

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)

01.04.02 Прикладная математика и информатика
(код и наименование направления подготовки)

Математическое моделирование
(направленность (профиль))

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)

на тему «Моделирование алгоритма для распознавания эмоций на изображении и видеопотоке»

Студент

А.В. Рытов

(И.О. Фамилия)

(личная подпись)

Научный
руководитель

к.т.н., доцент, В.С. Климов

(ученая степень, звание, И.О. Фамилия)

Тольятти 2022

Содержание

Введение.....	3
1 Исследование области распознавания эмоций.....	7
1.1 Введение в задачу классификации эмоций	7
1.2 Обзор существующих программных решений	11
1.3 Выбор алгоритма распознавания эмоций	17
1.4 Описание используемых моделей алгоритмов классификации	33
2 Проектирование моделей алгоритма для распознавания эмоций.....	51
2.1 Проектирование модели алгоритма распознавания лиц	51
2.2 Проектирование модели алгоритма классификации эмоций	58
3 Программные решения для реализации и сравнительный анализ представленных моделей.....	60
3.1 Программные решения, используемые для реализации спроектированных моделей	60
3.2 Сравнительный анализ реализованных моделей алгоритмов классификации эмоций.....	63
Заключение	67
Список используемой литературы	68

Введение

В настоящее время, когда человечество переступило порог двадцатого века и вступило в третье тысячелетие нашей эры, информационные технологии стали неотъемлемой частью человеческой жизни. Их грамотное применение на производственном поприще является целью любого профессионала. ЮНЕСКО определяет информационные технологии как комплекс взаимосвязанных технологических, научных, инженерных дисциплин, которые направлены на хранение, обработку и использование информации. ИТ включают в себя множество разнообразных областей развития. Например, интеллектуальные системы, когнитивные ИТ, технологии баз данных, компьютерная графика, управленческие информационные системы, а также многие другие. Одной из таких областей является компьютерное зрение.

Актуальность настоящего исследования определена потребностью в алгоритмах распознавания объектов, которые сохраняют высокую точность работы независимо от того, как изменяется положение объекта в пространстве относительно объектива камеры. В области распознавания эмоций ощущается недостаток подобных алгоритмов, поэтому на данный момент невозможно с высокой точностью классифицировать эмоции на лицах отличных от фронтального положения более чем на 45 градусов. Наличие подобных алгоритмов распознавания позволило бы значительно автоматизировать социальные исследования, а также исследования в области психологии.

Объектом исследования является алгоритм распознавания эмоций.

Предметом исследования является повышение точности алгоритма распознавания эмоций для лиц отличных от фронтального положения более чем на 45 градусов.

Цель исследования: моделирование алгоритма распознавания эмоций на лицах отличных от фронтального положения более чем на 45 градусов и

проведение сравнительного анализа точности полученного алгоритма относительно аналогичных.

Гипотеза исследования состоит в том, что точность работы алгоритма распознавания эмоций на лицах отличных от фронтального положения более чем на 45 градусов можно повысить, если:

- увеличить область для классификации передаваемую из алгоритма распознавания лиц в алгоритм классификации эмоций, для увеличения полезной информации для классификации;
- добавить в свёрточные слои алгоритма классификации эмоций пакетную нормализацию.

Для достижения поставленной цели необходимо решить следующие задачи:

- проанализировать предметную область;
- выбрать модель алгоритма для распознавания эмоций точность которой является наиболее высокой из уже существующих;
- спроектировать модернизированную модель алгоритма распознавания лиц для увеличения захватываемой им области лица;
- спроектировать модернизированную модель алгоритма классификации эмоций с применением пакетной нормализации;
- реализовать полученные модели;
- провести сравнительный анализ между реализованными моделями, стандартной моделью алгоритма распознавания эмоций и другими аналогами.

Методы исследования: в ходе выполнения работы применялись такие методы теоретического исследования, как изучение и анализ научной литературы по задачам распознавания объектов в машинном обучении, также в ходе выполнения работы применялись практические методы исследования, такие как компьютерное моделирование алгоритмов машинного обучения, а также их сравнительный анализ.

Научная новизна исследования заключается в:

- применении технологии DNN для алгоритма распознавания лиц, данная технология была применена не к последнему слою CNN, а к её предпоследнему слою, за счёт проведения операции обратного пулинга, что позволило увеличить размерность карт признаков, которые применялись для выделения границ обнаруженного лица, соответственно это позволило увеличить область в которой содержится лицо для алгоритма классификации эмоций;
- применении подхода пакетной нормализации в модели алгоритма классификации эмоций, что позволило не только ускорить процесс обучения, а также повысить точность её работы на 5,93%.

Теоретическая значимость исследования заключается в:

- предложен подход, для более глубокого применения технологии DNN, что позволяет расширить область изображения, в которой содержится искомый объект;
- доказано, что добавление слоёв пакетной нормализации может повысить точность работы CNN.

Практическая значимость исследования заключается в разработке модели алгоритма для классификации, которая со средней точностью в 74% способна распознавать эмоции на лицах отличных от фронтального положения более чем на 45 градусов.

На защиту выносятся:

- модель алгоритма распознавания эмоций на лицах отличных от фронтального положения более чем на 45 градусов;
- результаты сравнительного анализа алгоритмов распознавания эмоций.

В первом разделе производится исследование предметной области, даётся экскурс в задачу распознавания эмоций, рассматриваются уже существующие программные решения, описываются используемые в работе

базовые модели алгоритмов, а также производится выбор наиболее точной базовой модели, которая в дальнейшем будет использоваться для улучшения.

Во втором разделе проводится исследование с целью увеличения области выделяемой распознавателем лиц с помощью технологии DNN, также рассматривается возможность применения к свёрточным слоям классификатора эмоций пакетной нормализации для повышения точности работы.

В третьем разделе приводятся программные решения, используемые для реализации моделей полученных во втором разделе, а также производится сравнительный анализ реализованных моделей.

Работа состоит из введения, трёх разделов, заключения, содержит 23 рисунка, 34 формулы, список использованной литературы (30 источников). Основной текст работы изложен на 70 страницах.

1 Исследование области распознавания эмоций

1.1 Введение в задачу классификации эмоций

В течении жизни любой человек испытывает разнообразные эмоции, благодаря которым другие люди могут понять какие чувства этот человек испытывает в данный момент и как стоит с ним взаимодействовать для избегания конфликта. На данный момент классификация эмоций является одним из самых развивающихся направлений в области искусственного интеллекта. Программное обеспечение по классификации эмоций можно использовать в различных областях, например, можно проводить социологические исследования задавать людям какие-либо вопросы и оценивать их отношение к ним или можно использовать его в маркетинге для повышения посещаемости каких-либо развлекательных центров или магазинов. Эмоции являются очень важными элементами в жизни каждого человека. В книге, которую написали Don Hockenbury и Sandra E. Hockenbury "Открывая психологию" говорится, что эмоция – это сложное психологическое состояние, которое включает в себя три различных компонента: субъективное переживание, физиологическую реакцию и поведенческую или экспрессивную реакцию. В дополнение к попыткам определить, что такое эмоции, исследователи также пытались выявить и классифицировать различные типы эмоций. Одним из таких исследователей был американский психолог Paul Ekman в середине 1970-х годов он установил, что есть эмоции, которые не изменяются у людей разных культур и народностей. Данные эмоции были названы базовыми, всего их 7 видов.

Базовыми эмоциями являются следующие:

- печаль,
- гнев,
- отвращение,

- страх,
- удивление,
- презрение,
- радость.

Далее на рисунке 1 будет наглядно продемонстрировано, как выглядит лицо человека, когда он испытывает те или иные базовые эмоции.



Рисунок 1 – Базовые эмоции на лице человека

На рисунке, представленном выше можно видеть, как в зависимости от испытываемой эмоции изменяется лицо человека.

Современные психологи считают, что эмоция типа презрение больше выражается не через чувства, а через отношение, то есть она проявляется не с помощью мимики лица, а через действия, направленные на объект, к которому человек испытывает презрение. По этой причине эмоцию типа презрение чаще всего невозможно распознать по изображению лица человека. В связи с этим

большинство известных программных решений по распознаванию эмоций не пытаются классифицировать эмоцию типа презрение, вместо этого классифицируется отсутствие каких-либо чётко выраженных эмоций, данное состояния принято считать состоянием спокойствия или нейтральность. В данной работе в ходе моделирования алгоритмов классификации эмоций также планируется вместо эмоции типа презрение классифицировать состояние спокойствия.

Для классификации эмоций можно выделить 2 подхода: классический и с помощью «глубокого обучения».

Классический подход заключается в том, что на изображении где уже выделена область лица человека проставляют от 5 до 68 точек по методу ключевых точек, это позволяет гарантировано подхватить мимику лица человека [1]. Данные точки всегда привязываются к тем частям лица, которые могут быть использованы для классификации эмоций (губы, глаза, брови, нос). Далее массив данных точек нормализуется и подаётся на вход какому-либо алгоритму классификации, например, К случайных соседей (KNN) или метод опорных векторов (SVM). Данный подход позволяет получить базовое решение, но для повышения точности также можно вычислить дескрипторы ключевых точек, например, LATCH или HOG конкатенировать их после чего полученный вектор признаков можно применить для классификации эмоций. Использование дескрипторов повышает точность классического подхода, но по сравнению с «глубоким обучением» данный подход уступает по точности, поэтому на данный момент он используется достаточно редко.

Классификация эмоций при помощи «глубокого обучения» является наиболее современным и точным способом. Необходимо обучить нейронную сеть для распознавания изображений на ImageNet, после этого переобучить последние её слои под задачу классификации эмоций, таким образом получается модель алгоритма классификации, в которую уже можно подавать изображения для анализа. Поэтому изображение с лицом человека переводится в оттенки серого, что позволяет уменьшить признаковое

пространство, данная предобработка изображения необязательна, но благодаря ей нейронная сеть сможет обработать изображение намного быстрее, после чего изображение в оттенках серого подаётся в модель алгоритма нейронной сети и в результате классификации получается массив признаков, которые показывают насколько сильно выражен тот или иной класс эмоций на данном изображении.

Так как алгоритмы классификации являются частью алгоритмов машинного обучения то стоит отметить, что они используют обучение с учителем.

Обучение с учителем означает, что имеется полный набор размеченных данных для обучения модели на всех этапах её построения. Это позволяет заранее знать к какому ответу должна стремиться модель алгоритма классификации. Например, размеченный набор данных с изображениями котиков, позволит обучить модель алгоритма классификации распознавать любых котиков и в дальнейшем это может быть использовано для распределения животных по классам (кошки, собаки, мыши и т.д.), то есть обучение с учителем как раз отлично подходит для задач классификации. Также обучение с учителем может применяться в задачах регрессии, когда необходимо вычислить ожидаемое значение переменной y , в зависимости от конкретного значения переменной x .

Так же существуют другие способы обучения алгоритмов машинного обучения: без учителя и с частичным привлечением учителя. Так как в данной работе планируется использовать обучение моделей алгоритмов классификации с учителем, то подробнее рассматривать данные способы обучения не требуется.

Далее можно перейти к рассмотрению уже имеющихся аналогов программных решений для распознавания эмоций на изображении.

1.2 Обзор существующих программных решений

Рассмотрение существующих программных решений следует начать с приложения FaceReader от голландской компании Noldus Information Technology, так как именно этот программный продукт является на сегодняшний день наиболее точным.

FaceReader – самая надежная автоматизированная система для распознавания ряда специфических свойств изображений лица, включая шесть основных или базовых эмоций: грусть, удивление, гнев, счастье, отвращение и страх.

Кроме того, FaceReader может распознавать "нейтральное" состояние и анализировать "презрение".

Независимо от того, является ли человек младенцем, ребенком, взрослым или пожилым человеком, FaceReader подстраивает анализ под модель, которая лучше всего применима для исследования.

FaceReader работает следующим образом:

- обнаружение области лица на изображении, используя алгоритм поиска на основе «глубокого обучения»;
- создание точной искусственной модели лица с использованием почти 500 ключевых точек;
- обработка модели с помощью «глубокого обучения»;
- классифицирует выражения с помощью искусственных нейронных сетей (ANN).

На рисунке 2 представленном ниже будет продемонстрирован пример работы программного решения FaceReader.

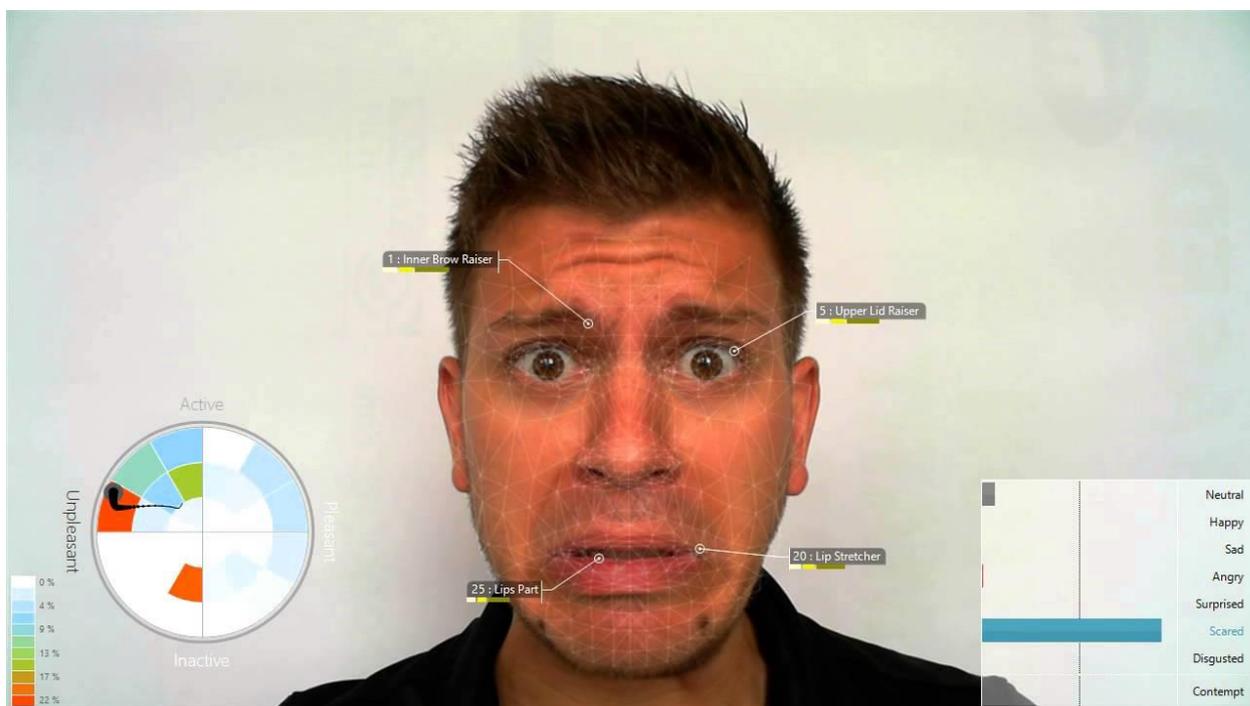


Рисунок 2 – Пример работы программного решения FaceReader

На рисунке 2 можно видеть, как программное решение FaceReader классифицирует эмоции. Кроме простой классификации эмоций оно также выделяет области лица исходя из которых была выявлена преобладающая эмоция, накладывает 3D-сетку на лицо и выполняет некоторые другие действия.

