



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

Факультет
Кафедра

Информатика и системы управления
Программное обеспечение ЭВМ и ИТ, ИУ7

Отчет по НИР

Определение лицевых движений человека на фотографии

Студент

Спасенов Иван Владимирович
(фамилия, имя, отчество)

Группа

ИУ7-21М

Тип практики

Производственная

Название предприятия

МГТУ им. Н.Э. Баумана

Студент


подпись

Спасенов В.А.
Фамилия И.О.

Руководитель


подпись

Вишневская Т.И.
Фамилия, И.О.

Оценка: отлично

31.05.21

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

Кафедра «Программное обеспечение ЭВМ и ИТ» (ИУ7)

ЗАДАНИЕ
на прохождение производственной практики

на предприятии МГТУ им. Н.Э. Баумана

Студент

Спасенов Иван Владимирович, ИУ7-21М

(фамилия, имя, отчество; индекс группы)

Во время прохождения производственной практики студент должен:

- 1) ввести систему кодирования лицевых движений;
- 2) проанализировать существующие методы определения лицевых движений;
- 3) произвести поиск и сбор информации о методах детектирования лица и его ключевых точек;
- 4) проанализировать методы классификации и композиции методов машинного обучения.

Дата выдачи задания « 08 » 02 2021г.

Руководитель практики от кафедры



08.02.21

(подпись, дата)

Вишневецкая Т.И.

(Фамилия И.О.)

Студент



(подпись, дата)

Спасенов И.В.

(Фамилия И.О.)

ОГЛАВЛЕНИЕ

Введение	3
Обзор методов	4
Заключение	38
Список использованных источников	40

ВВЕДЕНИЕ

Целью работы является обзор текущих решений методов определения лицевых движений человека на фотографии, а также обзор методов, которые могут быть использованы для разработки собственного алгоритма определения лицевых движений в рамках подготовки к написанию дипломной работы. К таким методам относятся алгоритмы детекции лица, определения на нем ключевых точек, методы классификации и композиции классификаторов.

Целью производственной практики является осуществление профессионально-практической подготовки студентов к будущей профессиональной деятельности, закрепление и углубление теоретических знаний в области проектирования информационных систем, овладение умениями и навыками воспринимать математические, естественнонаучные, профессиональные знания и применять их для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте.

Задачи проведения практики:

- ввести систему кодирования лицевых движений;
- проанализировать существующие методы определения лицевых движений;
- произвести поиск и сбор информации о методах детектирования лица и его ключевых точек;
- проанализировать методы классификации и композиции методов машинного обучения.

Вид практики – производственная.

Способы проведения практики – стационарная (МГТУ им. Н.Э. Баумана)

Форма проведения – распределенная – проходит в течение семестра.

Тип практики – научно-исследовательская работа: практика по получению профессиональных умений и опыта профессиональной деятельности.

ОБЗОР МЕТОДОВ

1. Система кодирования лицевых движений

В качестве системы кодирования лицевых движений использована система, изначально разработанная Экманом и Фризенем в 1978 году, так как она детально описывает основные лицевые движения [1] В таблице 1 представлена классификация лицевых движений.

Таблица 1. Классификация лицевых значений.

№	Значение	№	Значение	№	Значение
0	Нейтральное лицо	16	Опущенная нижняя губа	31	Сжатые челюсти
1	Поднятые внутренние части бровей	17	Поднятый подбородок	32	Покусывание губы
2	Поднятые внешние части бровей	18	Сморщенные губы	33	Выдувание
4	Опущенные брови	19	Показ языка	34	Раздувание щёк
5	Поднятое верхнее веко	20	Растянутые губы	35	Втягивание щёк
6	Поднятые щеки	21	Натянутая шея	36	Язык высунут

7	Натянутое веко	22	Губы воронкой	37	Облизывание губ
8	Губы навстречу друг другу	23	Натянутые губы	38	Расширенные ноздри
9	Сморщенный нос	24	Сжатые губы	39	Узкие ноздри
10	Поднятая верхняя губа	25	Губы разведены	41	Опущенное надпереносье
11	Углубленная носогубная складка	26	Челюсть опущена	43	Глаза закрыты
12	Поднятые уголки губ	27	Рот широко открыт	45	Моргание
13	Остро поднятые уголки губы	28	Втянутые губы		
14	Ямочка на щеке	29	Нижняя челюсть вперёд		
15	Опущенные уголки губ	30	Челюсть в бок		

На рис. 1 показаны примеры некоторых лицевых движений на фото.































