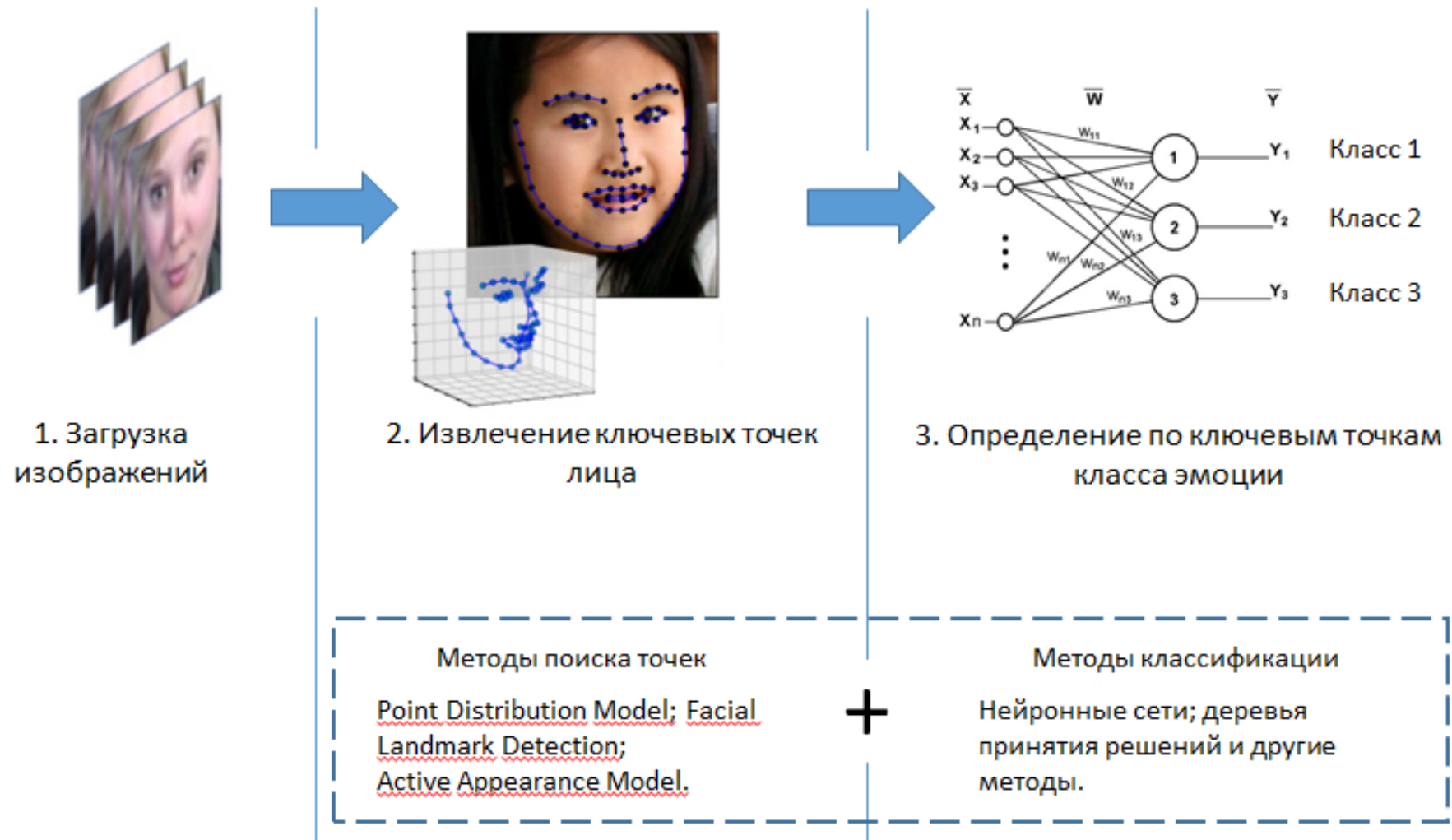


Рисунок 4 – Основные этапы классификации эмоций с использованием технологии определения ключевых точек лица

На третьем этапе, проводится классификация полученных числовых значений с использованием одного из известных методов. Основные этапы классификации эмоций с использованием технологии определения ключевых точек лица показаны на рисунке 4.

Основным отличием алгоритмов распознавания эмоций друг от друга заключается в использовании различных сочетаний метода поиска ключевых точек лица и метода классификации данных. При таком подходе, задача разработки алгоритма распознавания эмоций сводится к подбору оптимального сочетания методов поиска ключевых точек и классификации под имеющуюся обучающую выборка размеченных изображений. Так, например, если в обучающей выборке данных содержатся изображения лиц, находящихся под большим углом к плоскости изображения, и часть лица перекрыта волосами, рукой или находится в тени, то необходимо использовать методы поиска ключевых точек лица, умеющие самостоятельно моделировать скрытые области лица (рисунок 5).

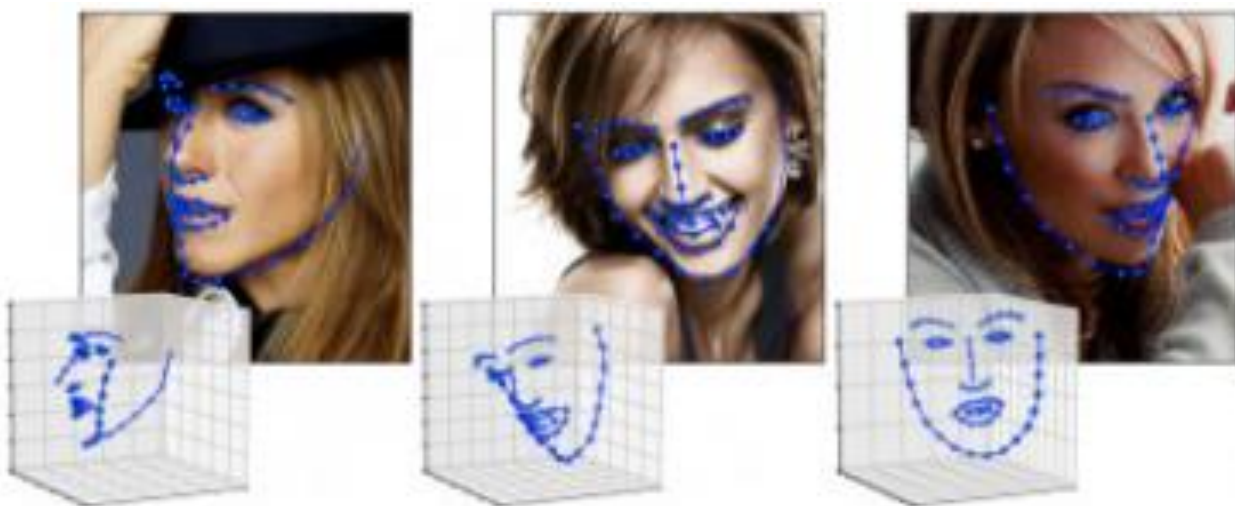



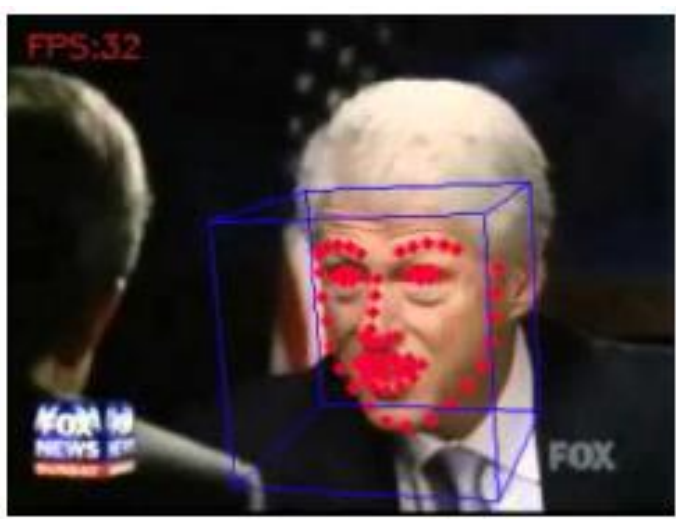
Рисунок 5 – Определение ключевых точек при больших углах расположения лица к плоскости фотографии, а также при частичном перекрытии лица.\

Наиболее известными алгоритмами поиска ключевых точек лица на изображении являются: Point Distribution Model, Facial Landmark Detection,

Active Appearance Model и 2D-and-3D-face-alignment. Примеры результатов работы данных алгоритмов показаны в таблице 2. Данные алгоритмы отличаются друг от друга математическим аппаратом, количеством ключевых точек и наличием различных ограничений, связанных с положением лица на анализируемом изображении.

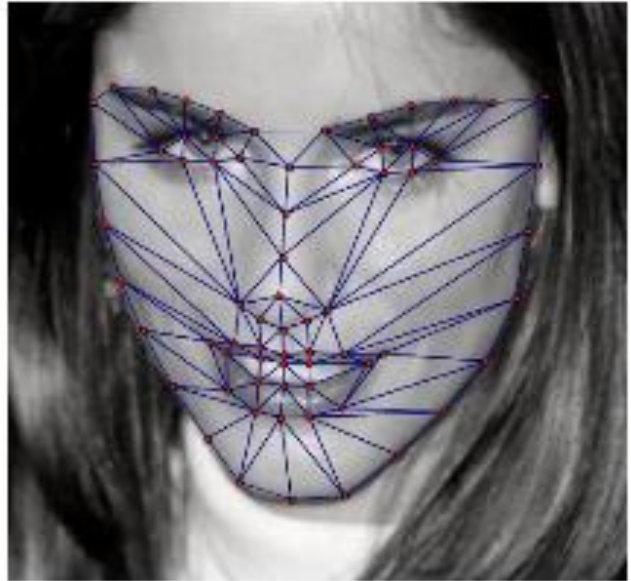
Существуют и другие методы поиска ключевых точек лица, однако в силу их малой популярности в данной работе они рассматриваться не будут.