Преимуществами данного приложения являются:

- средний процент распознавания эмоций равен 99, если лицо человека расположено фронтально или под не большим углом к камере;
- поддержка множества современных кодеков видеофайлов таких как MPEG1, MPEG2, DV-AVI и других, к тому же приложение может обрабатывать видеопоток, поступающий с веб-камеры пользователя, а также

данное программное решение способно обрабатывать отдельные изображения;

- поддержка многообразных возможностей визуализации, например, построение гистограмм, диаграмм, отрисовка сетки и разметка областей особого внимания на лице.

Недостатком данного программного решения является то, что в случае значительного отклонения лица от фронтального (45 градусов и больше) оно некорректно определяет эмоции.

Следующим приложением, которое также является аналогом разрабатываемого алгоритма, является Хеота, а именно модуль Детектор лиц (Эмоции).

Хеота – это программное решение для видеонаблюдения с простым и удобным интерфейсом, гибкими настройками, качественным сервисом и профессиональными функциями, с помощью неё можно получить доступ к видеокамерам абсолютно из любой точки мира. Разработчиком данной программы является российская компания ФеленаСофт. Данное приложение является лидером продаж среди программ, используемых для видеонаблюдения. Хеота – единственная программа для видеонаблюдения на Linux ARM и Android с подобным обширным функционалом. Программа для видеонаблюдения №1 для Mac OS X. Также данное приложение, входит в пятерку лучших программ для видеонаблюдения на Windows и Linux.

Начиная с версии 18.11.21 в данную программу входит программный модуль Детектор лиц (Эмоции). Он может анализировать изображения, находить на них лица и по ним распознавать 6 базовых эмоций: счастье, удивление, раздражение (гнев), отвращение, испуг (страх) и печаль. Также данное приложение может распознавать состояние – нейтральность (спокойствие). Все базовые эмоции и состояние нейтральности (спокойствия) присутствуют на лице в той или иной степени, в зависимости от чего им присваивается разный процент (от 0 до 100). Тот тип эмоции, который получит наибольший процент по результатам классификации будет принят за общее

состояние и отображён в пункте «Настроение». Также на экране будут представлены и все остальные результаты классификации, что позволит визуально определить её точность.

Преимуществами данного приложения являются:

- удобный и достаточно простой графический интерфейс позволяет достаточно быстро выполнить необходимые задачи;

- гибкие настройки классификатора эмоций, которые позволяют для каждого отдельного изображения или видеопотока задать уникальные особенности, например, указать дальность обнаружения лиц, указать процентное соотношение эмоций на которое следует реагировать, а также многие другие;

- возможность распознавать эмоций с помощью пользовательской веб-камеры и на видеопотоке из видеофайла, так как данная программа также, как и FaceReader поддерживает множество кодеков, а также возможность загружать обычные изображения или видеопоток с удалённой веб-камеры;

- возможность распознавать эмоции сразу нескольких людей на изображении.

Недостатками данного программного решения являются:

- присутствует не точное распознавания эмоций, иногда распознаются эмоций, которые человек не испытывает;

- в случае значительного отклонения лица от фронтального (45 градусов и больше) некорректно классифицируются эмоции.

На рисунке 3 представленном ниже будет приведён результат классификации эмоций, где представлены лица двух людей.



Рисунок 3 – Пример классификации эмоций на лицах в Хеота

На рисунке, расположенном выше, можно видеть, что модуль Детектор лиц (Эмоции) приложения Хеота смог классифицировать эмоции сразу нескольких человек, что является его значительным преимуществом перед другими аналогичными программными решениями, которые могут классифицировать эмоции только одного человека.

Последним аналогом, который хотелось бы рассмотреть в ходе выполнения данной работы, является EmoDetect.

EmoDetect – программа, разработанная российской компанией Нейроботикс, для регистрации (записи) лица человека и определения по нему эмоционального состояния. Классификатор программы EmoDetect распознаёт до 20 информативных локальных признаков лица, характеризующих

психоэмоциональное состояние человека (ASM). Данное приложение, как и его аналоги, может распознавать 6 базовых эмоций: радость (счастье), удивление, гнев, отвращение, страх и грусть (печаль), а также состояние нейтральность (спокойствие).

Также стоит отметить, что компания Нейроботикс начала свою деятельность в 2004 году в сфере разработки и производства оборудования для исследований в нейрофизиологии и психофизиологии. С тех пор Нейроботикс активно развивается и взаимодействует с различными международными компаниями. Выше сказанное говорит о том, что у Нейроботикс имеется достаточное количество опыта в области компьютерного зрения для разработки алгоритма распознавания человеческих эмоций на изображении с хорошей точностью.

Преимуществами данного приложения являются:

- возможность классификации эмоций 3 разнообразными классификаторами такими как нейронная сеть, система решающих правил и по взвешенной сумме признаков;
- возможность алгоритма адаптироваться под каждого человека индивидуально;
- возможность построения гистограмм и диаграмм для того чтобы более наглядно продемонстрировать как изменялось эмоциональное состояние человека с течением времени видеопотока;
- возможность распознавания эмоций при условии, что человек находится в очках;
- предоставление программного интерфейса приложения для того чтобы подключить модули распознавания EmoDetect к другой программе.

Недостатком данного приложения является отсутствие возможности распознавания лица, повернутого в профиль.

На рисунке 4 который представлен ниже будет продемонстрирован пример работы приложения по распознаванию эмоций EmoDetect.

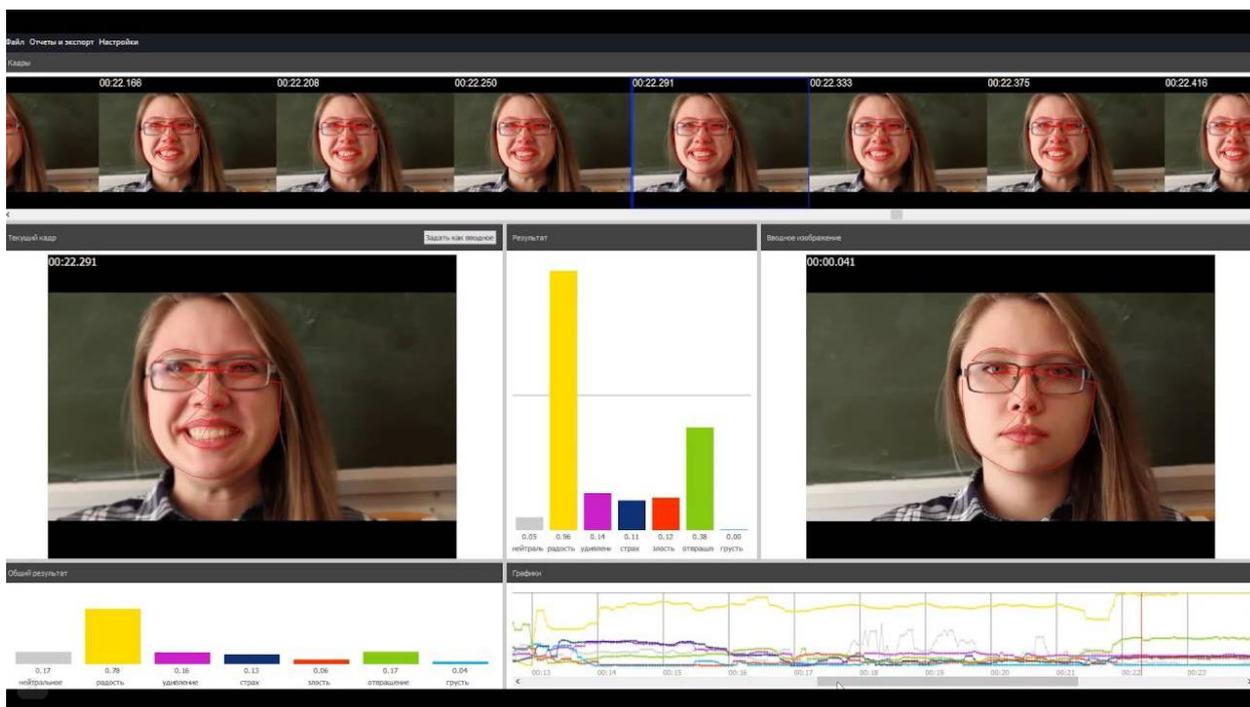


Рисунок 4 – Пример работы приложения EmoDetect

На рисунке 4 можно видеть, как приложение EmoDetect распознаёт эмоции на видеопотоке. Стоит отметить, что диаграммы эмоционального состояния человека создаются и изменяются в режиме реального времени. Также программа, как и FaceReader, размечает на лице человека ключевые точки исходя из которых распознаётся преобладающая эмоция.

1.3 Выбор алгоритма распознавания эмоций

За основу в данной работе планируется взять уже существующий алгоритм распознавания эмоций и путём его модернизации добиться увеличения точности, но для этого необходимо чтобы изначально алгоритм работал с высокой точностью для распознавания эмоций на лицах отличных от фронтального положения менее чем на 45 градусов.

Для распознавания эмоций планируется применять два классификатора, один из которых определяет есть ли на изображении лицо, если есть он

выделяет область, в которой было обнаружено лицо и подаёт эту область изображения в классификатор эмоций, который уже определяет какие эмоции присутствуют на лице человека. Данный подход повышает точность классификации, поскольку отбрасывает области изображения, не несущие полезной информации для работы классификатора эмоций.

В данной работе для классификации лиц будет использоваться свёрточная нейронная сеть (CNN).

Свёрточная нейронная сеть (CNN) – специальная архитектура искусственных нейронных сетей (ANN), которую предложил в 1988 году французский учёный Yann LeCun, предназначенная для эффективного распознавания образов, входит в состав технологии «глубокого обучения». Данный вид нейронных сетей был разработан опираясь на особенности зрительной коры головного мозга, в ходе её исследования было установлено что нейроны по-разному реагируют на воспринимаемый объект, так некоторые нейроны реагируют на горизонтальные очертания объекта, другие реагируют на вертикальные и так далее. Как любая другая нейронная сеть CNN состоит из нейронных слоёв. Свёрточная нейронная сеть (CNN) может содержать в себе следующие слои:

- свёрточный (convolution),
- подвыборки (pooling),
- активационный (flatten),
- полносвязный (fully connected).

Далее будут представлены термины, применяемы при описании алгоритма свёрточной нейронной сети (CNN).

Тензор – это 3D массив чисел или говоря другими словами массив матриц чисел. Как известно большинство цифровых изображений хранится в цветовом пространстве RGB, что подразумевает под собой для одного пикселя три числовых компоненты отвечающих за интенсивности красного, зелёного

и синего цветов соответственно. Говоря другими словами все изображения цветового пространства RGB являются тензорами.

Признак – это тензор, который на первых слоях CNN представляет собой какой-либо примитив (горизонтальная линия, кривая линия и т.д.) необходимый для классификации. На более поздних нейронных слоях признаки могут представлять более сложные объекты (уши или глаза).

Преимуществами алгоритма классификации свёрточной нейронной сети являются:

- за счёт использования ядер свёртки предполагает меньшее количество необходимых для настройки весовых коэффициентов, по сравнению с полносвязанной нейронной сетью, это приводит к повышению скорости обучения и снижает его ресурсозатратность;
- устойчивость к повороту и сдвигу;
- при качественном обучении высокая точность классификации;
- за счёт слоёв субдискретизации частично инвариантен к масштабу.

Недостатками алгоритма классификации свёрточной нейронной сети являются:

- значительное время обучения, если число свёрточных слоёв больше двух (несколько дней и более);
- необходимость в большом количестве примеров для обучения;
- большая вероятность полного переобучения в случае недостатка данных при обучении с учителем.

В качестве классификатора лиц была выбрана свёрточная нейронная сеть потому что точность её работы для распознавания лиц по многочисленным исследованиям является наилучшей, она позволяет распознавать лица даже если они не полностью присутствуют на изображении или находятся под углом относительно объектива камеры, также алгоритм CNN для классификации лиц устойчив к такому явлению, как окклюзия.

Окклюзия – это ситуации при, которой между камерой и наблюдаемым объектом появляется какой-либо предмет, который частично или полностью скрывает объект от камеры. В случае с человеческим лицом к окклюзии может приводить наличие бороды, усов, очков и т.д.

Для того чтобы можно было выделить область, в которой расположено лицо, к алгоритму классификации CNN будет применена технология обратной свёрточной нейронной сети (DNN). Данная технология предназначена для построения иерархических представлений изображения, которые потом можно использовать для задач первичной обработки сигналов или для шумоподавления и повышения точности работы CNN, но в данной работе применение этой технологии планируется для выделения границ признаков на изображении, по которым производилась классификация.

В качестве алгоритмов классификации эмоции можно выбрать один из следующих наиболее известных классификаторов:

- К ближайших соседей (K-Nearest Neighbors);
- метод опорных векторов (Support Vector Machines);
- наивный байесовский метод (Naive Bayes);
- свёрточная нейронная сеть (Convolutional Neural Network).