AU 1	AU 2	AU 4	AU 5	AU 6	AU 7
					
Поднятые внутренние части бровей	Поднятые внешние части бровей	Опущенные брови	Поднятые внутренние части бровей	Поднятые щеки	Натянутое веко
*AU 41	*AU 42	*AU 43	AU 44	AU 45	AU 46
					
Опущенное верхнее веко	Узкий разрез глаз	Глаза закрыты	Прищур	Моргание	Подмигивание
AU 9	AU 10	AU 11	AU 12	AU 13	AU 14
					
Сморщенный нос	Поднятая верхняя губа	Углубленная носогубная складка	Поднятые уголки губ	Остро поднятые уголки губ	Ямочка на щеке
AU 15	AU 16	AU 17	AU 18	AU 20	AU 22
					
Опущенные уголки губ	Опущенная нижняя губа	Поднятый подбородок	Сморщенные губы	Растянутые губы	Губы воронкой
AU 23	AU 24	*AU 25	*AU 26	*AU 27	AU 28
					
Натянутые губы	Сжатые губы	Губы разведены	Челюсть опущена	Рот широко открыт	Вытянутые губы

Рис. 1. Примеры лицевых движений

2. Анализ алгоритмов определения лицевых движений человека на изображении

В этом разделе будут рассмотрены методы определения лицевых движений.

2.1 Метод основан на обработке соседних кадров видеопотока

Метод предложен в 2014 году [2]. В его основе лежит анализ соседних кадров видео, по ним делается вывод об изменении лицевых движений, рис. 2.



Рис. 2 Принцип обработки кадров

Для каждого лицевого движения используется бинарный классификатор SVM[3], на вход которому поступают представления лиц соседних кадров. Классификатор предсказывает одно число: -1 - если лицевое движение начало исчезать, 0 - если осталось неизменным, 1 - если начало появляться. В получения вектора представления лица использовался метод CLM-Z[4].

2.2 Метод основанный на нейронной сети JAA-Net

Метод предложен Шао и Лио в 2018 г. [5] Основу метода составляют четыре модуля: иерархическое и многомасштабное обучение, модуль для получения ключевых точек лица, глобальное обучение представлений и адаптивное обучение внимания, рис. 3. Иерархическое и многомасштабное обучение разработано как основа JAA-Net, оно извлекает характеристики каждого локального региона изображения в разных масштабах. Модуль, получающий лицевые точки, помимо генерации лэндмарков используется для генерации начальных карт внимания, которые используются для обнаружения лицевых движений. Модуль глобального обучения представлений призван охватить особенности структуры и текстуры всего лица. Адаптивное обучение внимания разработано как центральная часть для обнаружения лицевых движений, этот модуль изучает карту внимания каждого лицевого движения.

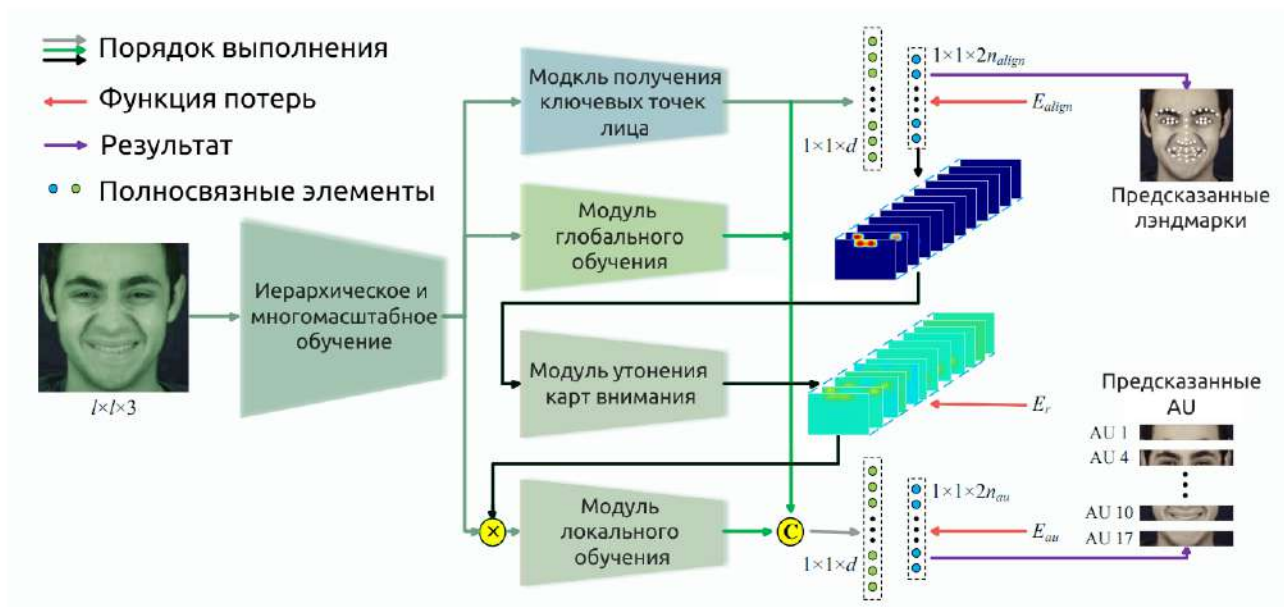


Рис. 3 Архитектура JAA-Net

Вместо использования простых сверточных слоев, авторы предлагают использовать иерархический и многомасштабный слой для изучения

особенностей каждого локального региона с различными масштабами. Один блок иерархического многомасштабного слоя содержит один сверточный слой и еще три иерархических слоя с различными размерами.