Таблица 2 – Методы определения ключевых точек лица на изображении

Название метода / веб-сайт	Пример работы метода поиска ключевых точек лица
<p>Point Distribution Model</p> <p>https://www.menpo.org/menpofit/pdm.html</p>	
<p>Facial Landmark Detection</p> <p>https://github.com/TadasBaltrusaitis/OpenFace</p>	

Active Appearance Model

<https://www.menpo.org/menpofit/aam.html>



2D-and-3D-face-alignment

<https://github.com/1adrianb/2D-and-3D-face-alignment>



Для того чтобы связать данные о положении ключевых точек лица с проявляемыми эмоциями необходимо обучить классификатор. В качестве классификаторов при решении данной задачи обычно используются: нейронные сети прямого распространения (neural networks), деревья принятия решений (decision trees) или классификатор SVM (support vector machine).

При обучении нейронной сети прямого распространения производится настройка весовых коэффициентов связей, соединяющих нейроны различных слоев. В процессе обучения нейронной сети эти коэффициенты подбираются таким образом, чтобы нейронная сеть максимально точно работала на обучающей выборке данных. В качестве алгоритма обучения чаще всего используется метод обратного распространения ошибки [2-4]. Структура нейронной сети состоит из:

- входного слоя X , размер которого соответствует суммарному количеству n параметров всех ключевых точек лица;
- одного или нескольких скрытых слоев, которые реализуют вычисления нейронной сети;
- выходного слоя Y , в котором количество нейронов равно количеству классов эмоций в обучающей выборке данных [6].

Пример нейронной сети прямого распространения, способной распознавать 3 класса эмоций на основе данных ключевых точек лица, показан на рисунке 6.

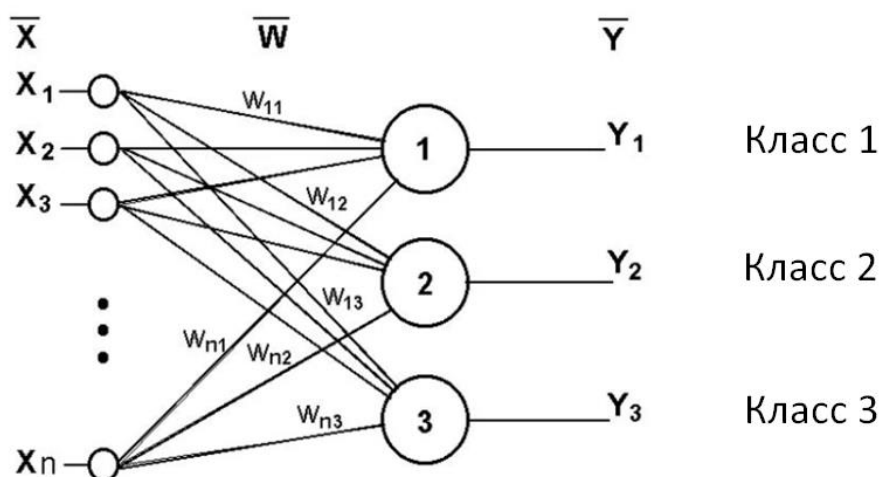


Рисунок 6 – Пример нейронной сети для распознавания 3 классов эмоций по данным ключевых точек лица

Дерево принятия решений также можно использовать в роли классификатора данных. Дерево решений – это граф, в узлах которого

содержаться условия перехода по ребрам к следующим узлам дерева. Работа такого классификатора начинается с определения последовательности переходов от корневого узла до одного из листьев. В листе содержится метка класса, которая и возвращается пользователю как результат классификации данных. Для построения деревьев принятия решений применяются различные алгоритмы, среди которых наиболее популярны: CART и C5.0. Таким образом на основе перемещения по узлам дерева решений по данным ключевых точек лица осуществляется определение класса эмоций (рисунок 7).

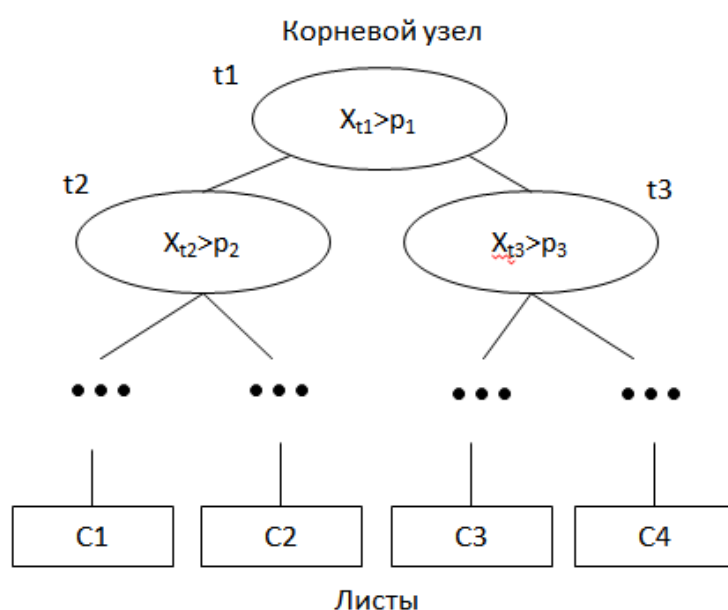


Рисунок 7 – Пример фрагмента дерева принятия решений: $t_1, t_2 \dots t_3$ – обозначение узлов; $X_{t_1}, X_{t_2} \dots X_{t_3}$ – параметры ключевых точек лица; C_1, C_2, C_3, C_4 – обозначение классов распознаваемых эмоций

Работа классификатора SVM основана на построении гиперплоскости в пространстве параметров ключевых точек лица, способной отделить различные классы эмоций.

Таким образом, большинство алгоритмов для распознавания эмоций основано на идее необходимости определения ключевых точек лица на изображении. Такой подход используется по следующим причинам:

- простота идеи - в реальном мире определение эмоций человека возможно за счет наблюдения за мимикой лица;

- технические ограничения методов классификации – они не адаптированы под анализ изображений, так как графическая информация содержит большое количество признаков данных.

Для анализа графической информации были разработаны сверточные нейронные сети. По идее, такой тип нейронных сетей за счет слоёв свертки способен самостоятельно определять значимые признаки изображения, а с помощью полносвязного слоя нейронов – способен решать задачу классификации данных. По идее, связку метода поиска ключевых точек лица и метода классификации можно заменить на сверточную нейронную сеть. И это позволит не только упростить математический аппарат распознавания эмоций на изображении, но и облегчить настройку классификатора эмоций за счет использования стандартных алгоритмов обучения нейронных сетей.

Выводы по главе 1

Приведем выводы по первой главе бакалаврской работы:

- большинство алгоритмов распознавания эмоций основано на совместной работе двух методов: метода поиска ключевых точек лица на изображении (Point Distribution Model, Facial Landmark Detection, Active Appearance Model и 2D-and-3D-face-alignment) и метода классификации данных (нейронные сети, деревья принятия решений, классификатор SVM);

- предложено связку метода поиска ключевых точек лица и метода классификации можно заменить на сверточную нейронную сеть, что позволит облегчить настройку классификатора эмоций за счет использования стандартных алгоритмов обучения нейронных сетей.

Глава 2 Проектирование алгоритма для анализа эмоций на изображении

2.1 Алгоритм нейросетевого распознавания эмоций

Распознавание класса эмоций человека по изображению его лица относится к задаче классификации. Формальное описание задачи классификации звучит следующим образом. Существует заранее известное и фиксированное множество классов. Каждый класс является отдельным типом эмоций. Для распознавания эмоций необходимо сформировать классификатор, способный на основе изображения, подаваемого на его вход определять номер класс эмоции, испытываемый человеком на изображении. В качестве классификатора в данной работе предложено использовать сверточную нейронную сеть.

Сверточные нейронные сети могут иметь различную архитектуру, причем заранее неизвестно, какая из архитектур покажет наилучшие результаты классификации на имеющемся наборе изображений. Чтобы подобрать архитектуру сверточной сети необходимо выбрать стратегию ее проектирования [9].

В данной работе выбраны следующие подходы для определения структуры нейронной сети:

- отдавать предпочтение использованию структуры нейронной сети с меньшим количеством слоев при условии сохранения ее работоспособности на имеющемся наборе данных;
- добавлять дополнительные слои в нейронную сеть только при невозможности улучшить результаты работы сети другими методами;
- для увеличения скорости обучения нейронной сети и упрощения ее математического аппарата отдавать предпочтения более простым

функциям активации – функция \max для слоев подвыборки, функция ReLU для промежуточных слоев сети;

– отдается предпочтение модифицированию обучающей выборки изображений вместо усложнения структуры нейронной сети.

На основе описанных выше принципов принятия проектных решений определим структуру нейронной сети.

Первый слой любой нейронной сети всегда является входным. Это значит, что на него подаются входные данные, которые впоследствии обрабатываются нейронной сетью. В нашем случае входными данными будут являться фотографии лиц, поэтому количество нейронов входного слоя следует выбирать в зависимости от параметров изображения:

- цветовая модель изображения (RGB, Lab или оттенки серого).
- количество пикселей изображения (размер изображения);

При анализе некоторых изображений из других предметных областей цвет пикселей может играть значительную роль в их классификации. Например, при распознавании дорожных знаков цвет элементов на изображении играет важную роль. И в этом случае, количество входных нейронов будет рассчитываться как произведение высоты на ширину изображения в пикселях, умноженное на количество компонентов пикселя, описывающего его цвет. Например, в цветовой модели RGB количество компонентов, описывающих цвет одного пикселя, равно трём. В задаче анализа эмоций, цвет не несет в себе значимых признаков. Поэтому для снижения количество нейронов во входном слое можно анализировать изображения без учёта цвета (изображение в оттенках серого). Отказ от цвета является обычной практикой в системах компьютерного зрения.

Так как мы будем анализировать изображения без учета световой составляющей, то количество нейронов входного слоя будет равно количеству пикселей анализируемого изображения. Количество входных нейронов постоянно и не подлежит изменению после настройки нейронной

сети. Если количество пикселей у анализируемого изображения больше, чем нейронов во входном слое, то исходное изображение уменьшают (масштабируют) до нужного размера, перед тем как подать на вход нейронной сети.

Так как обучение нейронной сети будет производиться на существующем наборе данных Facial Expression Recognition Challenge, то размер входного слоя был выбран равный размеру имеющихся в нем изображений – 48×48 пикселей.