Далее будет приведено описание для каждого указанного алгоритма классификации без ориентации на классификацию эмоций.

KNN – алгоритм классификации К ближайших соседей является одним из типов алгоритмов машинного обучения с учителем. KNN чрезвычайно прост в реализации, но при этом выполняет довольно сложные задачи классификации. Это алгоритм ленивого обучения, ему не требуется специальная фаза обучения, все необходимые данные для обучения обрабатываются и используются во время классификации на реальных данных. Данный алгоритм является непараметрическим это означает, что набор данных подаваемых ему на вход не должен удовлетворять определённым предположениям или параметрам [7]-[8]. Как правило такие

алгоритмы можно применять к большому количеству задач так как реальные задачи бывает сложно описать предположениями, например, линейная делимость, равномерное распределение и т.д.

Алгоритм KNN является одним из самых простых среди всех алгоритмов машинного обучения с учителем [10]. Он вычисляет расстояние новой точки данных до всех других точек данных с уже известными классами. Метрика может использоваться любая, например, евклидова, манхэттенская и т.д. Затем выбирается K ближайших точек данных, где K может быть любым целым числом. После этого алгоритм относит точку данных к тому классу, к которому принадлежит большинство из K точек данных. Процесс работы алгоритма можно представить в виде последовательности следующих шагов:

- определение числа K ;
- выбор используемой метрики (евклидова, манхэттенская и т.д.);
- загрузка данных из выборки, в которой требуется какой-либо точке проставить метку класса;
- нормализация данных;
- вычисление расстояния с учётом выбранной метрики для каждой точки в наборе данных относительно точки, для которой производится классификация;
- назначение класса точки, для которой производится классификация, в зависимости от классов K ближайших точек.

Про некоторые шаги необходимо рассказать по подробнее.

В KNN наиболее распространённой метрикой является евклидова. Евклидова метрика, также известна как евклидово расстояние, метрика в евклидовом пространстве, которая предполагает вычисление расстояния между двумя точками евклидова пространства по теореме Пифагора. Иными словами, евклидово расстояние между двумя точками это наименьшее расстояние между ними. Евклидово расстояние эффективно для малых измерений, оно не подходит для больших измерений, а также для

категориальных переменных. Причина тому существенный недостаток евклидова расстояния, оно игнорирует сходства между атрибутами, они рассматриваются как несвязанные, полностью отличные друг от друга [11]-[14].

Также существуют другие менее используемые в KNN метрики манхэттенская и расстояние Чебышёва.

Манхэттенская метрика, также известна как расстояние городских кварталов или L1 расстояние, это расстояние, измеряемое между двумя точками, равное сумме длин проекций отрезка прямой между точками на координатные оси. В данной метрике расстояние между точками получается не минимальным, так как положение отрезков может быть только горизонтальным или вертикальным, диагональное является недопустимым [15]-[17].

Расстояние Чебышева, известное также как метрика шахматной доски, равняется значению наибольшего модуля разности между свойствами признаков. Данная метрика как уже было сказано ранее имеет ещё одно название метрика шахматной доски потому что точку можно сравнить с королём в шахматах, который может сделать ход на одну клетку в любом направлении по вертикали диагонали и горизонтали.

Далее ещё одним шагом в алгоритме KNN требующим более подробного описания является шаг нормализации данных. У различных атрибутов из выборки обычно бывают отличные друг от друга диапазоны принимаемых значений (например, атрибут А принимает значения в диапазоне от 0 до 1, а атрибут Б принимает значения в диапазоне от 100 до 500). Это приводит к тому что атрибуты не могут быть сравнимы в исходном виде, а также значения расстояний будут зависимы от атрибутов с большими значениями. Для избегания описанных ранее ситуаций данные необходимо нормализовать. При кластерном анализе основными способами нормализации данных являются: MinMax-нормализация и Z-нормализация.

Далее будет представлен пример работы алгоритма классификации KNN. Дан набор данных с двумя метками классов, задача алгоритма, классифицировать новую точку данных, которая обозначена символом «X». В приведённом примере координаты данной точки будут равны: $x=45$, $y=50$, а число соседей $K=3$. Алгоритм KNN производит вычисления, начиная с расчёта расстояний от точки X до всех остальных точек. Затем выбираются три точки, у которых расстояние до точки X является минимальным. Данный процесс представлен на рисунке 5, расположенном ниже, три ближайшие точки, а также точка X обведены кружком.

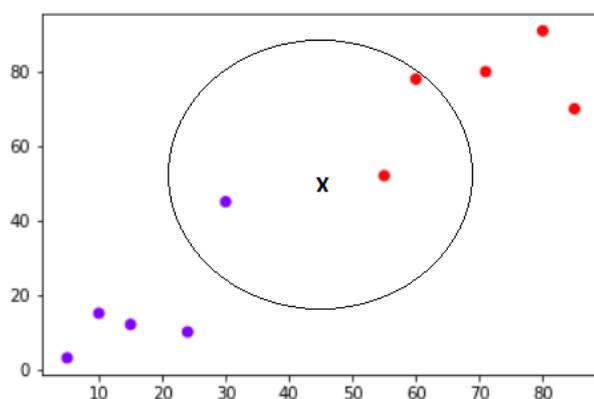


Рисунок 5 – Пример работы алгоритма классификации KNN

Последним шагом алгоритма KNN является присвоение метки класса точке X, в зависимости от классов три ближайших точек. Как можно видеть из рисунка 5, одна из ближайших точек принадлежит к классу «Синие», а две других принадлежат к классу «Красные», поэтому точка X будет отнесена к классу «Красные».

Преимуществами алгоритма классификации KNN являются:

- простота реализации;
- как было сказано ранее, это ленивый алгоритм обучения, поэтому его не требуется дополнительно обучать на тестовых данных перед тем, как производить классификацию в реальном времени;

- добавление новых данных (классов или атрибутов) может быть выполнено без затруднений так как алгоритм не требуется переобучать;
- в реализации KNN требуется всего два параметра: значение K ближайших соседей и функция вычисления расстояния по выбранной метрике.

Недостатками алгоритма классификации KNN являются:

- плохо работает с данными высокой размерности, поскольку при большом количестве измерений алгоритму придётся производить вычисления для каждого из них;
- имеет высокую стоимость классификации для больших наборов данных, это связано с тем, что в больших наборах данных стоимость вычисления расстояния между новой точкой и каждой существующей точкой возрастает.

SVM – алгоритм классификации метод опорных векторов, также как и KNN имеет тип обучения с учителем. SVM появился в 1960-х годах и был усовершенствован в 1990-х годах. SVM является алгоритмом у которого присутствует фаза обучения. Суть его работы сводится к следующему, все данные представляются в n -мерной плоскости, где n – количество признаков и требуется построить гиперплоскость, которая сможет разделить данные таким образом, чтобы с одной стороны были данные одного класса, а с другой стороны данные другого класса. Таким образом точка, которую необходимо классифицировать, в зависимости от её положения в n -мерном пространстве относительно гиперплоскости может быть легко отнесена к тому или иному классу. В ходе фазы обучения в алгоритм поступают уже классифицированные данные, его задача на данном этапе научиться строить гиперплоскость, которая будет максимально отдалена ото всех точек на плоскости, но в тоже время сможет их классифицировать. Также выделяются случаи линейно разделимости данных и линейной неразделимости данных, для них

различаются способы построения гиперплоскости и её вид, позднее про каждый случай будет рассказано подробно. Независимо от этих случаев шаги работы алгоритма классификации SVM, после его обучения можно представить следующим образом:

- загрузка реальных данных, загрузка модели алгоритма классификации SVM уже обученной на данных из обучающей выборки;
- нормализация данных;
- построение гиперплоскости с помощью загруженной модели алгоритма SVM для реальных данных;
- классификация искомой точки при помощи построенной гиперплоскости.

Существуют различные подходы для построения гиперплоскости, не смотря на то что между ними существуют различия все их принято называть SVM, они приведены ниже:

- классификатор максимального запаса (maximum margin classifier) используется в самом простом случае, когда данные линейно разделимы;
- опорный векторный классификатор (support vector classifier), этот подход подразумевает допущение некоторых неверных классификаций, для борьбы с шумом, его используют также для линейно разделимых данных;
- опорная векторная машина (support vector machine) используется, когда данные линейно неразделимы и требуется построить более сложную гиперплоскость.

Классификатор максимального запаса является наиболее простым из приведённых выше, как уже было сказано он применим, когда данные линейно разделимы, это означает, что гиперплоскость будет построена с помощью линейной функции.

На рисунке 6 представленном ниже будет приведён пример работы классификатора максимального запаса для построения гиперплоскости в алгоритме SVM.

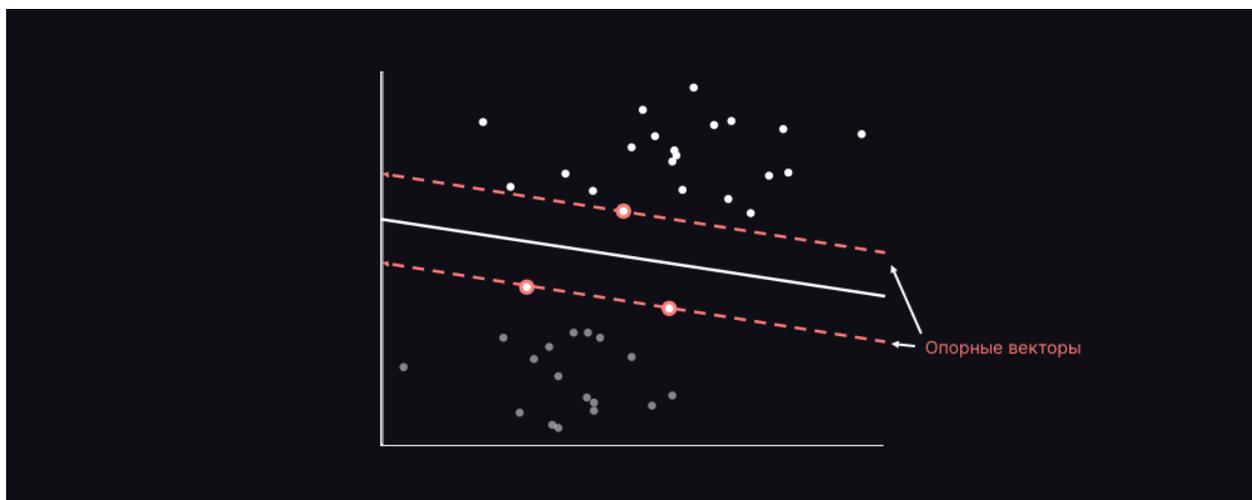


Рисунок 6 – Пример работы классификатора максимального запаса

На рисунке, представленном выше, можно видеть два класса данных «Светлые» и «Тёмные» точки. Красными отмечены точки, которые являются опорными для построения гиперплоскости, данные точки расположены ближе всего к гиперплоскости. Оптимальной гиперплоскостью будет такая для, которой расстояние от опорных векторов до самой гиперплоскости будет максимальным. Именно такую гиперплоскость и пытается построить классификатор максимального запаса, данное утверждение также верно и для всех классификаторов, которые применяются в алгоритме SVM. Также про опорные векторы можно сказать, что если убрать значения, соответствующие им, из набора данных то гиперплоскость измениться, а если же убрать все значения кроме тех которые равны значениям точек опорных векторов, то гиперплоскость не измениться. Возвращаясь к рисунку 6 можно видеть, что оптимальная гиперплоскость была построена, все «Светлые» и «Тёмные» точки лежат по разные её стороны [19]-[21].

Но бывают случаи, когда невозможно построить оптимальную гиперплоскость с помощью линейной функции. Если этому мешают один или два примера, то можно предположить, что они являются отклонениями от нормы и их называют шумом, значениями, вносящими неточности в набор данных и затрудняющими его классификацию. Из-за этого, такой набор разделить гиперплоскостью, полученной при помощи линейной функции невозможно, поэтому классификатор максимального запаса неприменим к данному набору. Поэтому к такому набору данных может быть применён опорный векторный классификатор или опорная векторная машина. Первый будет пытаться построить гиперплоскость с помощью линейной функции, при этом не обращая внимания на шум (то есть при классификации вероятно будут допустимы не точности), а второй построит гиперплоскость в основе, которой не будет лежать линейная функция.

При помощи опорного векторного классификатора гиперплоскость можно построить, но не все данные из обучающего набора будут классифицированы правильно, но для данного классификатора это является допустимым. Благодаря этому точки, которые могли внести погрешности в вычисления функции для построения гиперплоскости не учитываются.

До этого были рассмотрены примеры линейно разделимых данных с шумом и без него, в них гиперплоскость удавалось построить при помощи линейных функций, но чаще всего реальные данные бывают линейно неразделимыми и для их классификации применяется опорная векторная машина. В данном случае будут представлены данные двух различных классов, которые невозможно классифицировать при помощи гиперплоскости полученной при помощи линейной функции. В данном случае пространство является двумерным, но если добавить ещё одно измерение, то данная задача может быть легко решена. Вводится дополнительная ось z , которая равна $z=y^2+x^2$, на осях x и z данные будут располагаться по-другому.

Таким образом ранее линейно неразделимые данные в двумерном пространстве стали легко разделимы в трёхмерном пространстве. В результате

гиперплоскость, которая будет построена классификатором опорная векторная машина, будет выглядеть так как показано на рисунке 7 изображённом ниже.