На рис. 4 показана архитектура адаптивного обучения внимания. Он состоит из двух этапов: уточнение регионов внимания для лицевых движений и обучение локальным представлениям, где первый шаг заключается в уточнении карты внимания определенного лицевого движения, а второй шаг - в изучении и извлечении локальных представлений лицевых движений. Входами и выходами этапа уточнения внимания являются начальные и уточненные карты внимания, соответственно.

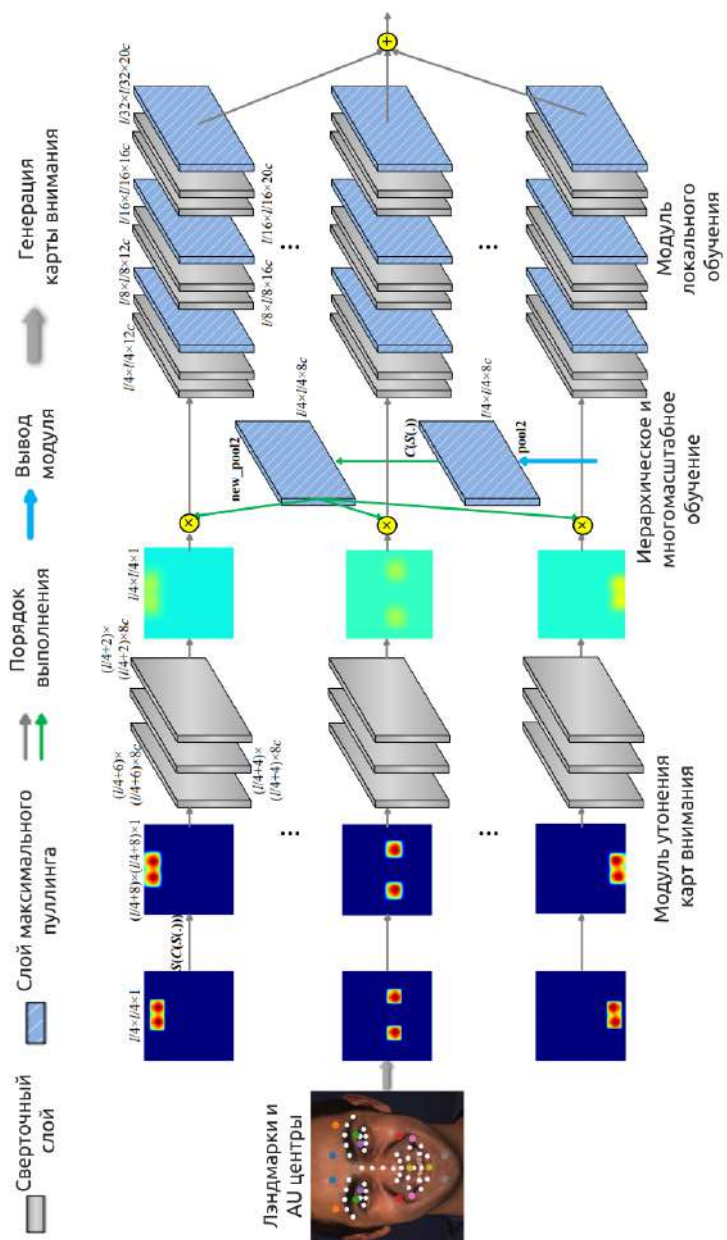


Рис. 4 Архитектура адаптивного обучения внимания

Модуль получения координат лицевых точек, модули обучения глобальным представлениям и адаптивного обучения внимания объединяются вместе и подаются в сеть из двух полносвязанных слоев. Таким образом, положения лицевых точек, глобальные лицевые представления и локальные представления лицевых движений объединяются вместе. Слой softmax используется для прогнозирования вероятности появления каждого лицевого движения.

2.3 Метод основанный на нейронной сети ARL

Метод предложен Шао и Лио в 2019 г.[6]. Рис. 5 иллюстрирует архитектуру структуры ARL, которая состоит из трех частей: иерархического и многомасштабного обучения, обучения пространственному вниманию на уровне каналов и обучения выявлению связей на уровне пикселей. Модуль иерархического и многомасштабного обучения областей направлен на извлечение многомасштабных представлений для адаптации к различным лицевым движениям с различными размерами.