Основными слоями сверточной нейронной сети являются: слой свертки и слой подвыборки. Первый обеспечивает выделение признаков на числовом наборе данных, а второй обеспечивает снижение размерности признакового пространства. Минимальное количество подряд идущих блоков состоящих из сочетания этих слоев равно двум. Поэтому включим их в нашу структуру.

Обязательным элементом сверточной нейронной сети является слой снижения размерности, который преобразует все данные предыдущего слоя в вектор. В конце сверточной нейронной сети используется обычный слой нейронов производящий классификацию данных, выделенных из изображения с помощью предыдущих слоев сети.

Таким образом, первоначальная структура нейронной сети будет состоять из следующих слоев.

- входной слой для получения изображения в виде числового массива $48 \times 48 \times 1$;
- сверточный слой (convolution) с ядром 5×5 , шагом свертки 2, количеством плоскостей (каналов) 32 и функцией активации ReLU;
- слой подвыборки (max-pooling) с группой уплотнения 2×2 ;
- сверточный слой (convolution) с ядром 3×3 , количеством плоскостей (каналов) 64 и функцией активации ReLU;
- слой подвыборки (max-pooling) с группой уплотнения 2×2 ;

- слой снижения размерности (flatten) данных до одномерного вектора;
- слой (core layer ReLU) из 500 нейронов прямого распространения с функцией активацией ReLU;
- выходной (core layer SoftMax) слой с функцией активацией SoftMax.

Графически архитектура нейронной сети показана на рисунке 8.

Рассмотрим математический аппарат работы слоев нейронной сети.

Размер ядра сверточного слоя определяется с использованием системы показанной на (1):

$$\begin{cases} w_c = w_u - K + 1 \\ h_c = h_u - K + 1 \end{cases}, \quad (1)$$

где w_c, h_c – ширина и высота сверточной плоскости, w_u, h_u – ширина и высота матрицы сигналов предыдущего слоя, K – ширина ядра свертки.

В качестве функции активации используется Rectified linear unit (ReLU):

$$f(a) = \begin{cases} 0 & a < 0 \\ a & a \geq 0 \end{cases}, \quad (2)$$

где $f(a)$ – искомое значение элемента, a – взвешенная сумма сигналов предыдущего слоя.

Данная функция является монотонной, с монотонной производной и не аппроксимирует тождественную функцию около начала координат.

В качестве преимуществ данной функции ReLU по сравнению с другими вариантами функции активации (гиперболический тангенс, логистическая функция) можно отметить следующее:

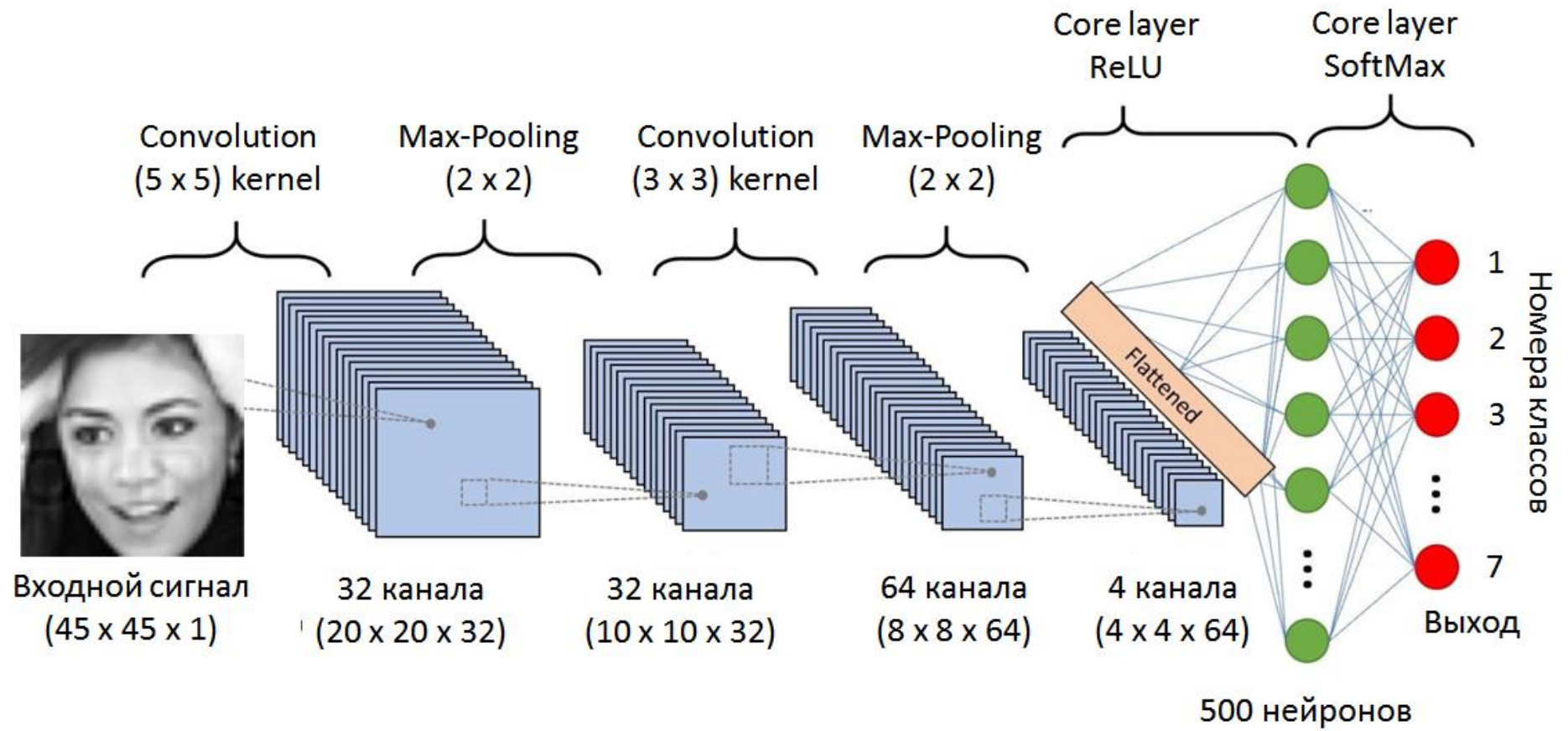


Рисунок 8 – Выбранная архитектура нейронной сети

- данная функция является наиболее простой с точки зрения математики;
- функция имеет простую производную, что дает экономию вычислений при обучении и работе нейронной сети.

Формула функционирования нейрона сверточного слоя (3):

$$y_k^{(i,j)} = b_k + \sum_{s=1}^K \sum_{t=1}^K w_{k,s,t} x^{((i-1)+s,(j+t))}, \quad (3)$$

где $y_k^{(i,j)}$ – нейрон k -й плоскости (k -ого канала) сверточного слоя, b_k – нейронное смещение k -й плоскости, K – размер рецептивной области нейрона (размер ядра свертки), $w_{k,s,t}$ – матрица синаптических коэффициентов, x – выходы нейронов предыдущего слоя.

Схема работы сверточного слоя для цветного изображения с тремя каналами (RGB) показана на рисунке 9.

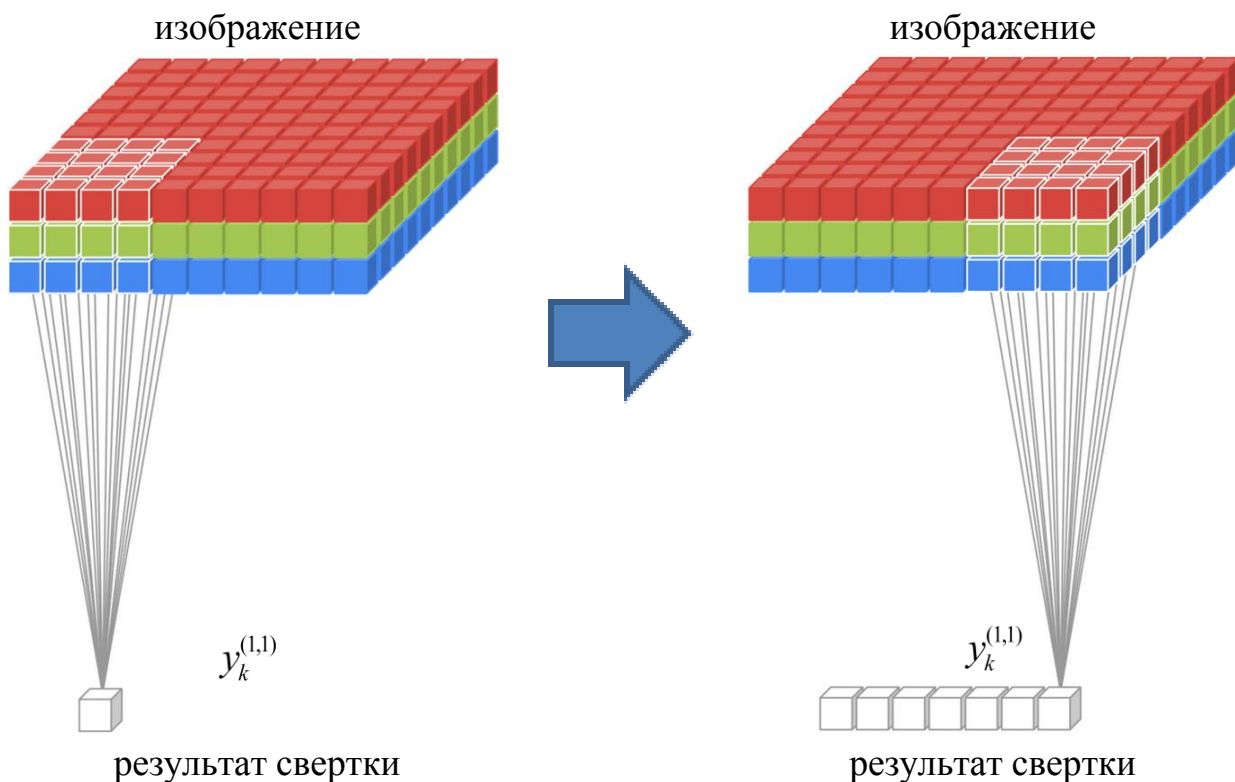


Рисунок 9 – Схема работы сверточного слоя k -й плоскости

Значения компонентов ядра определяется в ходе обучения сети. Начальные веса изначально имеют случайные значения, и корректируются в процессе обучения. Перемножение и суммирование повторяются для каждой локации, по которой проходит ядро.