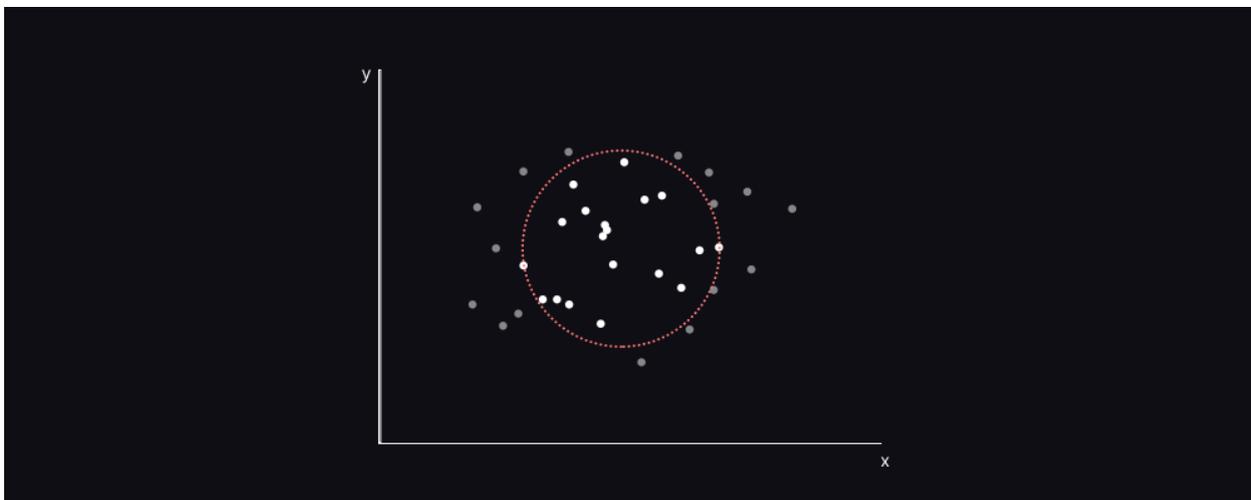


Рисунок 7 – Пример работы опорной векторной машины

Как можно видеть из рисунка 7 благодаря переходу в более высоко признаковое пространство, линейно неразделимые данные удалось разделить с помощью гиперплоскости в основе, которой лежит уравнение окружности. В опорной векторной машине могут применяться любые нелинейные функции, что позволяет ей классифицировать с высокой точностью любые наборы данных.

Преимуществами алгоритма классификации SVM являются:

- имеет высокую точность при четкой границе разделения классов;
- эффективен на больших размерностях;
- эффективен в тех случаях, когда количество признаков превышает количество наблюдений.

Недостатками алгоритма классификации SVM являются:

- при работе с большим набором данных, обучение модели алгоритма может занимать значительное количество времени;

- сильно подвержен влиянию шума, а также влиянию перекрёстных данных из разных классов.

Наивный байесовский метод – статистический алгоритм классификации, который основывается на использовании теоремы Байеса. Эта теорема является основой дедуктивного рассуждения, которое фокусируется на определении вероятности наступления события на основе предварительных знаний об условиях, которые могут быть связаны с этим событием. Изначально имеется набора данных, в котором указывается список примеров, для каждого примера указаны характеристики и то к какому классу этот пример относится. На основании данных для уже известных примеров требуется классифицировать новый пример с характеристиками отличными от табличных. Также в данном алгоритме принимаются следующие допущения:

- характеристики одного конкретного примера являются независимыми друг от друга;
- характеристики одного конкретного примера являются равнозначными по отношению друг к другу.

Стоит отметить, что в реальных условиях почти никогда не бывает того, что характеристики одного и того же примера друг от друга не зависимы. Именно поэтому данный алгоритм классификации называется наивным. В общем случае работа наивного байесовского метода сводится к выполнению следующих шагов:

- получение статистических данных;
- составление частотных таблиц;
- составление таблиц правдоподобия, на основе частотных таблиц;
- применение теоремы Байеса, значения в которую подставляются из таблиц правдоподобия.

В наивном байесовском метод могут применяться следующие типы классификаторов:

- мультиномиальный, в данном случае векторы признаков состоят из значений частотности, то есть каждый признак представлен частотой события, которая генерируется посредством мультиномиального распределения;
- Бернулли, в данном случае в многомерной модели событий Бернулли признаки представлены бинарными значениями (двоичными переменными), описывающим входные данные;
- Гаусса, в данном случае предполагается, что непрерывные значения всех характеристик имеют распределение Гаусса или нормальное распределение.

Преимуществами наивного байесовского метода являются:

- алгоритм работает сразу с реальными данными и не требует обучения;
- большое количество классов в наборе данных не сильно усложняет вычисления;
- для эффективного начала классификации не требует большого количества статистических данных.

Недостатками наивного байесовского метода являются:

- если для какого-либо признака отсутствует значение в статистическом наборе данных, то вероятность того что данный пример будет отнесён к какому-либо классу равна 0 (не актуально для числовых признаков);
- значения вычисленных вероятностей, не всегда являются достаточно точными;
- предположение о независимости признаков, которое для большинства реальных данных является ошибочным.

Для выбора наиболее эффективной в плане точности связки распознавателя лиц DNN с классификатором эмоций из представленных ранее был произведён сравнительный анализ точности этих классификаторов для задачи классификации эмоций без учёта окклюзии и с её учётом при этом их отклонение от фронтального положения не превышало 15 градусов.

На рисунке 8 представленном ниже можно будет видеть результаты замера точности алгоритмов классификации эмоций без учёта окклюзии.

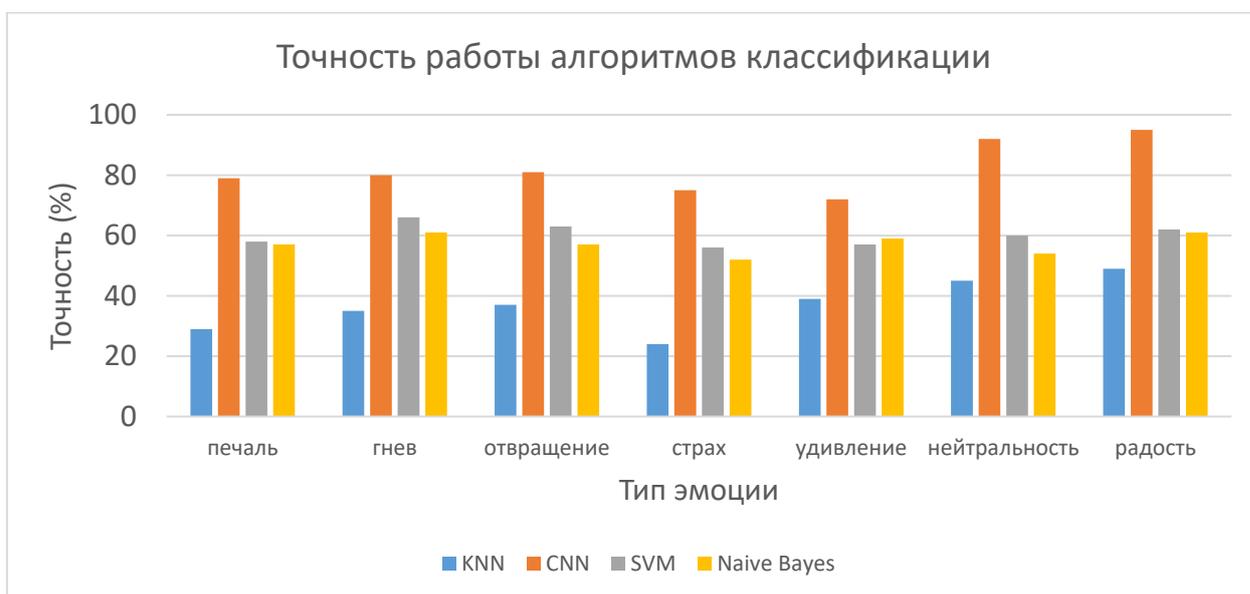


Рисунок 8 – Гистограмма с результатами замера точности работы классификаторов без учёта окклюзии

На рисунке 8 можно видеть результаты замера точности алгоритмов классификации эмоций. Тестирование проводилось при помощи набора данных AffectNet из которого было выбрано по 500 изображений для каждого класса, точность рассчитывалась как отношение верно классифицированных эмоций для каждого класса к общему количеству изображений, представленных в тесте для каждого класса. Из рисунка 11 можно видеть, что алгоритм классификации CNN показал наиболее высокие показатели точности в среднем 82%, следующим по средней точности идёт алгоритм классификации SVM с 60%, за ним байесовский наивный метод 57% и наименее точным является алгоритм классификации KNN с средней точностью 36%.

Далее на рисунке 9 будут представлены результаты замера точности работы алгоритмов классификации эмоций с учётом окклюзии значение, которой не превышает 15%.

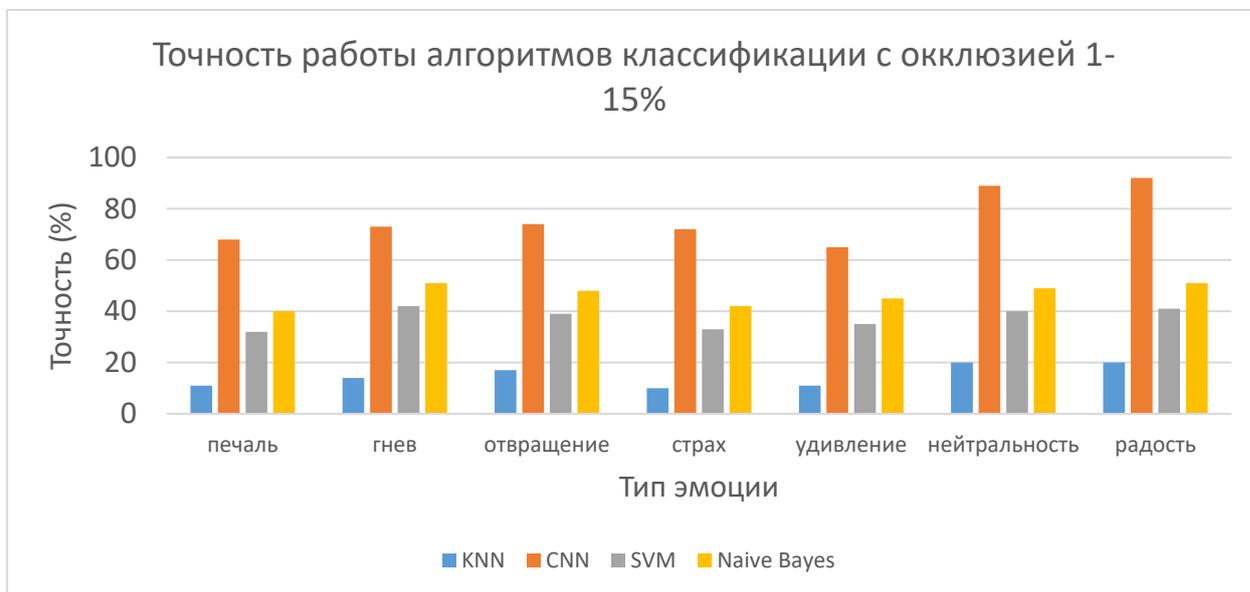


Рисунок 9 – Гистограмма с результатами замера точности работы классификаторов с учётом окклюзии

Как можно видеть из рисунка 9, появление окклюзии сказалось на точности всех алгоритмов. Более устойчивым, как и предполагалось оказался алгоритм классификации CNN, его точность снизилась в среднем на 6% и составила 76%, следующим алгоритмом по точности является наивный байесовский метод его точность работы снизилась на 11% и составляет 46%, что также является ожидаемым поскольку наивный байесовский метод можно верно производить классификация даже при малом количестве примеров. Среднее значение точности алгоритма классификации SVM уменьшилось на 23% процента и теперь составляет 37%. И наименее точным оказался алгоритм классификации KNN точность его работы уменьшилась на 22% и теперь составляет всего 14%, связано это с тем что KNN в значительной мере зависит от количества примеров, как для обучения, так и для самой классификации.

Таким образом можно сделать вывод, что наибольшая точность алгоритма распознавания эмоций будет достигнута с помощью связки двух свёрточных нейронных сетей, первая занимается классификацией лиц после чего к ней применяется технология DNN для получения границ лица, затем данная часть изображения передаётся во вторую свёрточную нейронную сеть, которая занимается классификацией эмоций. Именно эта связка и будет выбрана как базовый алгоритм распознавания эмоций, точность которого планируется увеличивать в задачи распознавания эмоций на лицах, положение которых отлично от фронтального более чем на 45 градусов.

1.4 Описание используемых моделей алгоритмов классификации

В данном разделе для каждого алгоритма будут представлены математическая модель, а также модель алгоритма в виде блок-схемы.

В алгоритме классификации KNN для нормализации данных будет использоваться MinMax-нормализация, которую можно представить формулой (1).

$$x_{norm_i} = \frac{x_i - \min(X)}{\max(X) - \min(X)}, \quad (1)$$

где X – набор входных данных;

x_i – элемент набора данных, который требуется нормализовать.

Данная нормализация была выбрана, потому что она значительно снижает признаковое пространство, в результате её применения, значения будут находиться на отрезке [0;1].

В данной модели алгоритма KNN для вычисления расстояния от классифицируемой точки до всех остальных будет использоваться евклидова метрика, которая рассчитывается по формуле (2).

$$d(x^*, x) = \sqrt{\sum_{i=1}^n (x_i^* - x_i)^2}, \quad (2)$$

где x_i^* – i -ый признак классифицируемой точки;
 x_i – i -ый признак точки из набора данных;
 n – размер признакового пространства.

Число ближайших соседей K будет выбираться по формуле (3).

$$K = \begin{cases} \lfloor \sqrt{n} \rfloor - 1, & \text{если } \lfloor \sqrt{n} \rfloor \bmod 2 = 0 \\ \lfloor \sqrt{n} \rfloor, & \text{если } \lfloor \sqrt{n} \rfloor \bmod 2 \neq 0 \end{cases}, \quad (3)$$

где n – количество входных данных.

Таким образом по формуле (3), будет выбрано оптимальное число соседей K , которое будет удовлетворять следующим критериям:

- изменяться с изменением количества данных в наборе данных;
- будет нечётным, что позволит избежать ситуации, когда равное число соседей относятся к разным классам;
- не будет слишком большим, что позволит ему не выродиться в константу.

Таким образом формулы (1-3) представляют собой математическую модель алгоритма классификации KNN.

Далее на рисунке 10 будет представлена блок-схема модели алгоритма классификации эмоций KNN.