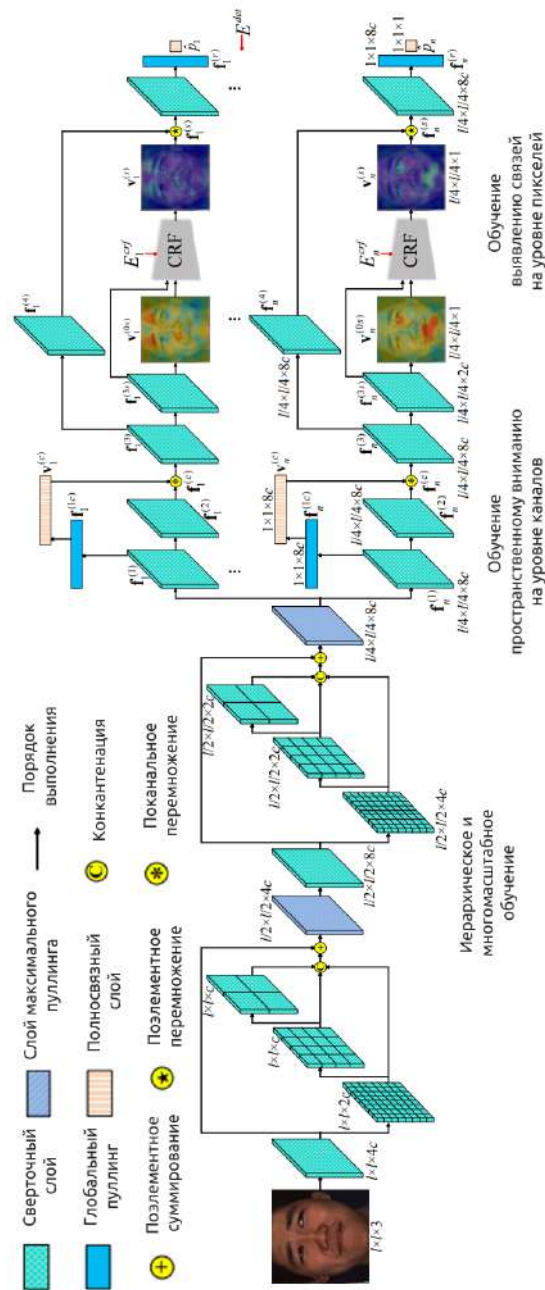


Рис. 5 Архитектура ARL

Для каждого лицевого движения применяется операция свертки на выходе модуля иерархического и многомасштабного обучения, чтобы извлечь представление $f(1)$, рис. 5. Канальный признак $f(1c)$ и представление $f(2)$ генерируются путем выполнения глобального пуллинга по среднему значению и свертки для $f(1)$ соответственно. Вес внимания по каналу $v(c)$ вычисляется как

$$v_i^{(c)} = \sigma(W_i^{(1c)T} f_i^{(1c)}),$$

где c - гиперпараметр, W обозначает весовые параметры, а $\sigma(x) = 1 / (1 + \exp(-x))$ является сигмоидная функция. Далее полученный вес поканально умножается на $f(2)$: $f(c) = v(c) * f(2)$. На этапе обучения выявлению связей на уровне пикселей с помощью условного случайного поля (CRF) происходит уточнение найденных ранее пространственных весов внимания. В конце, для каждого лицевого движения происходит бинарная классификация его наличия на фото.

2.4 Метод основанный на нейронной сети DRML

Метод предложен Шао и Чу в 2016 г. [7]. На рис. 6 показана архитектура DRML. Входными данными является изображение лица в RGB формате, которое затем передается на сверточный слой (conv1) с 32 фильтрами размером $11 \times 11 \times 3$. Слой conv1 генерирует 32 карты объектов, которые подаются в слой областей (region2). Далее следует слой пуллинга по максимальному значению (pool3) 2×2 с шагом 2, отдельно для каждого канала карт объектов. Слой pool3 делает сеть более устойчивой к небольшим ошибкам, вызванным выравниванием лица. В DRML используется только один слой пуллинга по максимальному значению, чтобы не потерять слишком много пространственной информации. За слоем pool3 следуют еще четыре сверточных и два полносвязных слоя.

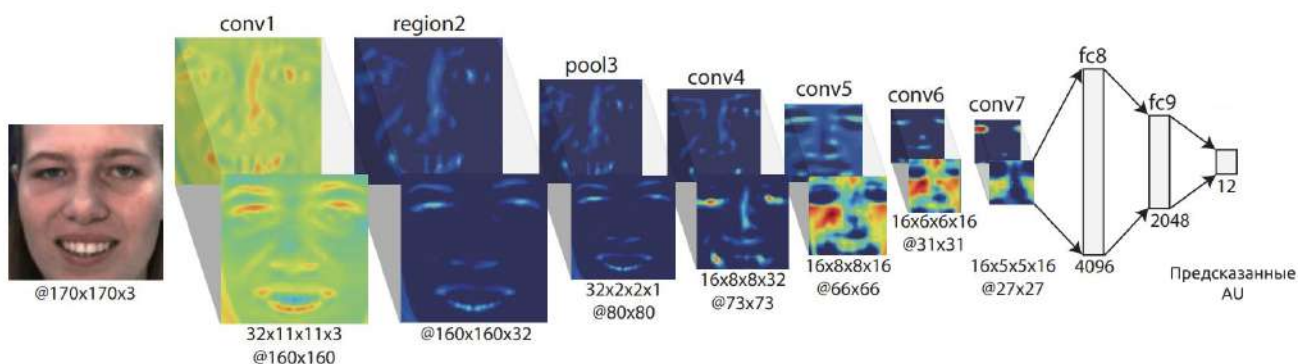


Рис. 6 Архитектура DRML

В сети применяется сигмоидальная кросс-энтропийная функция потерь:

$$-\sum t_i \log(f(s_i)) = -t_1 \log(f(s_1)) - (1 - t_1) \log(1 - f(s_1))$$

Одним из важнейших аспектов DRML является использование слоя области, который фиксирует локальные изменения для разных лицевых областей.