Формула функционирования слоя подвыборки (подвыборочного слоя) с группой уплотнения 2×2 показана на (4):

$$y_k^{(i,j)} = b_k + w_k \cdot \max(x^{((2i,2j),(2i,2j))}, x^{((2i,2j),(2i,2j)+1)}, x^{((2i,2j)+1,(2i,2j))}, x^{((2i,2j)+1,(2i,2j)+1)}) \quad (4)$$

Пример выполнения операции подвыборки с использованием функции $\max()$ с группой уплотнения 2×2 показан на рисунке 10.

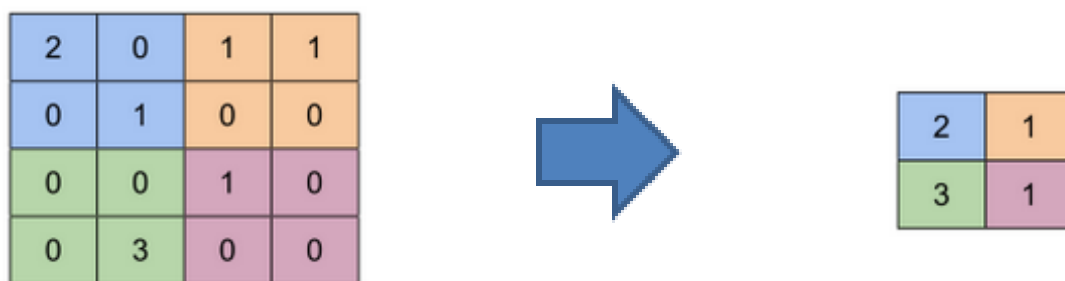


Рисунок 10 – Пример выполнения субдискретизации с использованием функции $\max()$ с группой уплотнения 2×2 (группы выделены цветом)

Аналогичным образом, на основе формул (1-4) работают второй сверточный слой и второй слой подвыборки.

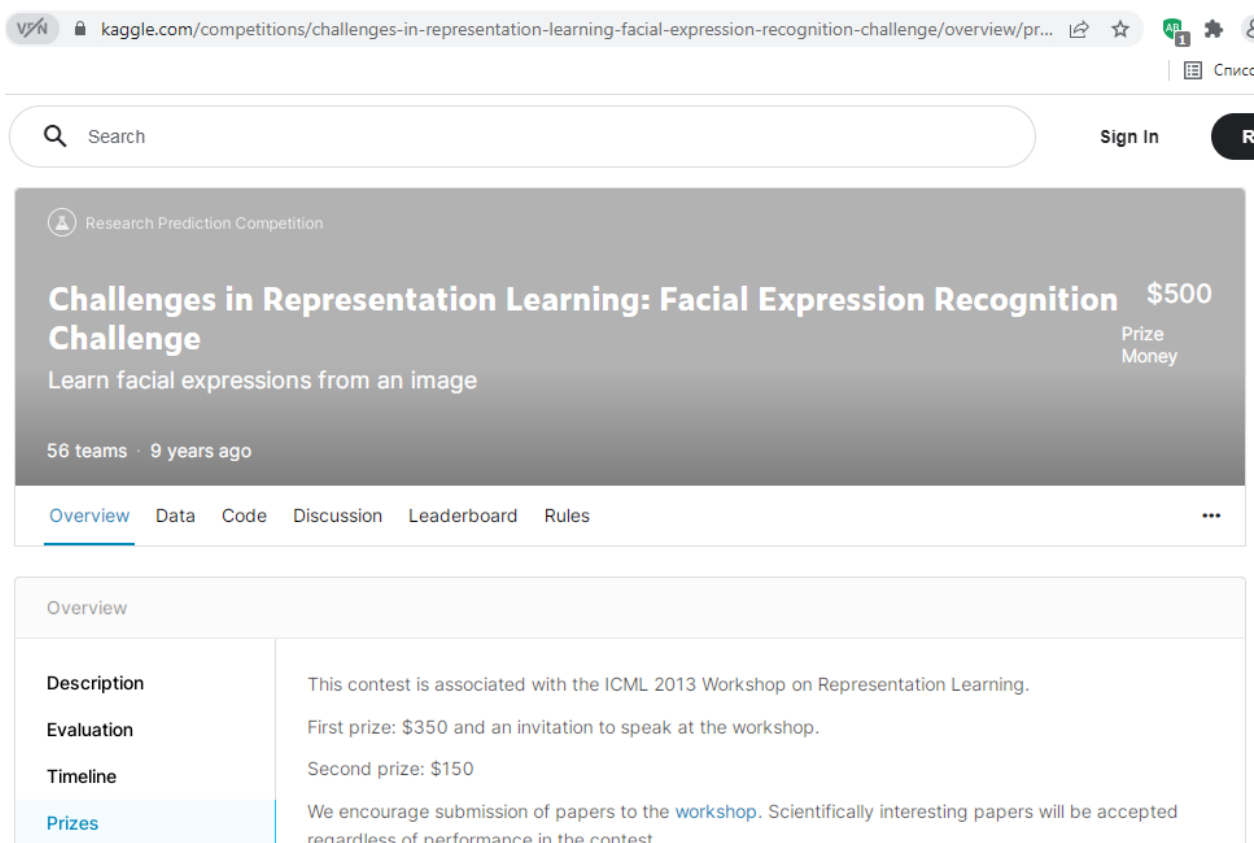
Слой снижения размерности (flatten) выполняет единственное действие – преобразует данные предыдущего слоя в одномерный вектор.

После слоя снижения размерности фактически идет стандартная нейронная сеть прямого распространения с одним скрытым слоем, состоящего из 500 нейронов. В выходном слое состоит из 43 нейронов (43 класса изображений) с функцией активации SoftMax. Таким образом, на выходе каждого нейрона выходного слоя будет число выражающее степень уверенности сети в каждом из классов для рассматриваемого изображения.

2.2 Источник данных для обучения нейронной сети

Обучение нейронной сети должно осуществляться на основе размеченной выборки данных. Чем сложнее нейронная сеть, тем больше должна быть обучающая выборка. Для выбранной архитектуры нейронной сети приблизительно количество обучающих примеров, которое определяется на основе количества весовых коэффициентов, должно составлять приблизительно от 20 000 до 100 000 изображений.

Для обучения и тестирования нейронной сети было принято решение использовать уже подготовленный набор данных «Facial Expression Recognition Challenge», выложенный в открытый доступ на веб-странице соревнования ICML Workshop on Representation Learning (рисунок 11).

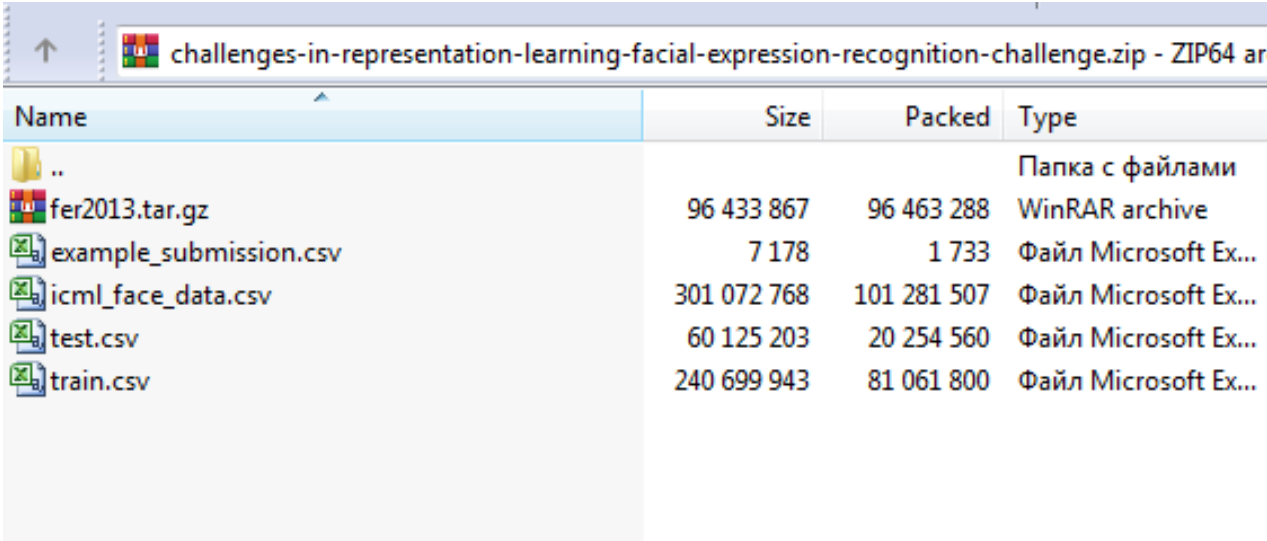


The image shows a screenshot of a web browser displaying a Kaggle competition page. The browser's address bar shows the URL: kaggle.com/competitions/challenges-in-representation-learning-facial-expression-recognition-challenge/overview/pr.... The page features a search bar, a 'Sign In' button, and a navigation menu with options: Overview, Data, Code, Discussion, Leaderboard, and Rules. The main content area is titled 'Challenges in Representation Learning: Facial Expression Recognition Challenge' with a prize money of '\$500'. Below the title, it says 'Learn facial expressions from an image' and '56 teams · 9 years ago'. A table provides details about the contest:

Overview	
Description	This contest is associated with the ICML 2013 Workshop on Representation Learning.
Evaluation	First prize: \$350 and an invitation to speak at the workshop.
Timeline	Second prize: \$150
Prizes	We encourage submission of papers to the workshop . Scientifically interesting papers will be accepted regardless of performance in the contest.

Рисунок 11 – Веб-странице соревнования ICML Workshop on Representation Learning

Архив с обучающим и тестовым наборами данных занимает порядка 698 МБ. Содержимое архива показано на рисунке 12.