Рисунок 10 – Модель алгоритма классификации эмоций KNN

На рисунке, представленном выше можно видеть модель алгоритма KNN. На подготовительном этапе в алгоритм поступает уже размеченный набор данных, набор примеров, которые необходимо классифицировать, а также рассчитывается число K . Затем на этапе нормализации данных, размеченный набор с разным признаковым пространством, преобразуется к набору данных с признаковым пространством на отрезке $[0;1]$. После этого на этапе расчёта расстояний для всех примеров, которые необходимо классифицировать, вычисляются расстояния до каждого примера из набора размеченных данных. И наконец на этапе определения метки класса, полученный набор расстояний сортируется по убыванию и из него выбираются первые K элементов, определяется их метка класса, и та метка класса, которая встречается чаще остальных, назначается для

классифицируемого примера, затем данный пример попадает в набор размеченных данных и в последствии используется для классификации других примеров.

Для алгоритма классификации эмоций SVM сначала будет представлена модель, которая работает в случаях линейной разделимости данных, а затем модель, которая работает в случаях линейной неразделимости, так как именно такие данные встречаются в задаче классификации эмоций. Стоит отметить, что в обоих случаях первым этапом является нормализация данных, которая производится по той же формуле что и в алгоритме KNN, а именно по формуле (1).

Задача построения гиперплоскости в случае линейной разделимости данных является задачей квадратной оптимизации, которая представлена формулой (4).

$$\begin{cases} \|w\|^2 \rightarrow \min \\ c_i(w * x_i - b) \geq 1, 1 \leq i \leq n \end{cases} \quad (4)$$

где w – вектор весовых коэффициентов;

c_i – i -ая метка класса, представлена числом -1 или 1;

x_i – i -ый пример (вектор признаков);

b – коэффициент;

n – размер набора данных.

Исходя из теоремы Куна–Таккера задача, представленная формулой (4) эквивалентна задаче поиска седловой точки функции Лагранжа, которая представлена формулой (5).

$$\begin{cases} L(w, b; \lambda) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \lambda_i (c_i(w * x_i - b) - 1) \rightarrow \min_{w,b} \max_{\lambda} \\ \lambda_i \geq 0, 1 \leq i \leq n \end{cases}, \quad (5)$$

где $\lambda = (\lambda_1, \dots, \lambda_n)$ – вектор двойственных переменных.

Задача, представленная формулой (5) сводится к эквивалентной задаче квадратичного программирования, в которой будут содержаться только двойственные переменные, данная задача описана формулой (6).

$$\left\{ \begin{array}{l} -L(\lambda) = -\sum_{i=1}^n \lambda_i + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j c_i c_j x_i x_j \rightarrow \min_{\lambda} \\ \lambda_i \geq 0, 1 \leq i \leq n \\ \sum_{i=1}^n \lambda_i c_i = 0 \end{array} \right. \quad (6)$$

В результате решения данной задачи, параметры w и b можно вычислить по формуле (7).

$$\begin{aligned} w &= \sum_{i=1}^n \lambda_i c_i x_i \\ b &= w * x_i - c_i, \lambda_i \end{aligned} \quad (7)$$

Таким образом алгоритм классификации для линейно разделимых данных можно представить формулой (8).

$$a(x) = \operatorname{sgn}\left(\sum_{i=1}^n \lambda_i c_i x_i * x - b\right) \quad (8)$$

Стоит отметить, что суммирование производится не по всему набору данных, а только по опорным векторам, для которых выполняется условие $\lambda_i \neq 0$.

Далее, чтобы алгоритм, представленный формулой (8) можно было применить к линейно неразделимым данным, требуется внести изменения, позволить алгоритму допускать ошибки при обучении. За основу можно взять формулу (5), после её изменения данную задачу можно представить формулой (9).

$$\begin{cases} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \rightarrow \min_{w,b,\xi_i} \\ c_i(w * x_i - b) \geq 1 - \xi_i, 1 \leq i \leq n, \\ \xi_i \geq 0, 1 \leq i \leq n \end{cases} \quad (9)$$

где ξ_i – значение характеризующее величину ошибки i -го примера x ;
 C – параметр, благодаря которому можно регулировать отношение между максимизацией ширины разделяющей полосы и минимизацией суммарной ошибки.

Далее задача, представленная формулой (9) сводится к задаче поиска седловой точки функции Лагранжа, которая представлена формулой (10).

$$\begin{cases} L(w, b, \xi; \lambda, \eta) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \lambda_i (c_i(w * x_i - b) - 1) - \\ - \sum_{i=1}^n \xi_i (\lambda_i + \eta_i - C) \rightarrow \min_{w,b,\xi} \max_{\lambda,\eta} \\ \xi_i \geq 0, \eta_i \geq 0, \lambda_i \geq 0, 1 \leq i \leq n \\ \begin{cases} \lambda_i = 0 \\ c_i(w * x_i - b) \geq 1 - \xi_i, 1 \leq i \leq n \end{cases} \\ \begin{cases} \eta_i = 0 \\ \xi_i = 0, 1 \leq i \leq n \end{cases} \end{cases}, \quad (10)$$

Далее задача, представленная формулой (10) сводится к эквивалентной двойственной задаче представленной формулой (6), после чего алгоритм работает по формулам (7) и (8). Отличие заключается в том, что в формуле (8) суммирование будет происходить не только по опорным векторам, но и по примерам-нарушителям, благодаря чему производится более гибкое построения гиперплоскости.

Также можно было заметить, что алгоритм SVM применяется для случаев бинарной классификации, то есть когда есть только два класса, поэтому чтобы его можно было применить к задаче классификации эмоций, в которой используется семь классов, будет применён метод классификации, который называется один к одному. Для каждой пары классов будет строиться своя гиперплоскость, таким образом можно будет достичь более высокой точности и избежать сложных расчётов для адаптации алгоритма к многоклассовости.

На этом описание математической модели для алгоритма SVM завершено. При построении блок-схемы алгоритма, она была разделена на две части, обучения модели и работы с реальными данными. Далее на рисунке 11 будет приведена блок-схема для фазы обучения модели алгоритма классификации эмоций SVM.



Рисунок 11 – Модель алгоритма классификации эмоций SVM на фазе обучения

На рисунке 11 представлена модель алгоритма SVM, которая будет использована на фазе обучения. На подготовительном этапе происходит

загрузка данных для обучения и инициализация начальных весовых коэффициентов. На этапе нормализации данные для обучения проходят нормализацию данных в результате чего их признаковое пространство снижается. Затем на этапе вычислений и корректировки происходят вычисления обновлённых весовых коэффициентов, после чего проверяется удалось ли построить оптимальную гиперплоскость, если да работа алгоритма заканчивается, если нет производится корректировка весовых коэффициентов и проводится повторная проверка, данные действия повторяются до тех пор, пока не будет построена оптимальная гиперплоскость.

Далее на рисунке 12 будет представлена модель алгоритма классификации эмоций SVM, которая работает с реальными данными.



Рисунок 12 – Модель алгоритма классификации эмоций SVM на работы

На рисунке, представленном выше можно видеть этапы, которые выполняются в ходе работы модели алгоритма SVM с реальными данными. На подготовительном этапе происходит загрузка обученной ранее модели и

загрузка набора данных для классификации. Далее на этапе нормализации у данных для классификации изменяется признаковое пространство, которое приравнивается к признаковому пространству данных для обучения. На этапе построения гиперплоскости, по весовым коэффициентам обученной модели происходит построение функции гиперплоскости. На этапе определения метки класса производятся вычисления, которые позволяют определить в какой части гиперплоскости находится классифицируемая точка в зависимости от чего ей присваивается одна из двух соответствующих меток класса.

Далее можно перейти к наивному байесовскому методу. Основой данного алгоритма конечно же является теорема Байеса, которая представлена формулой (11).

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}, \quad (11)$$

где $P(Y|X)$ – вероятность гипотезы Y при наступлении события X ;

$P(X|Y)$ – вероятность наступления события X при том, что гипотеза Y истинна;

$P(Y)$ – априорная вероятность гипотезы Y ;

$P(X)$ – вероятность наступления события X .

Далее необходимо преобразовать формулу (11) с учётом n -мерного вектора признаков X , получится формула (12).

$$P(Y|x_1, \dots, x_n) = \frac{P(x_1, \dots, x_n|Y)P(Y)}{P(x_1, \dots, x_n)} \quad (12)$$

Можно заметить, что в формулах (11) и (12), знаменатель дроби не зависит от Y , поэтому он является константой и на практике его можно не использовать.

Поэтому формулу (12) без знаменателя и в более развёрнутом виде можно представить формулой (13).

$$P(Y|x_1, \dots, x_n) = P(x_1|Y)P(x_2|Y, x_1) * \dots * P(x_n|Y, x_1, x_2, \dots, x_{n-1})P(Y) \quad (13)$$

С учётом того, что байесовский метод является наивным все признаки, содержащиеся в векторе X, будут независимыми, с учётом данного допущения формулу (13) можно представить формулой (14).

$$P(Y|x_1, \dots, x_n) = P(x_1|Y)P(x_2|Y) * \dots * P(x_n|Y)P(Y) = P(Y) * \prod_{i=1}^n P(x_i|Y) \quad (14)$$

Формула (14) является основой для наивного байесовского метода, но в связи с тем, что для её применения необходимо построение таблиц частотности и правдоподобия, которые невозможно построить для числовых признаков, данную формулу также необходимо преобразовать.

К числовым признакам, которые и будут использоваться при классификации будут применено нормальное распределение плотность которого зависит от таких параметров как среднее и среднеквадратичное отклонение, их можно рассчитать по формулам (15) и (16).

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (15)$$

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (16)$$

После вычисления среднего и среднеквадратичного отклонения плотность нормального распределения можно вычислить по формуле (17).

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\bar{x}}{\sigma}\right)^2} \quad (17)$$

С использованием формулы (17), можно переписать формулу (14), результат будет представлен в виде формулы (18).

$$P(Y|x_1, \dots, x_n) = P(Y) * \prod_{i=1}^n P(x_i|Y) = P(Y) * \prod_{i=1}^n f(x_i) \quad (18)$$

По формуле (18) можно производить вычисления и определять вероятность того, что классифицируемый пример принадлежит к тому или иному классу. И чтобы сразу получать необходимую метку класса требуется брать значение Y при котором вероятность, вычисляемая по формуле (18) наибольшая, решение данной задачи представлено формулой (19).

$$y_{res} = \operatorname{argmax}_Y (P(Y) * \prod_{i=1}^n f(x_i)) \quad (19)$$

Формулу (19) можно использовать непосредственно в модели алгоритма классификации для определения метки класса.

Далее на рисунке 13 будет представлена блок-схема модели наивного байесовского метода для классификации эмоций.

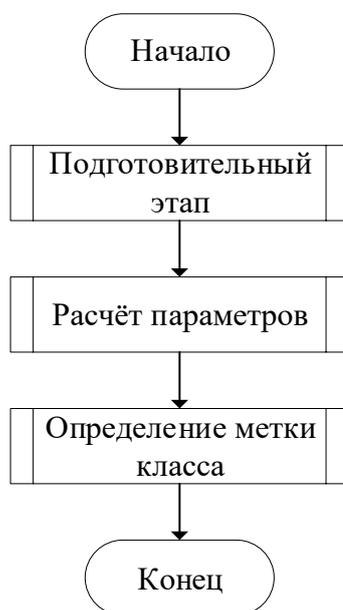


Рисунок 13 – Модель наивного байесовского метода для классификации эмоций

На рисунке 13 можно видеть этапы, которые выполняются в ходе работы модели наивного байесовского метода. На подготовительном этапе происходит загрузка статистических данных, а также загрузка данных для классификации. Далее на этапе расчёта параметров производится вычисление среднего и среднеквадратичного отклонения для каждого класса. На этапе определения метки класса происходит подстановка в формулу (19) вычисленных значений среднего, среднеквадратичного отклонения для каждого класса, а также всех значений признаков в результате чего для классифицируемого примера сразу же определяется метка класса. Стоит отметить, что в данной модели байесовского наивного метода нет необходимости в нормализации данных, так как размерность признакового пространства не сказывается на работе данной модели.

Далее можно приступать к построению математической модели для алгоритма свёрточной нейронной сети (CNN). Сперва будет представлено описание математической модели свёрточного слоя.

На данном слое операция, которая является основной, называется свёртка, она вычисляется по формуле (20):

$$w_{k,v}^z = \{w_{k,v}^z(n, u)\}, \quad (20)$$

где $w_{k,v}^z$ – свёртка, которая была использована для карты признаков k слоя $(z-1)$, на слое z при помощи карты признаков v .

Результатом свёрточного слоя z , будет являться карта признаков v , которую можно вычислить по формуле (21).

$$y_v^z = f_z(\sum_{k \in V_v^z} y_k^{z-1} \otimes w_{k,v}^z + b_v^z), \quad (21)$$

где y_v^z – v -ая карта признаков полученная на выходе слоя z ;

f_z – активационная функция применяем к слою z ;

V_v^z – список всех уровней слоя $(z-1)$, которые соединяются с картой признаков v слоя z ;

b_v^z – смещения, которые применяются к карте признаков v слоя z .

В представленной выше формуле (21) используется активационная функция, которой в данной модели CNN будет являться ReLU. Эту функцию можно вычислить, используя формулу (22).

$$f_z(i) = \max(0, i) \quad (22)$$

Размер выходной карты признаков вычисляется по формуле (23).

$$len^z = (h^{z-1} - r^z + 1) \cdot (w^{z-1} - c^z + 1), \quad (23)$$

где len^z – размер карты признаков слоя z ;

h^{z-1} – высота исходной карты признаков слоя $(z-1)$;

r^z – высота используемой свёртки на слое z ;

w^{z-1} – ширина исходной карты признаков слоя $(z-1)$;

c^z – ширина используемой свёртки на слое z .

Выше были представлены формулы, которые описывали математическую модель свёрточного слоя. Ниже будет представлено описание математической модели для слоя подвыборки.

На слое подвыборки происходит уменьшение признакового пространства, эти преобразование можно выполнить по формуле (24):

$$a_v^{z-1} = \max(y_v^{z-1}(2n - 1, 2u - 1), y_v^{z-1}(2n - 1, 2u), y_v^{z-1}(2n, 2u - 1), y_v^{z-1}(2n, 2u)) \quad (24)$$

где a_v^{z-1} – модуль максимального элемента, который содержится в области под выборки;

y_v^{z-1} – v -ая карта признаков полученная с свёрточного $(z-1)$ слоя.