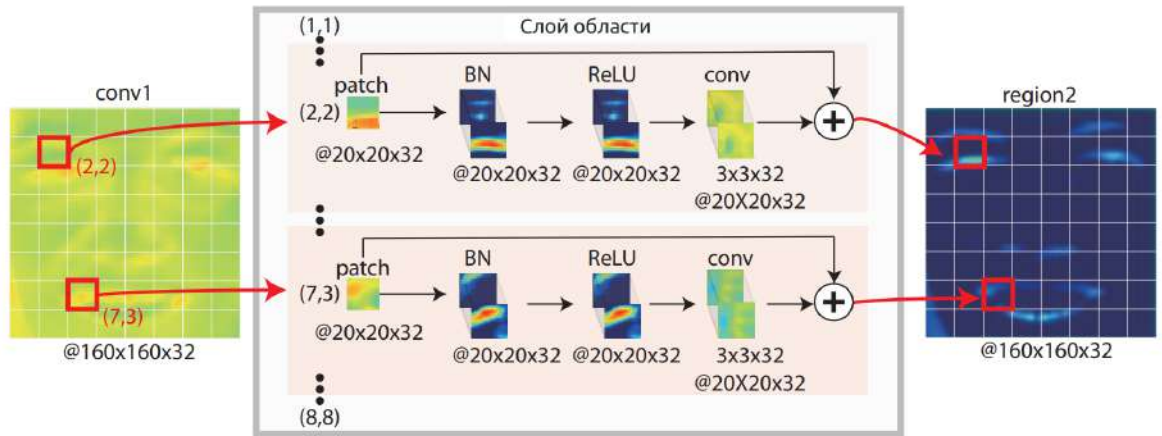


Рис 7. Слой области

На рис. 7 показан предлагаемый слой области (region2), который содержит три компонента: отсечение патчей, локальная свертка и пропуск соединения (skip connection). Компонент отсечения патчей равномерно нарезает карту 160×160 в сетку 8×8 . Каждая мини-партия нормализуется с помощью пакетная нормализации (BN) и проходит через функцию активации ReLU. Компонент локальной свертки учится фиксировать локальные изменения, заставляя веса в каждом патче обновляться независимо. Пропуск соединения помогает избежать проблем исчезновения градиента во время обучения сети.

2.5 Сравнение методов определения лицевых движений человека на изображении

При сравнении методов брались усредненные значения F1 меры полученные на тестовой части датасета DISFA составляющей 20 тысяч изображений. Сравнения рассмотренных методов приведены в таблице 2.

Таблица 2. Достоинства и недостатки методов определения мимики человека.

Название метода	Преимущества	Недостатки
метод основан на обработке соседних кадров видеопотока	+ простота реализации	- работает только для видео или последовательности изображений
метод на основе нейронной сети JAA-Net	+ высокая точность работы (средняя F1 мера по всем AU = 56.0) + определение положения ключевых точек[5]	- сложная реализация - низкая скорость работы
метод на основе нейронной сети ARL	+ точность работы выше, чем у остальных рассмотренных методов (средняя F1 мера по всем AU = 58.7) [6]	- низкая скорость работы
метод на основе нейронной сети DRML	+ высокая скорость работы	- точность работы меньше, чем у остальных рассмотренных методов

		(средняя F1 мера по всем AU = 26.7) [7]
--	--	---

Для распознавания эмоций и мимики на фотографии в первую очередь необходимо определить на ней положение лица человека, которое будет использоваться для последующей обработки. Лицевые движения формируются частями лица такими как губы, брови, глаза и тд. Их форму и положение показывают ключевые точки лица - лэндмарки. Опираясь на положение лэндмарков, можно сформировать вектор признаков для каждого AU, на основе которого сделать вывод о наличии лицевых движений с помощью одного из классификаторов.

3. Анализ алгоритмов определения положения лица на фото

В этом разделе рассматриваются методы детекции лица.

3.1 Алгоритм на основе гистограммы направленных градиентов (HOG)

В основе алгоритма лежит предположение, что форма объекта на участке изображения может быть описана распределением градиентов интенсивности или направлением краев. Реализация дескрипторов может быть произведена с помощью деления изображения на ячейки, и расчетом для каждой ячейки гистограммы направлений градиентов интенсивности или направлений краев для пикселей, находящихся внутри ячейки. Комбинация этих гистограмм является дескриптором. Для увеличения точности локальные гистограммы подвергаются нормализации по контрасту.

Для тренировки системы определения положения лиц используются два набора изображений. В первом содержатся изображения лиц при различном наклоне и освещении, во втором лица отсутствуют. Для каждого набора находятся гистограммы направленных градиентов. Вычисление градиента осуществляется для каждого пикселя изображения с помощью маски. Наиболее оптимальной маской для данного алгоритма является одномерная

дифференцирующая маска, этот метод требует фильтрации яркостной составляющей при помощи фильтрующих ядер [8]:

$[-1, 0, 1], [-1, 0, 1]^T$. Затем изображение делится на ячейки фиксированного размера. В каждой ячейке производится расчет преобладающего направления градиента путем анализа значения градиента каждого пикселя ячейки. Каналы гистограммы равномерно распределяются от 0 до 180 или же от 0 до 360°, в зависимости от того, вычисляется «знаковый» или «беззнаковый градиент». Полученные ячейки группируются в более крупные связанные блоки для нормирования значения градиентов. Это позволяет учитывать яркость и контрастность каждой ячейки. Deskriptor HOG, таким образом, является вектором компонент нормированных гистограмм ячеек из всех областей блока. Затем происходит классификация полученных дескрипторов при помощи системы обучения с учителем. Далал и Триггс в своей работе использовали метод опорных векторов [8].