The screenshot shows a ZIP archive viewer window titled 'challenges-in-representation-learning-facial-expression-recognition-challenge.zip - ZIP64 ar'. The main area displays a table of files and folders within the archive.

Name	Size	Packed	Type
..			Папка с файлами
fer2013.tar.gz	96 433 867	96 463 288	WinRAR archive
example_submission.csv	7 178	1 733	Файл Microsoft Ex...
icml_face_data.csv	301 072 768	101 281 507	Файл Microsoft Ex...
test.csv	60 125 203	20 254 560	Файл Microsoft Ex...
train.csv	240 699 943	81 061 800	Файл Microsoft Ex...

Рисунок 12 – Структура архива с данными

Архив содержит в себе следующий список элементов.

1. Файл `train.csv`, в котором хранится обучающая выборка данных с примерами 28709 изображений. Содержимое файла организовано следующим образом. Сначала в виде числа задан класс эмоции, представленной на изображении, а затем после запятой в кавычках идут значения пикселей самого изображения, представленного в виде вектора. Так как изображения размером 48×48 пикселей, то длина вектора – 2304 значений (рисунок 13).

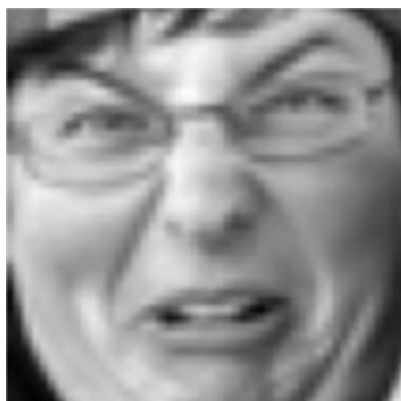
2. Файл `test.csv`, в котором хранятся данные для тестирования обученного классификатора эмоций. В нем содержатся данные 7178 изображений, организованные таким же образом как в файле `train.csv`.

3. Файл `icml_face_data.csv` является результатом объединения файлов `train.csv` и `test.csv`. Он необходим, если возникнет потребность самостоятельно разделить данные на обучающую и тестовую выборку.

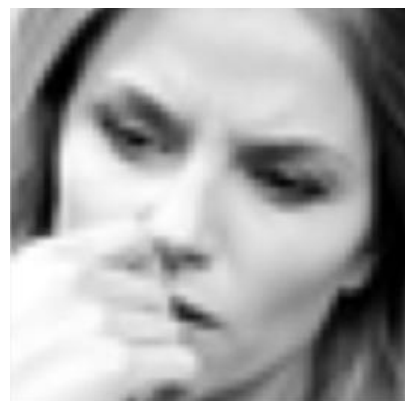
4. Файл `example_submission.csv` примеры того, как должен работать классификатор эмоций на изображении.



Класс «Angry» (злость)



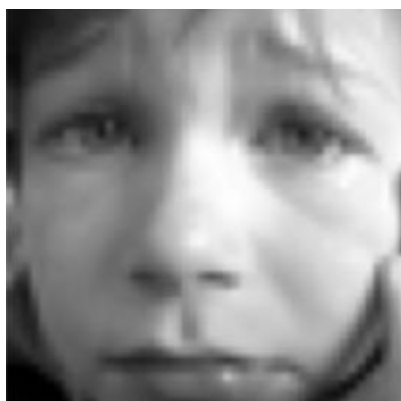
Класс «Disgust»
(отвращение)



Класс «Fear» (страх)



Класс «Happy»
(счастье)



Класс «Fear» (страх)



Класс «Surprise»
(удивление)



Класс «Neutral»
(спокойствие)

Рисунок 14 – Примеры изображений обучающей выборки для каждого класса

2.3 Оптимизация структуры нейронной сети и данных для обучения

Проектирование нейронной является итерационным процессом, это означает, что может потребоваться несколько итераций до того, как будет найдена оптимальная под имеющиеся данные архитектура сети.

Было проведено несколько циклов обучения нейронной сети с архитектурой, представленной на рисунке 8, на наборе данных «Facial Expression Recognition Challenge». По результатам оценки работы обученной сети на тестовом наборе данных было установлено, что средняя точность распознавания эмоций по изображениям лиц составляет 66,4%. Данные результаты были признаны неудовлетворительными, поэтому был проведен анализ причин низкой точности работы сети.

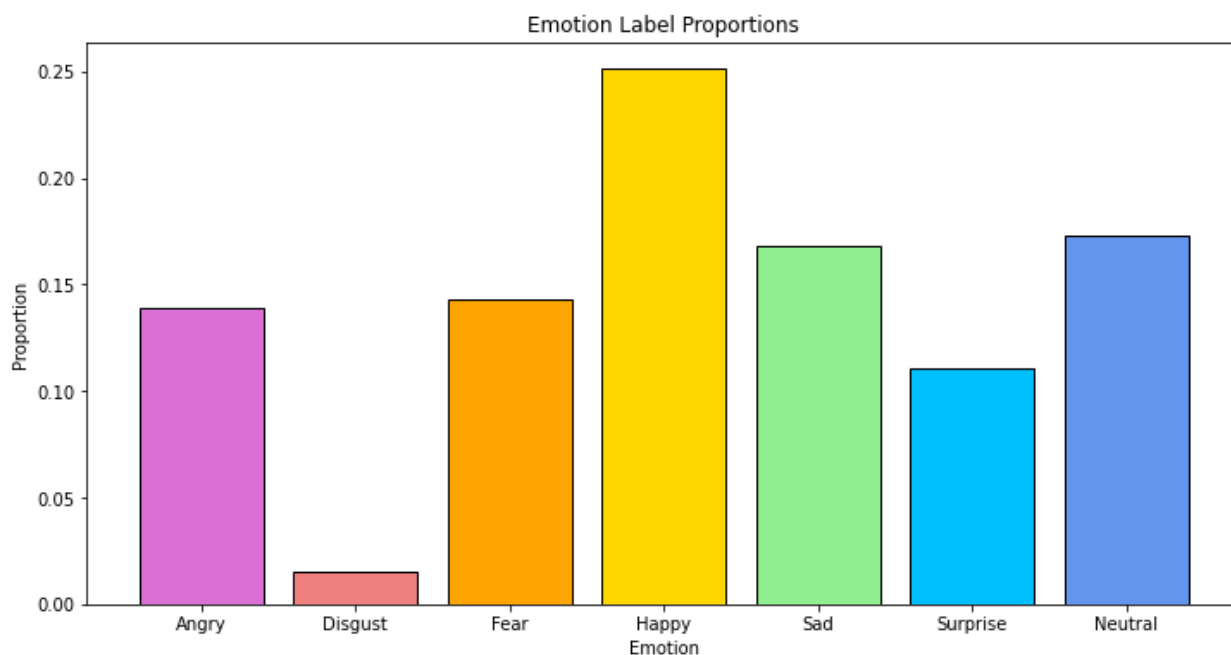


Рисунок 15 – Пропорции распределения изображений обучающей выборке по 7 классам

В ходе такого анализа изучались результаты работы на матрице ошибок показывающей, на изображения каких классов чаще всего возникает некорректная работа сети. В ходе тестирования выяснилось, что нейронная

сеть чаще всего ошибается на изображениях из класса «Disgust» (отвращение). Подсчитав программным способом распределение изображений в обучающей выборке по классам, была построена столбчатая диаграмма, показанная на рисунке 15. Из диаграммы видно, что класс «Disgust» является самым малочисленным, т.е. примеров в обучающей выборке недостаточно, для того, чтобы нейронная сеть научилась корректно работать с изображениями из данного класса.

Для решения данной проблемы необходимо увеличить численность примеров в данном классе. Для решения данной задачи было принято решение использовать процедурную генерацию новых изображений на основе уже имеющихся.

Для этого к существующим изображениям применяются последовательно операции вращения по оси и смещения углов изображения. С использованием такого подхода создавались новые примеры изображений, вплоть до увеличения доли класса «Disgust» до уровня «Surprise».

Примеры сгенерированных изображений показаны на рисунке 16

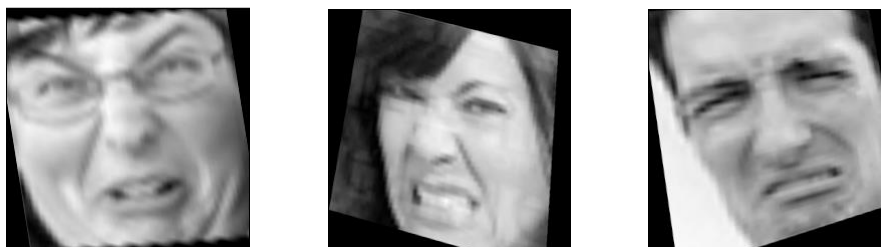


Рисунок 16 – Примеры синтезированных изображений, на основе имеющихся в обучающей выборке изображений (для класса эмоций «Disgust» / отвращение)

Такая генерация дополнительных изображений не может являться полноценной заменой расширения обучающей выборки за счет добавления уникальных изображений. И так как сгенерированные изображения во

многим перекликаются с оригинальными изображениями необходимо предпринять меры для защиты от эффекта переобучения сети.

В этом случае стандартным решением является добавление специальных слоев - «dropout». Данные слои функционируют следующим образом. В процессе обучения они случайным образом блокируют передачу заданной доли сигналов в следующий слой нейронов. Это заставляет нейронную сеть на каждой итерации обучения обращать внимание на разные признаки изображений, снижая доминирование одних признаков над другими. Это особенно важно, при выбранном методе увеличения изображений класса «Disgust» в обучающей выборке данных. Также стоит отметить, что слои «dropout» функционируют только в процессе обучения сети, при использовании обученной сети эти слои не оказывают никакого влияния на сигналы.

Таким образом, оптимизированная структура нейронной сети включает в себя следующие слои (слои dropout работают только в процессе обучения сети). Слои «dropout» были добавлены после двух слоев подвыборки и после одного слоя core layer ReLU.