Также необходимо отметить, что максимальный пулинг используемый в формуле (24) является наиболее эффективным, что было доказано в результате многих исследований.

Элементы полученные в результате работы слоя подвыборки объединяются в матрицу подвыборки по формуле (25).

$$a_v^{z-1} = \{a_v^{z-1}(n, u)\} \quad (25)$$

В результате работы слоя подвыборки будет получена v -ая карта признаков для слоя z , данный процесс представлен формулой (26).

$$y_v^z = a_v^{z-1} \times w_v^z + b_v^z \quad (26)$$

Высота и ширина полученной карты признаков рассчитывается по формулам (27) и (28).

$$h^z = \frac{h^{z-1}}{2}, \quad (27)$$

$$w^z = \frac{w^{z-1}}{2}. \quad (28)$$

Выше было представлено описание математической модели для слоя подвыборки. Далее можно было бы привести описание математической модели активационного слоя, но он уже встроен в свёрточный слой и поэтому его отдельное описание не требуется. Далее будет представлено описание математической модели полносвязного нейронного слоя.

Модуль v -го выходного нейрона может быть рассчитан по формуле (29).

$$y_v^z = f_z(\sum_{k=1}^{V^{z-1}} y_k^{z-1} \times w_{k,v}^z + b_v^z), \quad (29)$$

где V^{z-1} – количество нейронов представленное на $(z-1)$ слое.

В данной модели свёрточной нейронной сети на слое выходных данных активационной функцией является sigmoid, это позволяет сократить возможный разброс значений и стабилизировать их на интервале $(0;1)$ для расчёта будет использоваться формула 30.

$$f_l(x) = \frac{1}{1 + e^{-x}} \quad (30)$$

В результате работы данной модели свёрточной нейронной сети будет построен вектор признаков значения, которого будут отражать критерий силы присутствия то или иной эмоции на изображении, для построения которого используется формула (31).

$$y = [y_1^Z, y_2^Z, \dots, y_{V^Z}^Z], \quad (31)$$

где Z – номер выходного слоя нейронной сети.

Также не стоит забывать о том, что данные перед тем как подавать в модель свёрточной нейронной сети необходимо нормализовать, а именно уменьшить их признаковое пространство, для этого все цветные изображения переводятся в оттенки серого, а после этого получившиеся интенсивности цвета пикселя делятся на максимальное значение интенсивности цвета это позволяет ускорить работу CNN. Данные преобразования представлены формулами (32), (33).

$$frame_gray = frame_0 \times 0,299 + frame_1 \times 0,587 + frame_2 \times 0,114 \quad (32)$$

$$frame_gray_normalized = \frac{frame_gray}{255} \quad (33)$$

На этом описание математической модели алгоритма свёрточной нейронной сети завершено. Далее будет представлена архитектура используемой свёрточной нейронной сети, а также алгоритм её работы в виде блок-схемы. Стоит отметить, что у CNN также, как и у SVM присутствует фаза обучения, но поскольку уже существует множество эффективных готовых решений для обучения свёрточных нейронных сетей одно из них и будет использоваться, поэтому отдельное представление алгоритма обучения не требуется.

На рисунке 14 представленном ниже будет изображён алгоритм работы модели алгоритма свёрточной нейронной сети (CNN).



Рисунок 14 – Модель алгоритма CNN

На рисунке, представленном выше можно видеть модель алгоритма свёрточной нейронной сети. На подготовительном этапе считывается изображение содержащее лицо, производится простановка параметров классификации. На этапе нормализации изображение из формата RGB переводится в оттенки серого, а также производится деление интенсивностей цвета пикселей на максимальную интенсивность цвета. После этого данные попадают в модель алгоритма CNN, где начинают работать свёрточные слои и слои подвыборки выделяя всё более высокоуровневые признаки. Затем полносвязные слои обрабатывают данные полученные из первой части CNN постепенно уменьшая их количество и на слой выходных данных выводятся вероятности принадлежности обнаруженной эмоции к тому или иному классу.

Далее на рисунке 15 будет представлена архитектура используемой нейронной сети.

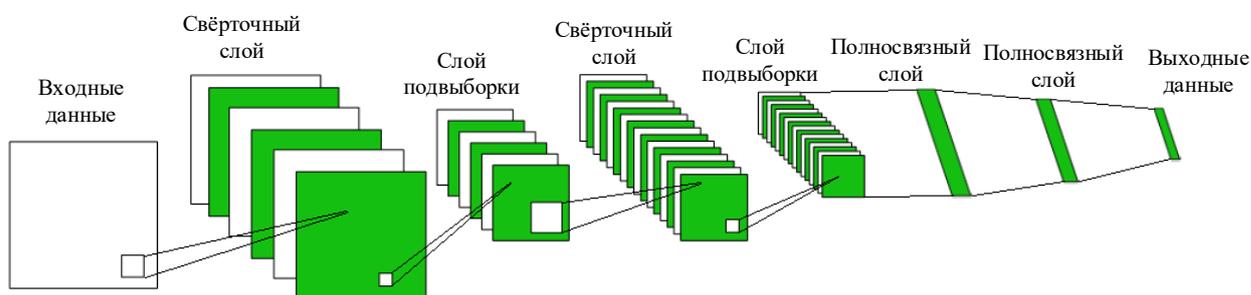


Рисунок 15 – Архитектура свёрточной нейронной сети LeNet 5

Из рисунка 15 можно видеть, что модель свёрточной нейронной сети состоит из двух частей, в которых присутствуют разнообразные слои. Первая часть CNN состоит из слоя входных данных, двух свёрточных слоёв и двух слоёв подвыборки. Свёрточные слои, как уже было сказано ранее имеют функцию активации ReLU, на слоях подвыборки применяется максимальный пулинг. Вторая часть свёрточной нейронной сети состоит из двух полносвязных слоёв и слоя выходных данных.

В данном разделе был дан вводный экскурс в задачу классификации эмоций, рассмотрены различные подходы к её решению, представлены наиболее известные и эффективные программные решения для классификации эмоций, были представлены модели наиболее известных классификаторов, а также был произведён выбор наиболее высокоточной модели, которая в дальнейшем будет модернизирована для повышения точности её работы.

2 Проектирование моделей алгоритма для распознавания эмоций

2.1 Проектирование модели алгоритма распознавания лиц

Ранее была представлена модель алгоритма свёрточной нейронной сети, которая будет использоваться как для классификатора лиц, так и для классификатора эмоций. Также уже говорилось, что в случае с CNN для классификации лиц, к ней применяется технология DNN, которая по последнему слою свёрточной нейронной сети определяет признаки, используемые на этом слое и где они находятся на изображении в результате выводятся границы, которые содержащие эти признаки и так как они использовались для классификации, это область будет содержать в себе искомый объект в данной работе человеческое лицо.

Пример работы алгоритма распознавания лиц будет представлен на рисунке 16.

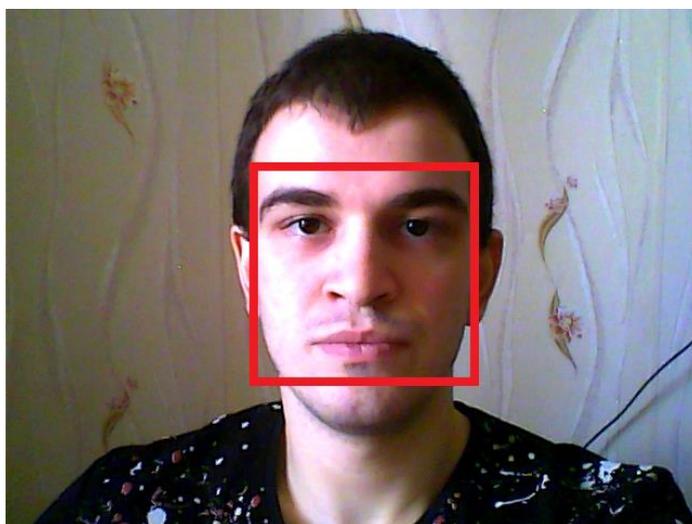


Рисунок 16 – Пример работы алгоритма распознавания лиц DNN

На рисунке, представленном выше можно видеть, что технология DNN применённая к классификатору лиц CNN успешно справилась со своей

задачей и выделила на изображении область, в которой содержится лицо. Но можно заметить, что в выделенную область не попали лоб и часть подбородка, что несомненно в последствии повлияет на точность классификации эмоций так как не все части лица, передающие эмоции были захвачены. Таким образом можно сделать вывод, что необходимо увеличение области, в которой будет располагаться лицо, это может повысить точность работы классификатора эмоций.

Это можно сделать путём введения некоего коэффициента k , инициируемого производным значением, на который будет умножаться область, выделенная технологией DNN, это позволит равномерно увеличить область в k раз. Пример применения данного подхода представлен ниже на рисунке 17.

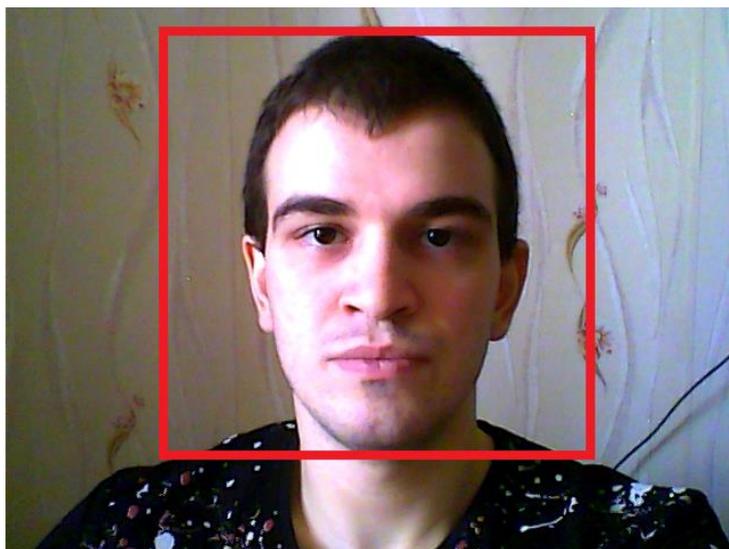


Рисунок 17 – Пример работы алгоритма распознавания лиц DNN с применением коэффициента

На рисунке, представленном выше можно видеть, как работает предложенный ранее подход. Можно заметить, что область содержащая лицо увеличилась и теперь в неё действительно входят все части лица. Но также можно заметить, что равномерное увеличение области привело к попаданию в

неё частей изображения, которые не содержат лица, на этапе классификации эмоций они не несут полезной информации и поэтому их не нужно выделять.

Преимуществами предложенного подхода являются:

- простота реализации;
- коэффициент является гиперпараметром и поэтому может быть удобно настроен для каждого конкретного изображения.

Недостатками данного подхода являются:

- равномерное увеличение области, в результате чего захватываются не нужные для дальнейшей классификации эмоций области изображения;
- не универсальность коэффициента, в случае применения к изображениям с разным масштабом и соотношением высоты и ширины;
- не применим для видеопотока.

Перечисленные выше недостатки являются достаточно серьёзными, так как они значительно будут ограничивать область применения данного распознавателя лиц. Поэтому было принято решение от него отказаться и продолжить дальнейшее исследование.

Следующий подход связан с непосредственным применением технологии DNN. Как уже отмечалось в предыдущим разделе основной областью её применение является выделение иерархий изображений, полученных с помощью каждого слоя CNN, для дальнейшего анализа точности работы данной сети. В задаче распознавания лиц можно воспользоваться особенностями данной технологии для увеличения области распознавания, при этом данная область будет увеличиваться не равномерно, а только там, где действительно присутствуют признаки, которые могут являться частью лица.

В технологии DNN применяются операции обратные операциям в свёрточной нейронной сети, а именно:

- обратный пулинг;
- ректификация;
- обратная свёртка.

Обратный пулинг – операция, позволяющая вернуть размерность карты признаков до того, как к ней была применена операция пулинга. Так как сама операция пулинга является необратимой и после неё невозможно восстановить точные значения карты признаков, то восстанавливается только размерность карты признаков за счёт того, что известная размерность ядра на слое подвыборки, а также примерные значения за счёт переменных, размещённых на слое выше.

Ректификация – применение точно такой же функции активации, которая используется на свёрточном слое для выходных данных, только теперь она применяется к данным из обратного пулинга.

Обратная свёртка – её суть заключается в том, чтобы получить значения фильтров, которые применялись, к той или иной карте признаков, для этого используются транспонированные версии фильтров.

Далее стоит вспомнить рисунок 15 на котором представлена архитектура используемой свёрточной нейронной сети. Видно, что последним слоем является слой подвыборки, на данный момент именно к нему применяется технология DNN, то есть по признакам с данного слоя определяются границы изображения, в которых присутствуют указанные признаки. Для того, чтобы увеличить размерность карты признаков, а соответственно и размерность области изображения, в которой содержатся данные признаки необходимо произвести операцию обратного пулинга. Для этого матрица признаков слоя подвыборки будет обработана в результате чего каждый её элемент будет разбит на количество элементов равному размерности ядра подвыборки.

Далее на рисунке 18 будет продемонстрирована операция обратного пулинга.