3.2 Метод Виолы-Джонса

Метод позволяет обнаруживать объекты на изображениях в реальном времени. Его предложили Паул Виола и Майкл Джонс в 2001 году [9]. В алгоритме используются изображения в интегральном представлении, то есть в виде матрицы, совпадающей по размерам с исходным изображением. В каждом элементе матрицы хранится сумма интенсивностей всех пикселей, находящихся левее и выше данного элемента. Основу метода составляют признаки Хаара, с помощью которых происходит поиск объекта на фотографии. В стандартном методе Виолы – Джонса используются прямоугольные признаки, изображенные на рис. 8, они называются примитивами Хаара.

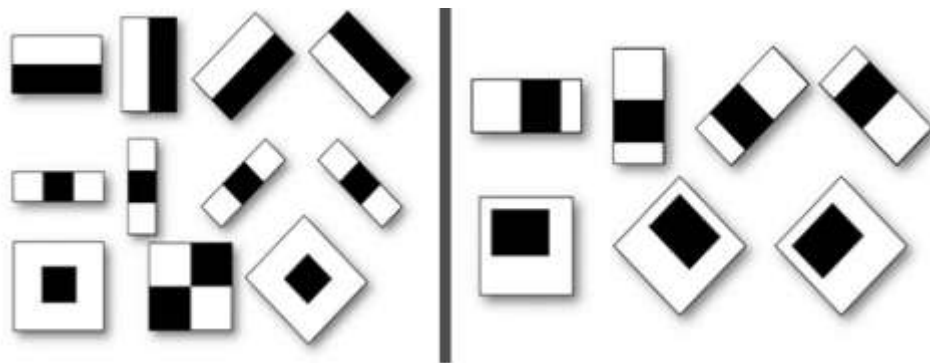


Рис. 8. Примитивы Хаара

Результатом применения такого примитива к интегральному изображению будет разность суммы значений яркостей точек, закрываемых светлой частью признака, и суммы значений яркостей точек, закрываемых темной частью признака. Признаки Хаара дают точечное значение перепада яркости по оси x и y . В методе Виолы - Джонса применяются методика скользящего окна. Окно начинает последовательно двигаться по изображению с шагом в 1 ячейку. При сканировании изображения в каждом окне вычисляется приблизительно 200 000 вариантов расположения признаков, сканирование производится последовательно для различных масштабов, масштабируется не само изображение, а сканирующее окно. Все найденные признаки попадают к классификатору. Для того, чтобы из большого количества признаков выбрать только важные, авторы метода использовали алгоритм Adaboost, основанный на бустинге. Бустинг - это процедура последовательного построения композиции алгоритмов машинного обучения, когда каждый следующий алгоритм стремится компенсировать недостатки композиции всех предыдущих. Бустинг помимо непосредственно классификации позволяет определить признаки, наиболее влияющие на результат классификации, что значительно сокращает их количество. Так как большая часть изображения не имеет детектируемых образов, применять все классификаторы к окну затратно по времени, поэтому Виолла и Джонс ввели понятие каскада классификаторов. Вместо применения

всех классификаторов, их применяют поэтапно и, если окно не проходит очередной этап, алгоритм переходит к поиску объекта в следующем окне.

3.3 Метод на основе сверточных нейронных сетей

Сверточная нейронная сеть - специальная архитектура нейронных сетей, предложенная Яном Лекуном [10], изначально нацеленная на эффективное распознавание изображений. В сверточной нейронной сети в операции свертки используется ограниченная матрица весов небольшого размера, которая сдвигается по всему обрабатываемому слою, формируя после каждого сдвига сигнал активации для нейрона следующего слоя с аналогичной позицией, путем скалярного умножения матрицы весов на часть обрабатываемого слоя того же размера. Пример операции свертки показан на рис. 9.