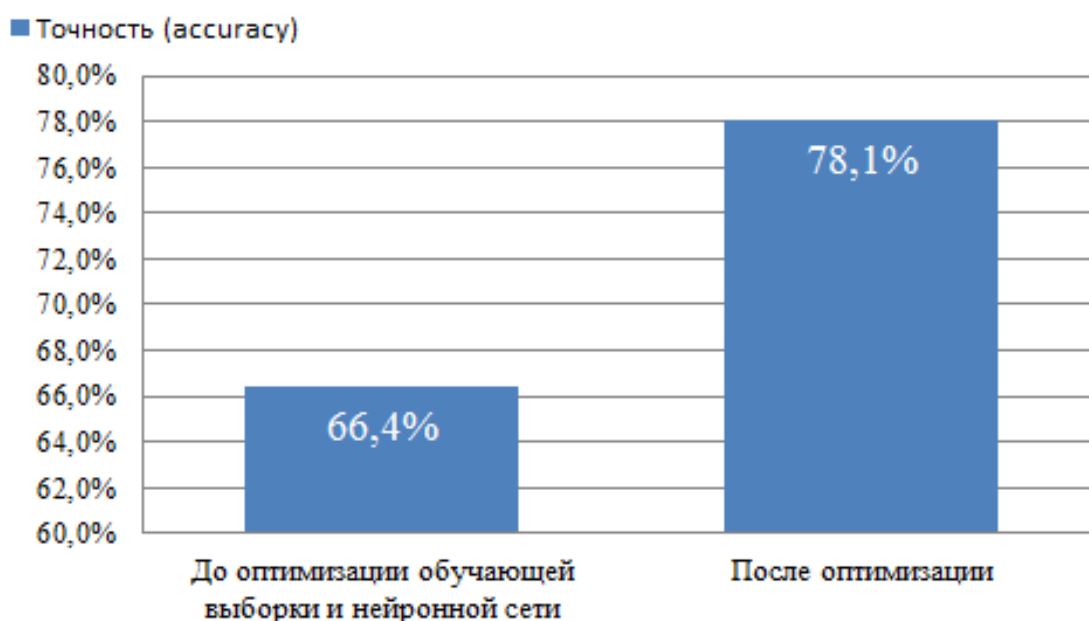


Рисунок 17 – Влияние оптимизации на точность работы сети на тестовой выборке данных

При повторном обучении нейронной сети на выборке данных с дополненным классом «Disgust» и использованием слоев «dropout» были получены следующие результаты. Скорость обучения снизилась в 1,5 раза, однако точность распознавания эмоций на тестовой выборке изображений увеличилась до 78,1% (рисунок 17).

Выводы по главе 2

Приведем выводы по второй главе бакалаврской работы:

– спроектирована максимально простая архитектура нейронной сети для решения задачи распознавания эмоций, которая включает в себя входной слой для получения изображения в виде числового массива $48 \times 48 \times 1$; сверточный слой (convolution) с ядром 5×5 , шагом свертки 2, количеством плоскостей (каналов) 32 и функцией активации ReLU; слой подвыборки (max-pooling) с группой уплотнения 2×2 ; слой dropout, блокирующий сигналы 25% нейронов предыдущего слоя; сверточный слой (convolution) с ядром 3×3 , количеством плоскостей (каналов) 64 и функцией активации ReLU; слой подвыборки (max-pooling) с группой уплотнения 2×2 ; слой dropout, блокирующий сигналы 25% нейронов предыдущего слоя; слой снижения размерности (flatten) данных до одномерного вектора; слой (core layer ReLU) из 500 нейронов прямого распространения с функцией активацией ReLU и выходной (core layer SoftMax) слой с функцией активацией SoftMax.

– Увеличение точности работы нейронной сети с 66,4% до 78,1% удалось достигнуть за счет добавления слоев dropout функционирующих только в процессе настройки сети и ослабляющих эффект переобучения, а также за счет синтезирования дополнительных изображений в самый малочисленный класс (класса эмоций «Disgust» / отвращение) изображений обучающей выборки.

Глава 3 Программная реализация алгоритма для распознавания эмоций на изображении

3.1 Описание разработанного приложения

Для реализации приложения нейросетевого распознавания эмоций был выбран язык программирования Python, так как для него существует большое количество свободно распространяемых по работе с изображениями и алгоритмами машинного обучения [17, 18, 20].

В качестве среды разработки приложения была выбрана Google colab, так как через эту среду обеспечивается удаленный доступ суперкомпьютеру, на котором обучение сверточной нейронной сети осуществляется на много быстрее, чем с использованием стандартных низкопроизводительных компьютеров и ноутбуков [12, 14].

Разработанное программное обеспечение реализует следующие функции:

- загрузка и подготовка к анализу данных с изображением лиц из выборки «Facial Expression Recognition Challenge»;
- генерирование дополнительных изображений в малочисленный класс на основе операций масштабирования, смещения и вращения;
- визуализация состава исходной выборки данных в виде столбчатой диаграммы
- моделирование работы сверточной нейронной сети, с архитектурой, представленной на рисунке 8;
- обучение сверточной нейронной сети в соответствии с данными из обучающей выборки;
- ведение лога процесса обучения с выводом статистической информации (номер итерации, затраченное время на итерацию);

- демонстрация результатов работы нейронной сети на выбранных данных;
- проверка точности работы нейронной сети на тестовом наборе данных.

3.2 Особенности программной реализации и примеры работы приложения

Для сокращения написания программного кода приложения по распознаванию эмоций в проект были включены различные внешние библиотеки. Код для подключения внешних библиотек показан на рисунке 18.

```
In [1]:  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
import matplotlib.image as mpimg  
import pickle  
  
from sklearn.model_selection import train_test_split  
from sklearn.metrics import classification_report  
  
import tensorflow as tf  
from tensorflow.keras import datasets, layers, models  
from tensorflow.keras.models import Sequential  
from tensorflow.keras.layers import *  
from tensorflow.keras.preprocessing.image import ImageDataGenerator  
  
from keras.models import Model
```

Рисунок 18 – Используемые в приложении библиотеки

При реализации программного обеспечения были использованы следующие библиотеки:

- библиотека `numpy`, обеспечивающая поддержку работы с массивами данных [15];
- библиотека `pandas` для отображения табличных данных [16];
- библиотека `matplotlib` для построения столбчатых диаграмм и графиков;
- библиотека `sklearn` для предобработки данных [13];
- библиотека `keras` для моделирования нейронной сети [19].

Следующим шагом после подключения библиотек является загрузка обучающей выборки из файла `train.csv` (рисунок 12). Для этого применяется метод `read_csv()`, реализованный в библиотеке `pandas`. Программный код показан на рисунке 19.

```

:
train = pd.read_csv('/kaggle/input/challenges-in-representation-learning-facial-expressi
print(train.shape)

(28709, 2)

```

Рисунок 19 – Загрузка тренировочных данных

Для проверки корректности загруженных данных воспользуемся методом `head()` для вывода фрагмента данных на экран. Код и результата вывода данных показан на рисунке 20

```

In [3]:
train.head()

Out[3]:

```

	emotion	pixels
0	0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121...
1	0	151 150 147 155 148 133 111 140 170 174 182 15...
2	2	231 212 156 164 174 138 161 173 182 200 106 38...
3	4	24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 1...
4	6	4 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84...

Рисунок 20 – Вывод тренировочных данных в виде числовых векторов

Как уже было сказано ранее, примеры изображений, хранящихся в файле train.csv изначально были представлены в виде одномерных векторов. Такая форма представления удобна для анализа данных, но она не подходит для вывода изображений на экран. Поэтому, одномерные векторы преобразуются в матрицы размером 48 на 48 значений для их отображения в виде растровых изображений (48 на 48 пикселей). Код показан на рисунке 21.

```
train['pixels'] = [np.fromstring(x, dtype=int, sep=' ').reshape(-1,48,48) for x in train['pixels']]

pixels = np.concatenate(train['pixels'])
labels = train.emotion.values

print(pixels.shape)
print(labels.shape)

(28709, 48, 48)
(28709,)
```

Рисунок 21 – Преобразование векторов в матрицы 48×48 пикселей

Следующим этапом является оценка распределения примеров обучающей выборки по классам. Код для оценки доли каждого класса в обучающей выборке показан на рисунке 22.

```
emotion_prop = (train.emotion.value_counts() / len(train)).to_frame().sort_index(ascending=True)

emotion_prop
```

	emotion
0	0.139155
1	0.015187
2	0.142708
3	0.251315
4	0.168240
5	0.110453
6	0.172942

Рисунок 22 – Оценка распределения примеров обучающей выборки по классам

Для удобства восприятия данные по распределению объектов по классам было принято визуализировать с помощью столбчатой диаграммы. Для этого дополнительно в виде вектора `emotions` были заданы названия классов, а с помощью `palette` – столбцов диаграммы. Код для построения столбчатой диаграммы показан на рисунке 23.

```
emotions = ['Angry', 'Disgust', 'Fear', 'Happy', 'Sad', 'Surprise', 'Neutral']

palette = ['orchid', 'lightcoral', 'orange', 'gold', 'lightgreen', 'deepskyblue', 'cornflowerblue']

plt.figure(figsize=[12,6])

plt.bar(x=emotions, height=emotion_prop['emotion'], color=palette, edgecolor='black')

plt.xlabel('Emotion')
plt.ylabel('Proportion')
plt.title('Emotion Label Proportions')
plt.show()
```

Рисунок 23 – Код для построения столбчатой диаграммы распределения объектов обучающей выборки по классам

Результат отображения построенной диаграммы показан на рисунке 15.

Для визуализации данных обучающей выборки необходимо вывести по несколько примеров изображений каждого класса. Для этого было создано 2 цикла `for`. Первый цикл осуществляет перебор номеров классов, а второй вложенный цикл отсчитывает изображения текущего класса и добавляет их в массив `subplot`.

После окончания выполнения циклов содержимое `subplot` выводится на экран в виде таблицы в помощью метода `show()`.