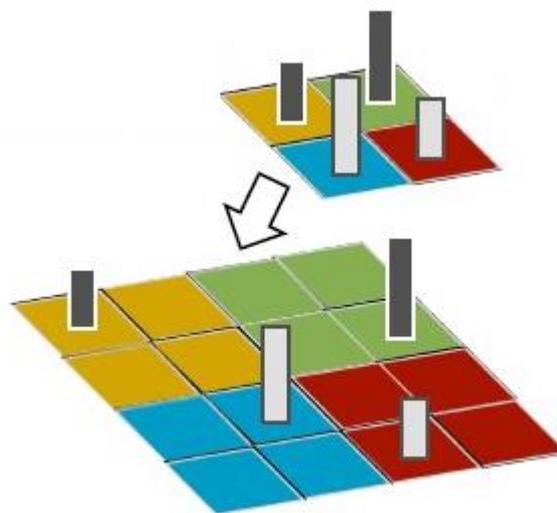


Рисунок 18 – Пример операции обратного пулинга

В данном алгоритме для максимального повышения области выделения значения максимумов будут записываться не случайным образом в пределах, какой-либо области равной размерности ядра, они будут записываться таким образом, чтобы оказаться внутренними. Таким образом в результате применения обратного пулинга будут получены карты признаков, которые были использованы в свёрточной нейронной сети после применения функции активации ReLU к второму свёрточному слою. После применения к данным картам признаков технологии DNN будет получена область, которая содержит признаки больше размерности, а соответственно это будет означать, что область изображения будет увеличена.

На рисунке 19 будет представлен результат применения технологии DNN не к последнему слою подвыборки, а к последнему слою свёртки.

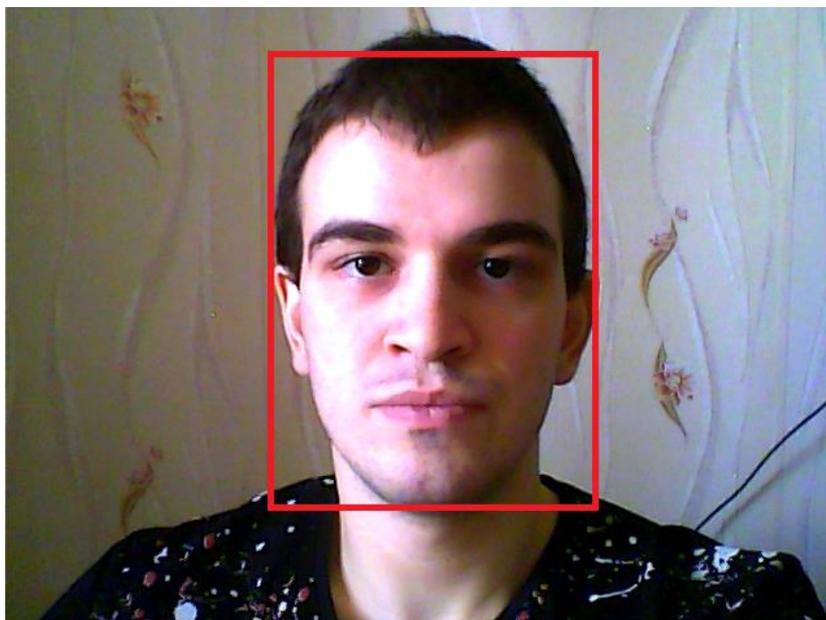


Рисунок 19 – Пример работы алгоритма распознавания лиц DNN с применением обратного пулинга

Как можно видеть из рисунка 19 область выделения увеличилась по сравнению с исходной, она захватывает всё лицо и не захватывает лишние области, которые не несут полезной информации для классификации.

Преимуществами данного подхода являются:

- захват всего лица при этом почти отсутствуют области с не нужной информацией для классификации эмоций;
- не зависит от размера изображения;
- может применяться на видеопотоке.

Недостатком данного подхода является проведение дополнительных вычислений – операция обратного пулинга.

Таким образом можно сделать вывод, что технологию DNN для выделения большей области на изображении можно эффективно применять на

более низких слоях свёрточной нейронной сети. В данной работе планируется использовать именно этот подход. Учитывая его, алгоритм работы распознавателя лиц можно описать следующей блок-схемой, представленной на рисунке 20.

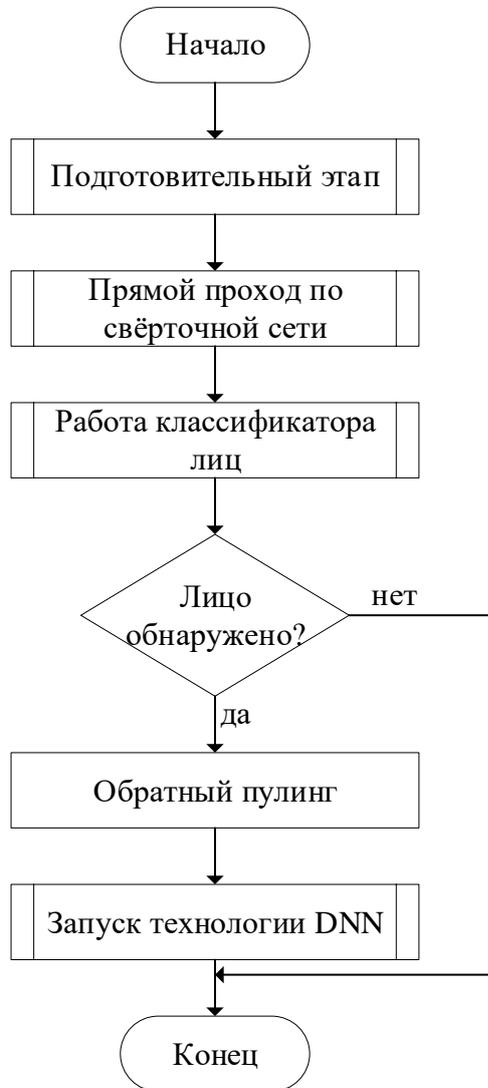


Рисунок 20 – Модель алгоритма распознавания лиц DNN

Из рисунка 20 можно видеть, что на подготовительном этапе происходит загрузка модели классификатора лиц CNN и проставление параметров классификации. На этапе прямого прохода по свёрточной сети, выполняются стандартные операции для свёрточной нейронной сети. На этапе работы

классификатора полносвязные слои обрабатывают информацию, полученную от свёрточных слоёв и слоёв подвыборки, благодаря соединению Гаусса происходит классификация. Затем если лицо было обнаружено, запускается процесс обратного пулинга, а после к полученной карте признаков применяется технология DNN, которая по картам признаков позволяет определить область содержащую лицо на изображении.

На этом проектирование модели распознавания лиц завершено.

2.2 Проектирование модели алгоритма классификации эмоций

Для повышения точности алгоритма классификации эмоций было принято решения применить ко всем свёрточным слоям вместе с функцией активации ReLU пакетную нормализацию.

Пакетная нормализация хорошо зарекомендовала себя для повышения скорости обучения и стабилизации работы полносвязных нейронных сетей. Суть её заключается в том, что при реализации алгоритма градиентного спуска на каждой итерации обучающая выборка просматривается целиком и только после этого изменяются весовые коэффициенты модели. Также в нейронных сетях, где применялась пакетная нормализация отмечалось повышение точности классификации.

Применение пакетной нормализации к свёрточным слоям можно описать формулой (34).

$$z = g(BN(Wu)), \quad (34)$$

где z – выходные карты признаков свёрточного слоя;

g – функция активации свёрточного слоя, в используемой архитектуре CNN это ReLU;

BN – функция, обозначающая пакетную нормализацию;

W – вектор весовых коэффициентов.

Таким образом после применения пакетной нормализации изменится архитектура используемой свёрточной нейронной сети CNN, ранее представленная на рисунке 15. Новая архитектура алгоритма классификации CNN будет представлена на рисунке 21.

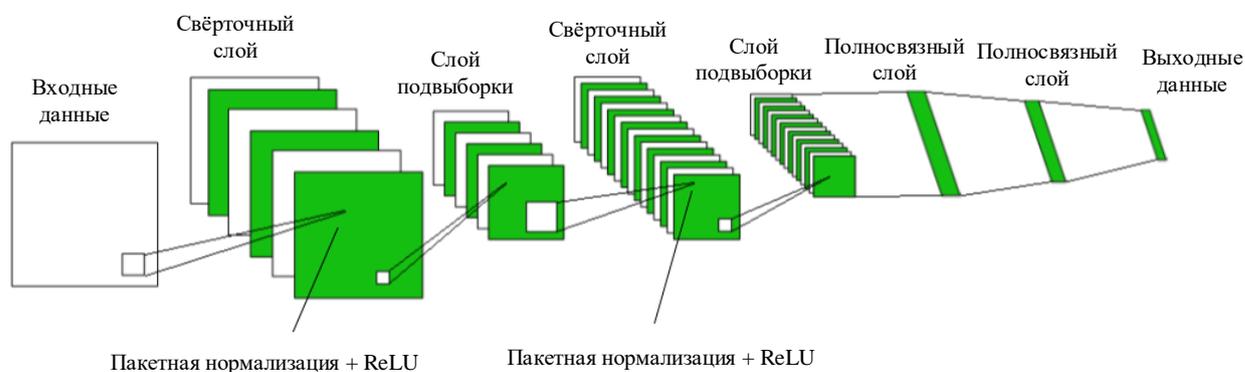


Рисунок 21 – Архитектура свёрточной нейронной используемой для классификации эмоций

Из рисунка 21 можно видеть, что на выходе свёрточного слоя теперь будет также применяться пакетная нормализация, при том согласно формуле (34) сначала будет применяться именно она, а потом уже будет вычисляться функция активации ReLU. Для блок-схемы на рисунке 14, где представлен алгоритм работы классификатора CNN изменения не требуются, так пакетная нормализация является частью свёрточных слоёв.

В данном разделе были рассмотрены основные способы увеличения точности работы алгоритмов распознавания лиц и классификации эмоций. Были спроектированы модель алгоритма распознавания лиц с увеличенной областью выделения лица при помощи применения технологии DNN, а также модель алгоритма классификации эмоций, благодаря пакетной нормализации, которая должно помочь повысить точность её работы.

3 Программные решения для реализации и сравнительный анализ представленных моделей

3.1 Программные решения, используемые для реализации спроектированных моделей

Для реализации существует различное множество программного обеспечения. Далее будет представлено описание программных решений, которые планируется использовать для реализации спроектированных ранее моделей.

В качестве языка программирования был выбран Python версии 3.10. Python – язык программирования, является высокоуровневым и интерпретируемым. Высокоуровневый означает, что язык программирования поддерживает абстракцию, то есть код написанный на таком языке будет работать не для какого-то конкретного компьютера, а для любой машины, на которую установлен соответствующий компилятор или интерпретатор. Интерпретируемый означает, что преобразование кода, написанного человеком в машинный код будет происходить во время выполнения программы. Данный язык программирования разработал Guido van Rossum в конце 1980-х годов [22]-[26]. Python способен поддерживать различные парадигмы программирования, такие как:

- структурное,
- объектно-ориентированное,
- аспектно-ориентированное,
- функциональное,
- императивное.

Преимуществами языка программирования Python являются:

- доступность и открытый исходный код;

- кроссплатформенность, при переносе между различными ОС достаточно чтобы в них был установлен Python;
- интеграция языков C и C++, позволяет расширить функционал основного языка;
- динамическая типизация, позволяет не акцентировать внимание на том какой тип должна принимать та или иная переменная;
- множество различных библиотек и фреймворков.

Недостатками языка программирования Python являются:

- сниженная скорость выполнения программы, данный недостаток свойственен всем интерпретируемым языкам программирования поскольку преобразование в машинный код происходит как раз на этапе выполнения программы;
- увеличенная нагрузка на память является следствием динамической типизации.

Данный язык программирования был выбран для реализации моделей прежде всего за то, что он очень гибкий, а также для него существует большое количество уже готовых библиотек и фреймворков для машинного обучения, примером таких являются TensorFlow и Keras.

TensorFlow – программная библиотека с открытым исходным кодом предназначена для машинного обучения, в частности может быть использована для создания нейронных сетей, а также их обучения. Разработчиком данной библиотеки является компания Google. Отмечается, что алгоритмы распознавания и классификации обученные с помощью библиотеки, а также платформы TensorFlow в этих вопросах не уступают человеку. Поскольку основной API данной библиотеки реализован для работы с Python, именно она и будет использоваться для обучения модели алгоритма CNN [18].

Keras – библиотека с открытым исходным кодом, написанная на языке программирования Python. Её ведущий разработчик François Chollet, инженер

компании Google. Данная библиотека представляет высокоуровневый API, которая позволяет с большим удобством заниматься созданием и обучением алгоритмов, не зависимо от используемого бэкенда. Обычно она является надстройкой над TensorFlow [28].

CUDA – это платформа параллельных вычислений и модель программирования, разработанная компанией Nvidia для общих вычислений на собственных графических процессорах (GPU). CUDA позволяет разработчикам ускорять приложения с интенсивными вычислениями, используя мощность GPU для распараллеливания части вычислений. Хотя были предложены и другие API для GPU, например, OpenCL, и существуют конкурирующие GPU от других компаний, например, AMD, сочетание CUDA и GPU Nvidia доминирует в нескольких областях применения, включая глубокое обучение, и является основой для некоторых из самых быстрых компьютеров в мире. В 2003 году группа исследователей под руководством Ian Buck представила Brook – первую широко распространенную модель программирования, расширяющую язык C за счет параллельных конструкций данных. Позже Ian Buck присоединился к компании Nvidia и возглавил запуск CUDA в 2006 году, первого коммерческого решения для вычислений общего назначения на графических процессорах. Основными языками программирования где применяется и для которых была написана данная платформа являются C, C++ и Fortran, но несмотря на это часть функций можно применять и языке Python. Разумеется, чтобы была возможность применить данную технологию необходимо чтобы в компьютере была видеокарта от компании Nvidia. На компьютере, где планируется обучать модель CNN такая видеокарта присутствует, ей является Nvidia GeForce GTX 1660 Super. Применение технологии CUDA позволит значительно ускорить процесс обучения модели свёрточной нейронной сети, но нужно отметить, что и без использования данной технологии его можно производить. В таком случае вычисления будут производится параллельно на том количестве потоков, которое равно количеству ядер процессоров или в случае с

некоторыми процессорами компании Intel это количество можно умножить на два, но по сравнению с технологией CUDA, которая может производить параллельные вычисления сразу на нескольких сотнях тысяч потоках, это значение не идёт ни в какое сравнение.

Ранее были представлены все программные решения, которые будут использоваться при реализации спроектированных моделей, в качестве языка программирования был выбран Python, для обучения модели CNN будет использоваться Keras + TensorFlow, для распараллеливания вычислений будет использоваться технология CUDA.