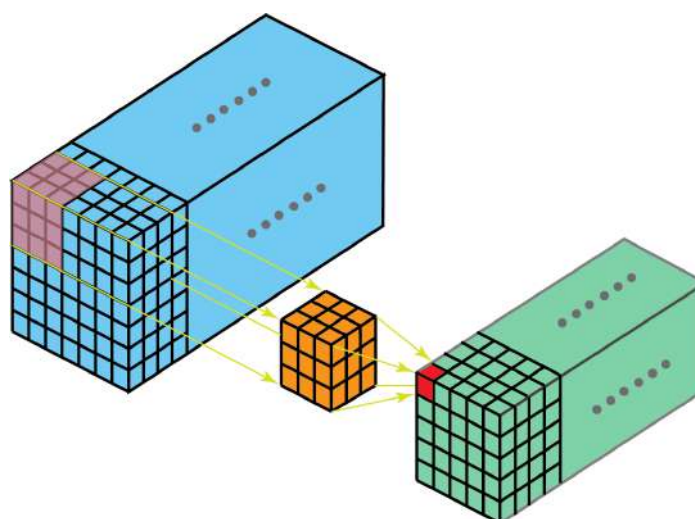


Рис. 9. Пример операции свертки

То есть для различных частей слоя используются одна и та же матрица весов, которую также называют ядром свертки. При этом такие ядра свертки не закладываются исследователем заранее, а формируются самостоятельно путём обучения сети классическим методом обратного распространения ошибки. Проход каждым набором весов формирует свой собственный экземпляр карты признаков, делая нейронную сеть многоканальной.

Операция субдискретизации, или пуллинга, выполняет уменьшение размерности сформированных карт признаков. Исходное изображение делится на блоки и для каждого блока вычисляется некоторая функция. Чаще всего используется функция максимума или (взвешенного) среднего. Пример операции пуллинга представлен на рис. 10. Обучаемых параметров у этого слоя нет.

Основные цели пуллингового слоя:

- уменьшение изображения, чтобы последующие свертки оперировали над большей областью исходного изображения;
- увеличение инвариантности выхода сети по отношению к малому переносу входа;
- ускорение вычислений.[11]



Рис. 10 Пример операции пуллинга по максимальному значению

Скалярный результат каждой свертки попадает на функцию активации, которая представляет собой некую нелинейную функцию. Наиболее часто в сверточных сетях используется ReLU (сокращение от англ. rectified linear unit) - $f(x)=\max(0,x)$. Это операция отсечения отрицательной части скалярной величины. По состоянию на 2017 год эта функция и её модификации (Noisy ReLU, Leaky ReLU и другие) являются наиболее часто используемыми функциями активации в глубоких нейросетях [12].

Существует множество архитектур сверточных сетей, которые можно использовать для детекции лица человека. Одной из них является архитектура YOLO, представленная на рис. 11. Она имеет 24 сверточных слоя и два полносвязных слоя. Также в ней применяется пулинг по максимальному значению.

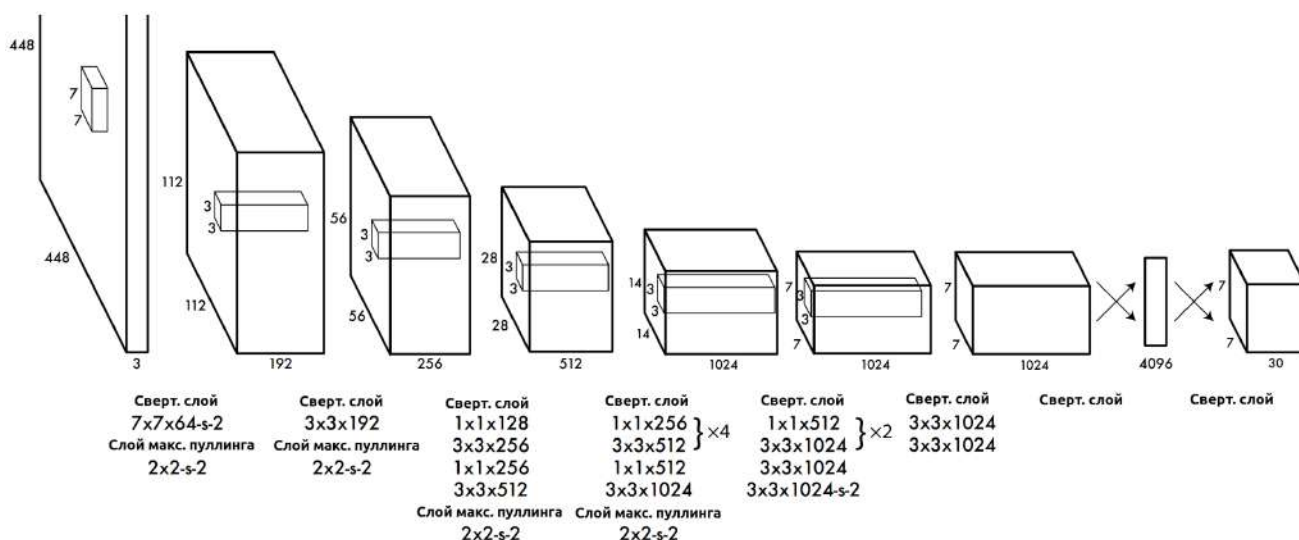


Рис. 11 Архитектура YOLO

YOLO просматривает изображение один раз. Она делит его на квадраты и для каждого квадрата определяет вероятность нахождения в нем искомого объекта, а также его координаты, ширину и высоту. Для каждого квадрата изображения YOLO может обнаружить максимум пять искомых объектов. В функцию потерь для YOLO стандартно входит ошибка локализации, ошибка уверенности в том, что искомый объект присутствует и ошибка классификации. Последняя ошибка в случае определения положения лиц не учитывается, так как система определяет один объект - лицо человека. Значения весов рассчитываются методом обратного распространения ошибки.