Программный код, для вывода по несколько примеров изображений обучающей выборки для каждого класса показан на рисунке 24. А сам результат вывода изображений показан на рисунке 25.


```

row = 0
for emotion in np.unique(labels):

    all_emotion_images = train[train['emotion'] == emotion]
    for i in range(5):

        img = all_emotion_images.iloc[i,].pixels.reshape(48,48)
        lab = emotions[emotion]

        plt.subplot(7,5,row+i+1)
        plt.imshow(img, cmap='binary_r')
        plt.text(-30, 5, s = str(lab), fontsize=10, color='b')
        plt.axis('off')
    row += 5

plt.show()

```

Рисунок 24 – Код для вывода примеров изображений, содержащихся в обучающей выборке

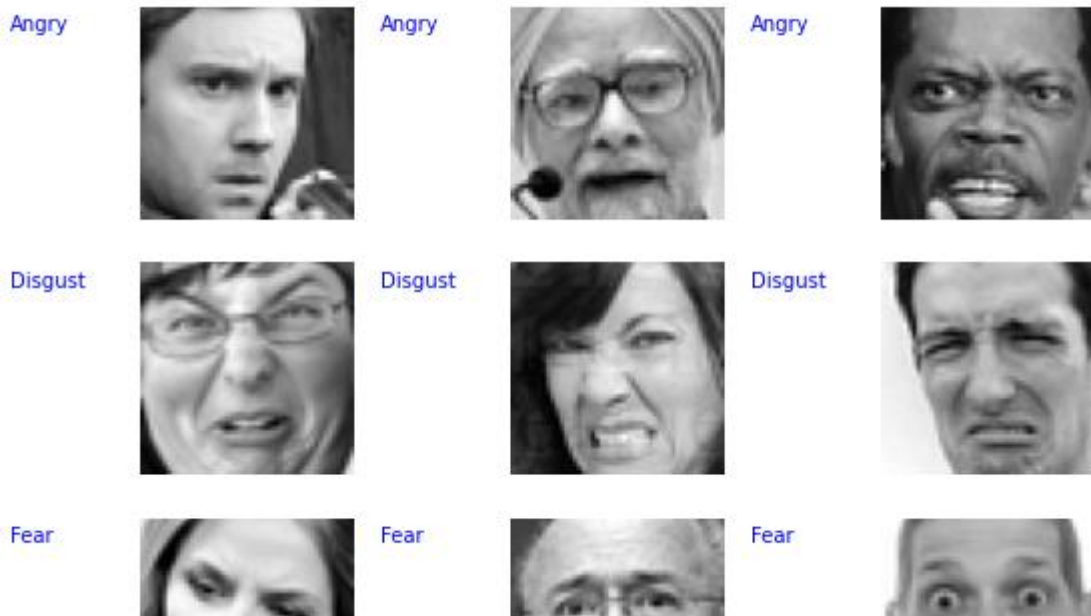


Рисунок 25 – Примеры изображений, содержащихся в обучающей выборке данных

Как отмечалось ранее, из-за того, что класс изображений «Disgust» является самым малочисленным, требуется увеличить его размерность за счет генерирования дополнительных изображений. Для этого применяется функция `ImageDataGenerator()` с параметрами, указанными на рисунке 26. Примеры сгенерированных таким образом изображений на основе имеющихся данных демонстрировались ранее на рисунке 16.

```
train_datagen = ImageDataGenerator(
    rotation_range = 40,
    width_shift_range = 0.3,
    height_shift_range = 0.3,
    zoom_range = 0.3,
    horizontal_flip = True,
    fill_mode = 'reflect'
)

train_loader = train_datagen.flow(rgb_X_train, y_train, batch_size=64)
```

Рисунок 26 – Код для генерирования новых и изображений в малочисленные классы на основе имеющихся изображений

Обучение сверточных нейронных сетей происходит эпохами. На каждой эпохе применяется свой фрагмент тренировочных данных и валидационных данных. Валидационные данные необходимо для того, чтобы контролировать прогресс обучения нейронной сети.

Таким образом исходный тренировочный набор данных необходимо разбить на тестовые и валидационные данные. Это шаг можно сделать с помощью функции `train_test_split()`. 80% данных использоваться в качестве тренировочных данных, а 20% - в качестве валидационных данных.

Программный код для выполнения данного шага показан на рисунке 27.

```

X_train, X_valid, y_train, y_valid = train_test_split(
    pixels, labels, test_size=0.2, stratify=labels, random_state=1
)

print('X_train Shape:', X_train.shape)
print('y_train Shape:', y_train.shape)
print()
print('X_valid Shape:', X_valid.shape)
print('y_valid Shape:', y_valid.shape)

```

```

X_train Shape: (22967, 48, 48)
y_train Shape: (22967,)

```

```

X_valid Shape: (5742, 48, 48)
y_valid Shape: (5742,)

```

Рисунок 27 – Код для разбиения обучающей выборки данных на тренировочные и валидационные данные

Теперь, когда валидационные и тренировочные данные подготовлены, можно переходить к заданию архитектуры нейронной сети. Для этого требуется вызвать конструктор `tf.keras`, описав в нем все слои сети.

Пример выбора стандартной архитектуры нейронной сети показан на рисунке 28.

```

resnet_model = tf.keras.applications.resnet50.ResNet50(
    include_top=False, weights='imagenet', input_shape=(48,48,3))

resnet_model.trainable = False

```

Рисунок 28 – Код для задания нейронной сети в keras на примере ResNet50

Также необходимо инициализировать алгоритм обучения нейронной сети и задать оцениваемый в процессе обучения параметр. В качестве алгоритма обучения используется алгоритм Adam, а в качестве оцениваемого параметра – точность работы нейронной сети. Код для задания этих параметров показан на рисунке 29.

```
opt = tf.keras.optimizers.Adam(0.0001)
cnn.compile(loss='sparse_categorical_crossentropy', optimizer=opt, metrics=['accuracy'])
```

Рисунок 29 – Задание алгоритма обучения сети

Для запуска процесса обучения сверточной нейронной сети используется команда `fit`. При вызове этой команды в качестве параметров можно, например, задать количество эпох обучения, валидационные данные. В процессе обучения нейронной будет выводиться различная статистическая информация о затраченном времени, текущем номере эпохи, достигнутой точности работы сети (рисунок 30).

```
h1 = cnn.fit(
    train_loader,
    batch_size=32,
    epochs = 30,
    verbose = 1,
    validation_data = (rgb_X_valid, y_valid)
)
```

```
2021-12-11 04:42:50.012801: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185]
n Passes are enabled (registered 2)
```

```
Epoch 1/30
```

```
2021-12-11 04:42:54.336578: I tensorflow/stream_executor/cuda/cuda_dnn.cc:369] Loaded cuDNN
```

```
359/359 [=====] - 36s 74ms/step - loss: 2.9185 - accuracy: 0.1577 -
racy: 0.3037
```

Рисунок 30 – Код для задания параметров обучения нейронной сети

После окончания обучения будет выведена информация о суммарном затраченном на обучение сети времени (рисунок 31).

```
CPU times: user 24min 27s, sys: 14.8 s, total: 24min 42s
Wall time: 17min 23s
```

Рисунок 31 – Вывод статистики по затраченному времени на обучение нейронной сети

Для контроля успешности обучения сети разработан программный код, строящий график изменения точности работы нейронной сети в течение различных эпох. При этом на графике демонстрируется 2 линии: первая – это изменения точности нейронной сети на тренировочных данных, в второй – на валидационных данных. Программный код для построения графика показан на рисунке 32.

```
for k in history.keys():
    history[k] += h2.history[k]

epoch_range = range(1, len(history['loss'])+1)

plt.figure(figsize=[14,4])
plt.subplot(1,2,1)
plt.plot(epoch_range, history['loss'], label='Training')
plt.plot(epoch_range, history['val_loss'], label='Validation')
plt.xlabel('Epoch'); plt.ylabel('Loss'); plt.title('Loss')
plt.legend()
plt.subplot(1,2,2)
plt.plot(epoch_range, history['accuracy'], label='Training')
plt.plot(epoch_range, history['val_accuracy'], label='Validation')
plt.xlabel('Epoch'); plt.ylabel('Accuracy'); plt.title('Accuracy')
plt.legend()
plt.tight_layout()
plt.show()
```

Рисунок 32 – Код для формирования графика обучения нейронной сети

График обучения нейронной сети, показывающий изменение точности ее работы в течение разных эпох на валидационных и тренировочных данных продемонстрирован на рисунке 32.

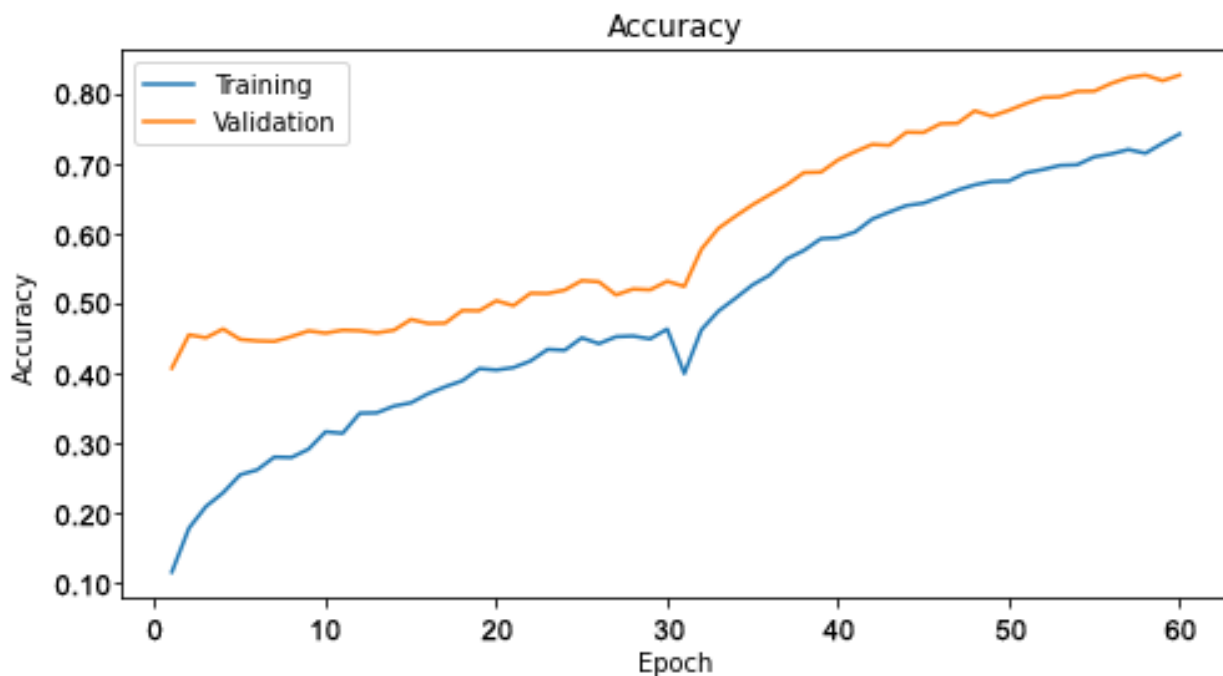


Рисунок 32 – График увеличения точности работы нейронной сети на валидационных и тренировочных данных в процессе обучения

Конечная точность обученной нейронной сети оценивалась на отдельном наборе данных test.csv (рисунок 12). Точность работы сети составила 78,1%. Примеры работы нейронной сети на тестовых изображениях показаны на рисунке 33.