3.2 Сравнительный анализ реализованных моделей алгоритмов классификации эмоций

Сравнительный анализ реализованных моделей будет происходить по следующим параметрам:

- точность работы алгоритма на изображениях без окклюзии;
- точность работы алгоритма с учётом окклюзии, значение, которой не превышает 15% для той области лица, которая полезна для классификации.

Для тестирования точности будет выбрано по 500 изображений для каждого класса, каждое из которых удовлетворяет условию значительного отклонения лица от фронтального положения (более 45 градусов). Набором данных используемых как для обучения, так и для тестирования является AffectNet. Точность будет рассчитываться как отношение верно классифицированных эмоций для каждого класса к общему количеству изображений, представленных в тесте для каждого класса.

Сначала будет произведено сравнение моделей алгоритмов, спроектированных в ходе выполнения данной работы с первоначальной моделью.

На рисунке 22 представлен результат замера точности промежуточных моделей алгоритма распознавания эмоций без учёта окклюзии.

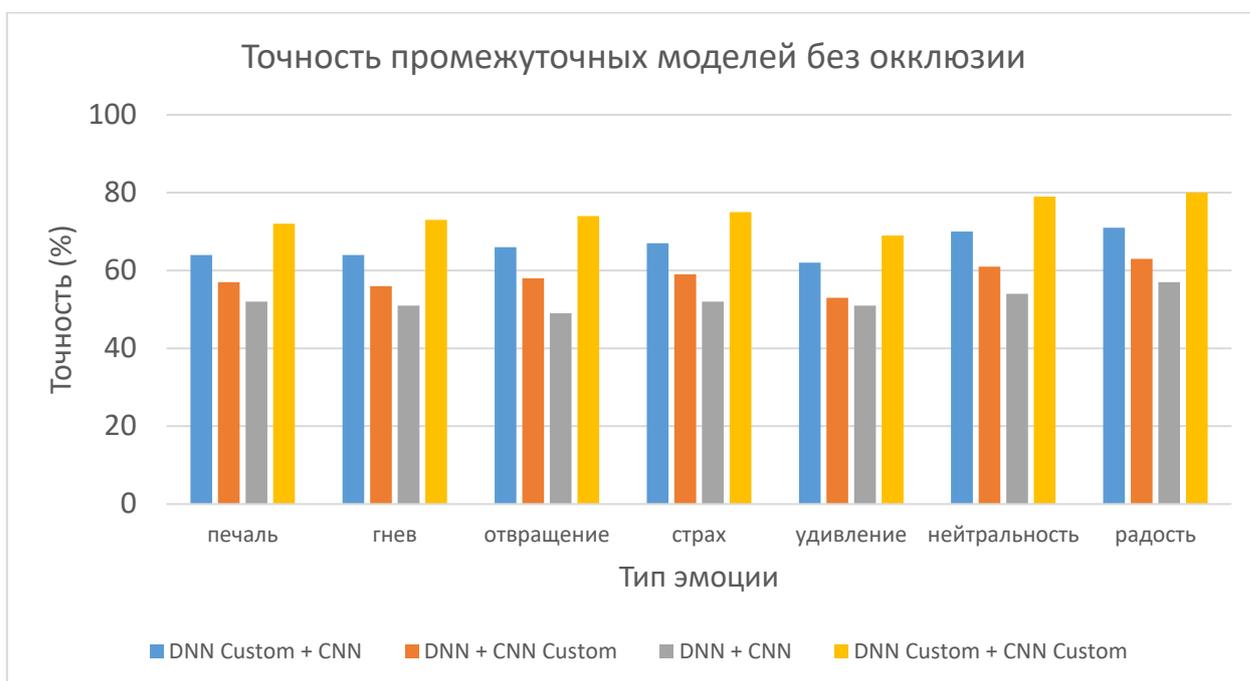


Рисунок 22 – Гистограмма с результатами замера точности промежуточных моделей без окклюзии

На рисунке 22, представлены четыре вида моделей, слово Custom означает, что модель была спроектирована в данной работе, таким образом производилось сравнение следующих моделей:

- базовой модели алгоритма распознавания эмоций;
- модели алгоритма распознавания эмоций с доработанным распознавателем лиц и базовым классификатором эмоций;
- модели алгоритма распознавания эмоций с доработанным классификатором эмоций и базовым распознавателем;
- модели алгоритма распознавания эмоций собранной из моделей, спроектированных в данной работе.

По рисунку 22 можно сделать вывод, что изначальные гипотезы полностью подтвердились алгоритм распознавания лиц с увеличенной областью для классификации, позволил повысить точность относительно базового алгоритма на 13,78%, что является значительным приростом, алгоритм классификации эмоций с применением пакетной нормализации позволил повысить точность распознавания эмоций на 5,93% относительно базового, а алгоритм обладающий обоими новшествами повысил точность относительно базового алгоритма на 22%. Средние точности алгоритмов, представленных на рисунке в порядке слева направо 66%, 58%, 52%, 74%.

Далее на рисунке 23 будут представлены результаты применения тех же алгоритмов к изображениям на которых присутствует окклюзия значение, которой не превышает 15%

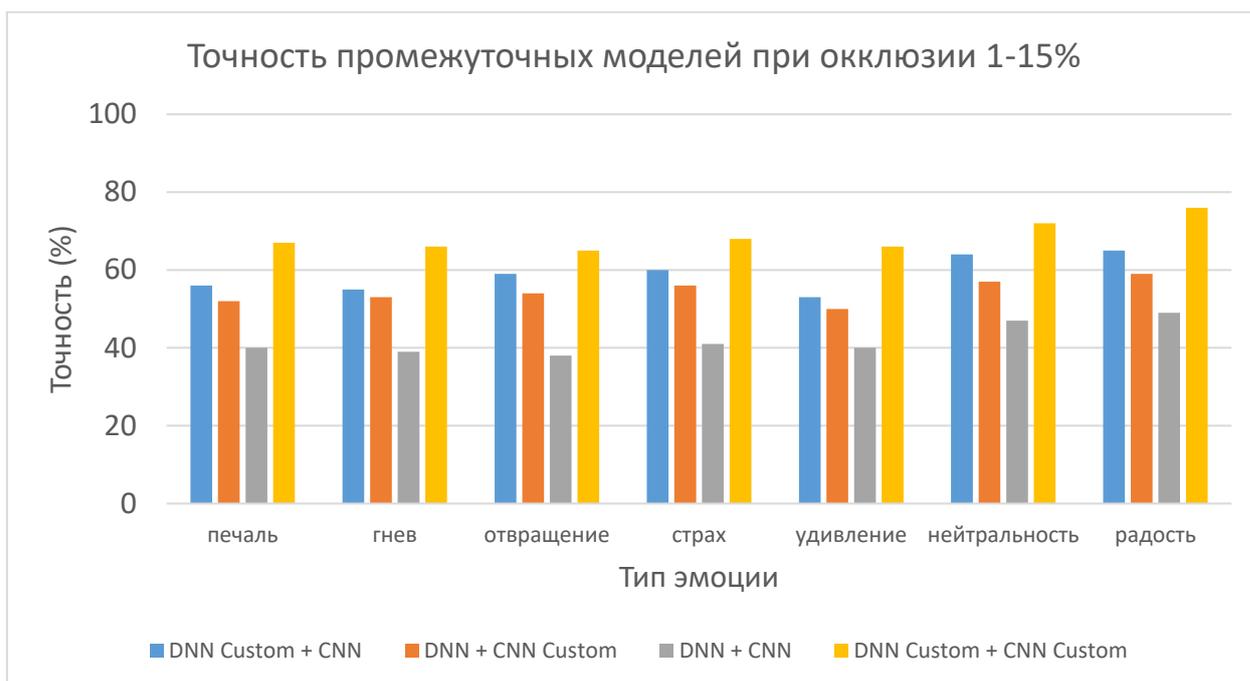


Рисунок 23 – Гистограмма с результатами замера точности промежуточных моделей с учётом окклюзии

Как можно видеть из рисунка 23 окклюзия повлияла на точность работы всех алгоритмов. Для базового алгоритма средняя точность упала на 10% и

теперь составляет 42%, для алгоритма с улучшенным распознавателем лиц точность упала на 8% и теперь составляет 58%, это является следствием того, что значимость размера области классификации при появлении окклюзии снижается, далее для алгоритма с улучшенным классификатором эмоций снижение точности составило всего 4% и его средняя точность равна 54%, наиболее же точным остаётся алгоритм включающий в себя обе модернизации, его точность упала на 6% и в среднем равняется 68%.

Таким образом можно сделать вывод, что объединение реализованных моделей значительно повысило точность распознавания эмоций на лицах отличных от фронтального положения более чем на 45 градусов.

В данном разделе были представлены программные решения, которые использовались для реализации ранее спроектированных моделей алгоритмов. Также был произведён сравнительный анализ реализованных моделей с аналогичными по результатам, которого наиболее точным является алгоритм распознавания эмоций с модернизированным распознавателем лиц и модернизированным классификатором эмоций.

Заключение

В результате выполнения данной работы была спроектирована и реализована модель алгоритма для распознавания эмоций, данная модель эффективно решает поставленную задачу, не смотря на значительное отклонение лица от фронтального положения (свыше 45 градусов). Для получения данного результата в каждом разделе работы решались свои задачи.

По итогам данной работы можно сделать следующие выводы:

- увеличение области выделяемой распознавателем лиц при помощи технологии DNN позволило увеличить точность работы алгоритма распознавания эмоций на 13,78% и составила 66% с учётом того, что на изображениях отсутствует окклюзия, в случае её присутствия данный показатель уменьшается до 58%, потому как значимость величины области при появлении окклюзии снижается;
- применение пакетной нормализации для свёрточных слоёв классификатора эмоций также позволило увеличить точность базовой модели на 5,93% и составила 58% для изображений на которых окклюзия отсутствует, с появлением окклюзии точность опустилась всего на 4% и составила 54% в среднем, что позволяет говорить о том, что данный алгоритм более устойчив к окклюзии чем базовая модель;
- объединение обеих описанных выше моделей в один алгоритм позволило повысить среднюю точность распознавания эмоций на 22%, и она равняется 74% для изображений без окклюзии, а для изображений с окклюзией средняя точность работы данного алгоритма равняется 68%;
- можно сделать вывод, что разработанный алгоритм распознавания эмоций позволяет качественно распознавать эмоций на лицах отличных от фронтального положения более чем на 45 градусов при том что на них будет отсутствовать окклюзия.

Список используемой литературы

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. Гарсия Г. Б., Исмаэль С.Г., Ноэлия В. Э. Обработка изображений с помощью OpenCV. М.: ДМК Пресс, 2016. 210 с.
4. 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.
5. 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.
6. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – Москва: Техносфера, 2012. – 1104 с.
7. Каллан, Р. Нейронные сети: Краткий справочник / Р. Каллан. - М.: Вильямс И.Д., 2017. - 288 с.
8. Клейнберг Дж., Тардос Е. Алгоритмы: разработка и применение. Классика Computers Science / Пер. с англ. Е. Матвеева. – СПб.: Питер, 2016. – 800 с.
9. Клетте Р. Компьютерное зрение. Теория и алгоритмы. М.: ДМК Пресс, 2019. 508 с.
10. Красильников Н. Цифровая обработка 2D- и 3D-изображений / Н. Красильников: Отдельное издание. - БХВ-Петербург, 2011. - 608 с.
11. Редько, В.Г. Эволюция, нейронные сети, интеллект: Модели и концепции эволюционной кибернетики / В.Г. Редько. - М.: Ленанд, 2019. - 224 с.

12. Шумский С.А. Машинный интеллект. Очерки по теории машинного обучения и искусственного интеллекта / С.А. Шумский. — М.: РИОР, 2019. - 340 с
13. Bagnall A, Lines J, Hills J, Bostrom A. Time-series classification with COTE: the collective of transformation-based ensembles. *IEEE Trans Knowl Data Eng.* 2015; 27:2522 – 2535. doi: 10.1109/TKDE.2015.2416723. – DOI.
14. Batista G, Keogh E, Tataw O, deSouza V. CID: an efficient complexity-invariant distance measure for time series. *Data Min Knowl Discov.* 2014;28(3):634–669. doi: 10.1007/s10618-013-0312-3. – DOI.
15. Baydogan M, Runger G. Time series representation and similarity based on local autopatterns. *Data Min Knowl Discov.* 2016;30(2):476–509. doi: 10.1007/s10618-015-0425-y. – DOI.
16. Benavoli A, Corani G, Mangili F. Should we really use post-hoc tests based on mean-ranks? *J Mach Learn Res.* 2016; 17:1–10.
17. Beyerer J. Automated Visual Inspection: Theory, Practice and Applications / J. Beyerer, P. Fernando, F. Christian. – Springer Berlin Heidelberg, 2016. – 798 p.
18. Bostrom Aaron, Bagnall Anthony. Big Data Analytics and Knowledge Discovery. Cham: Springer International Publishing; 2015. Binary Shapelet Transform for Multiclass Time Series Classification; pp. 257–269.
19. 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.
20. Chollet F. Deep Learning with Python – Manning Publications, 2017. – 384 p.
21. Foster D. Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play – O'Reilly Media, 2019. – 330 p.
22. 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.

23. Jeong Y, Jeong M, Omitaomu O. Weighted dynamic time warping for time series classification. *Pattern Recogn.* 2011; 44:2231 – 2240. doi: 10.1016/j.patcog.2010.09.022. - DOI
24. Lubanovic B. *Introducing Python: Modern Computing in Simple Packages Second Edition* – O'Reilly Media, 2019. – 605 p.
25. Lutz M. *Python Pocket Reference, 5th Edition* – O'Reilly Media, 2014. – 264 p.
26. McKinney W. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, 2nd Edition* – O'Reilly Media, 2017. – 544 p.
27. 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.
28. Singh P. *Learn TensorFlow 2.0: Implement Machine Learning and Deep Learning Models with Python / P. Singh, A. Manure* – Apress, 2019. – 177 p.
29. 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.
30. VanderPlas J. *Python Data Science Handbook: Essential Tools for Working with Data* – O'Reilly Media, 2016. – 672 p.