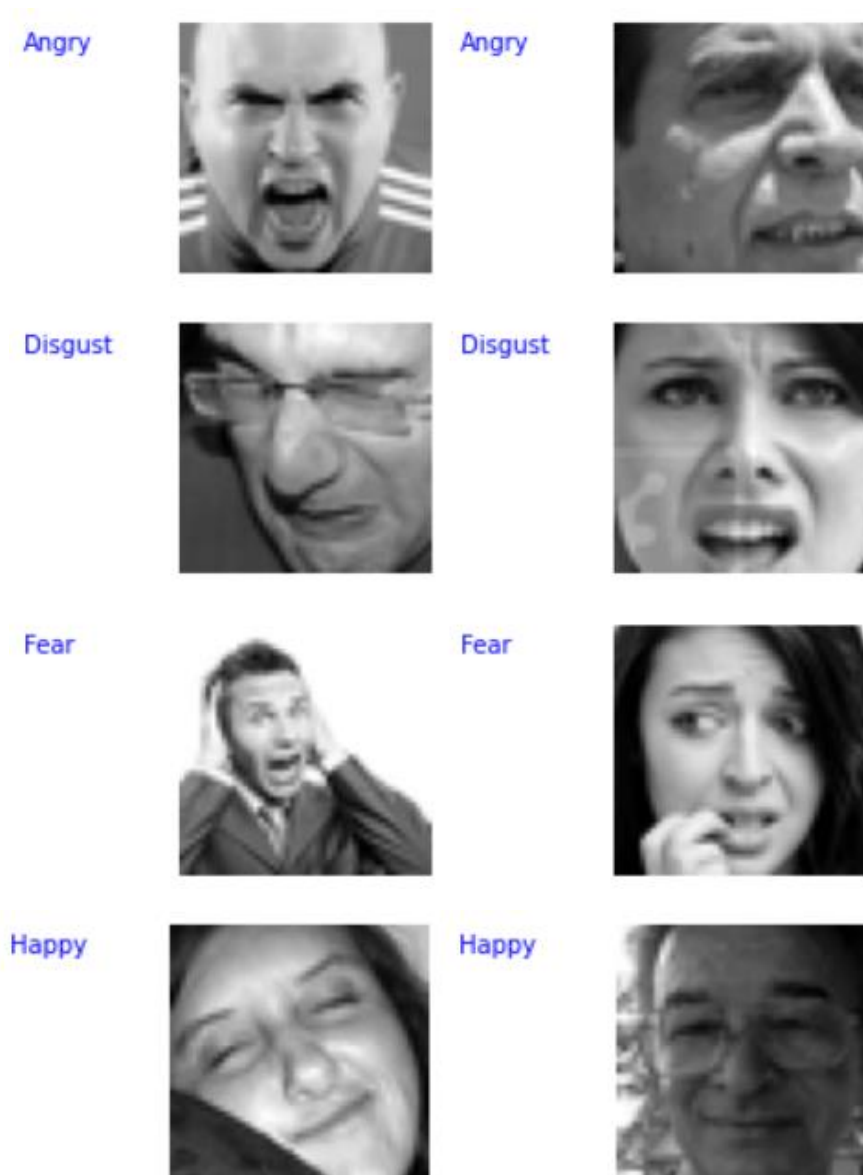


Рисунок 33 – Примеры работы нейронной сети

Выводы по главе 3

Приведем выводы по третьей главе бакалаврской работы:

– на языке программирования python разработано приложение, основанное на работе нейронной сети и позволяющее на основе анализа изображения лица определять один из 7 типов эмоций испытываемых человеком.

Заключение

В ходе выполнения бакалаврской работы были получены следующие результаты:

– на основе анализа литературных источников установлено, что большинство алгоритмов распознавания эмоций основано на совместной работе двух методов: метода поиска ключевых точек лица на изображении (Point Distribution Model, Facial Landmark Detection, Active Appearance Model и 2D-and-3D-face-alignment) и метода классификации данных (нейронные сети, деревья принятия решений, классификатор SVM);

– в работе предложено связку метода поиска ключевых точек лица и метода классификации можно заменить на сверточную нейронную сеть, что позволит облегчить настройку классификатора эмоций за счет использования стандартных алгоритмов обучения нейронных сетей;

– спроектирована максимально простая архитектура нейронной сети для решения задачи распознавания эмоций, которая включает в себя входной слой для получения изображения в виде числового массива $48 \times 48 \times 1$; сверточный слой (convolution) с ядром 5×5 , шагом свертки 2, количеством плоскостей (каналов) 32 и функцией активации ReLU; слой субдискретизации (max-pooling) с группой уплотнения 2×2 ; слой dropout, блокирующий сигналы 25% нейронов предыдущего слоя; сверточный слой (convolution) с ядром 3×3 , количеством плоскостей (каналов) 64 и функцией активации ReLU; слой субдискретизации (max-pooling) с группой уплотнения 2×2 ; слой dropout, блокирующий сигналы 25% нейронов предыдущего слоя; слой снижения размерности (flatten) данных до одномерного вектора; слой (core layer ReLU) из 500 нейронов прямого распространения с функцией активацией ReLU и выходной (core layer SoftMax) слой с функцией активацией SoftMax.

– Увеличение точности работы нейронной сети с 66,4% до 78,1%

удалось достигнуть за счет добавления слоев dropout, функционирующих только в процессе настройки сети и ослабляющих эффект переобучения сети за счет случайного блокирования некоторой части сигналов предыдущего слоя. Также точности сети была увеличена за счет синтезирования дополнительных изображений в самый малочисленный класс (класса эмоций «Disgust» / отвращение) изображений обучающей выборки. Синтез дополнительных изображений осуществлялся за счет применения со случайными параметрами операций масштабирования и вращения к исходным изображениям, содержащимся в классе.

– на языке программирования python разработано приложение, основанное на работе нейронной сети и позволяющего на основе анализа изображения лица определять один из 7 типов эмоция испытываемых человеком.

Список используемой литературы и используемых источников

1. Визильтер Ю. В. Обработка и анализ изображений в задачах машинного зрения: Курс лекций и практических занятий. – М.: Физматкнига, 2010. – 672 с.
2. Галушкин, А.И. Нейронные сети: основы теории. / А.И. Галушкин. - М.: РиС, 2015. - 496 с.
3. Гелиг, А. Х. Введение в математическую теорию обучаемых распознающих систем и нейронных сетей. Учебное пособие / А.Х. Гелиг, А.С. Матвеев. - М.: Издательство СПбГУ, 2014. - 224 с.
4. Головкин В.А. От многослойных персептронов к нейронным сетям глубокого доверия: парадигмы обучения и применение // XVII Всероссийская научно-техническая конференция с международным участием, 2015. – С. 47-84.
5. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – Москва: Техносфера, 2012. – 1104 с.
6. Каллан, Р. Нейронные сети: Краткий справочник / Р. Каллан. - М.: Вильямс И.Д., 2017. - 288 с.
7. Клейнберг Дж., Тардос Е. Алгоритмы: разработка и применение. Классика Computers Science / Пер. с англ. Е. Матвеева. – СПб.: Питер, 2016. – 800 с.
8. Красильников Н. Цифровая обработка 2D- и 3D-изображений / Н. Красильников: Отдельное издание. - БХВ-Петербург, 2011. - 608 с.
9. Редько, В.Г. Эволюция, нейронные сети, интеллект: Модели и концепции эволюционной кибернетики / В.Г. Редько. - М.: Ленанд, 2019. - 224 с.
10. Beyerer J. Automated Visual Inspection: Theory, Practice and Applications / J. Beyerer, P. Fernando, F. Christian. – Springer Berlin Heidelberg, 2016. – 798 p.

11. Calonder M. BRIEF: binary robust independent elementary features / M. Calonder, V. Lepetit, C. Strecha, P. Fua // European Conference on Computer Vision, 2010. – P. 778-792.
12. Chollet F. Deep Learning with Python – Manning Publications, 2017. – 384 p.
13. Geron Au. Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems Second Edition (Third Release) – O'Reilly Media, 2019. – 856 p.
14. Lubanovic B. Introducing Python: Modern Computing in Simple Packages Second Edition – O'Reilly Media, 2019. – 605 p.
15. Lutz M. Python Pocket Reference, 5th Edition – O'Reilly Media, 2014. – 264 p.
16. McKinney W. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, 2nd Edition – O'Reilly Media, 2017. – 544 p.
17. Muller A.C. Introduction to Machine Learning with Python: A Guide for Data Scientists / A.C. Muller, S. Guido – O'Reilly Media, 2016. – 392 p.
18. Singh P. Learn TensorFlow 2.0: Implement Machine Learning and Deep Learning Models with Python / P. Singh, A. Manure – Apress, 2019. – 177 p.
19. Sutskever I. On the importance of initialization and momentum in deep learning / I. Sutskever, J. Martens, G. Dahl, G. Hinton // Journal of Machine Learning Research. – 2013. – V. 28, No. 3. – P. 1139–1147.
20. VanderPlas J. Python Data Science Handbook: Essential Tools for Working with Data – O'Reilly Media, 2016. – 672 p.