3.4 Сравнение методов определения положения лица на фото

Достоинства и недостатки рассмотренных методов приведены в таблице 3.

Таблица 3. Достоинства и недостатки методов определения положения лица.

Название метода	Преимущества	Недостатки
метод на основе HOG	+ высокая скорость работы на CPU + хорошая точность при незначительных наклонах головы и небольших перекрытиях объектов [14]	- неточная работа при значительных перекрытиях и поворотах головы [14]
метод Виолы-Джонса	+ высокая скорость работы на CPU + детекция лиц различных масштабов	- неточная работа при наклонах и поворотах головы и небольших перекрытиях [14]
метод на основе сверточных нейронных сетей	+ устойчивость к перекрытиям [14]	- медленная работа на CPU

4. Анализ алгоритмов определения положения ключевых точек лица

В этом разделе рассматриваются методы определения положения ключевых точек лица.

4.1 Алгоритм на основе бустинга деревьев решений

Метод предложен Каземи и Сулливан в 2014 году [15]. На основе тренировочного множества изображений и соответствующих им ключевых точек лица - лэндмарков вычисляется среднее положение лэндмарков, оно будет

использоваться как начальное приближение искомым ключевых точек лица. Для уточнения положения лэндмарков авторы использовали градиентный бустинг деревьев решений - это техника машинного обучения для задач классификации и регрессии, которая строит модель предсказания в форме ансамбля деревьев. В качестве признаков были использованы значения интенсивности областей пикселей изображения, соответствующие координатам лэндмарков. В бустинге применялась квадратичная ошибка. Таким образом, алгоритм Каземи и Сулливан принимает изображение лица человека и вычисляет 194 пар координат точек лица на изображении. Пример работы алгоритма показан на рис. 12, на нем Т - этап уточнения координат ключевых точек.

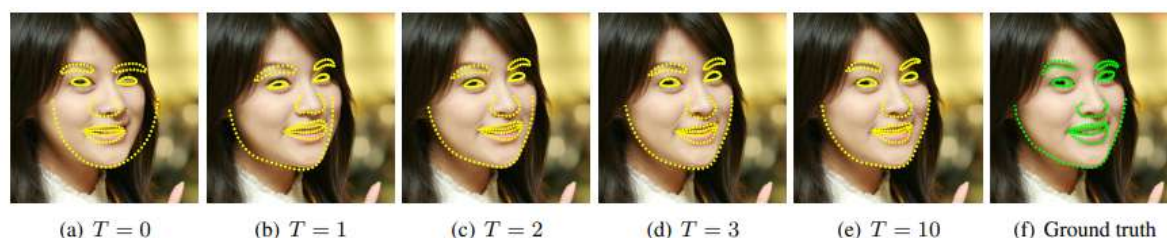


Рис. 12 Процесс уточнения положения лэндмарков с помощью бустинга

4.2 Алгоритмы на основе сверточной нейронной сети

В основе данных методов лежит следующий принцип: из слоев свертки, пуллинга, пакетной нормализации, функций активации формируется нейронная сеть. Функцией потерь в ней является квадратичная ошибка между полученными координатами лицевых точек и их истинными значениями. Веса рассчитываются итеративно с помощью обратного распространения ошибки.

4.2.1 Метод FAN

Метод [31] предложен Эдрианом Булатом и Георгисом Цимиropулосом в 2017 г. В основе метода лежит сверточная сеть HG [32], используемая для предсказания позы человека. В HG используется сеть в виде песочных часов, рис. 13.

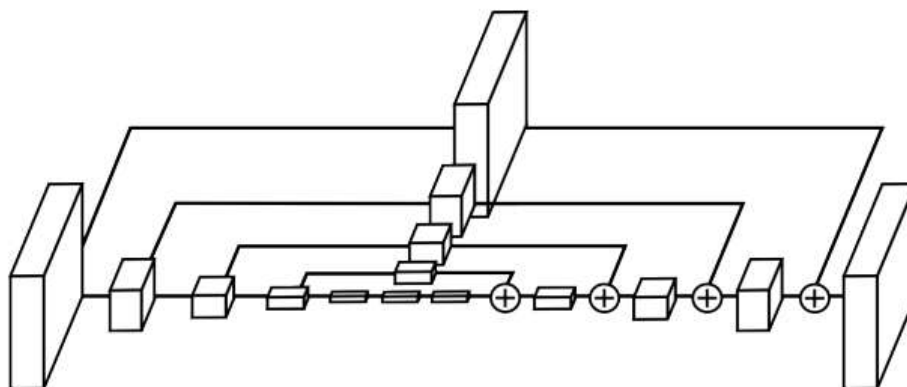


Рис. 13 Блок сети HG

Сверточные слои и слои максимального пуллинга используются для преобразования входной матрицы изображения к матрице низкого разрешения. После достижения самого низкого разрешения сеть начинает восходящую последовательность с апсемплингом, при этом комбинируя матрицы разных масштабов.

В методе FAN используется последовательность из 4 HG сетей, рис. 14.

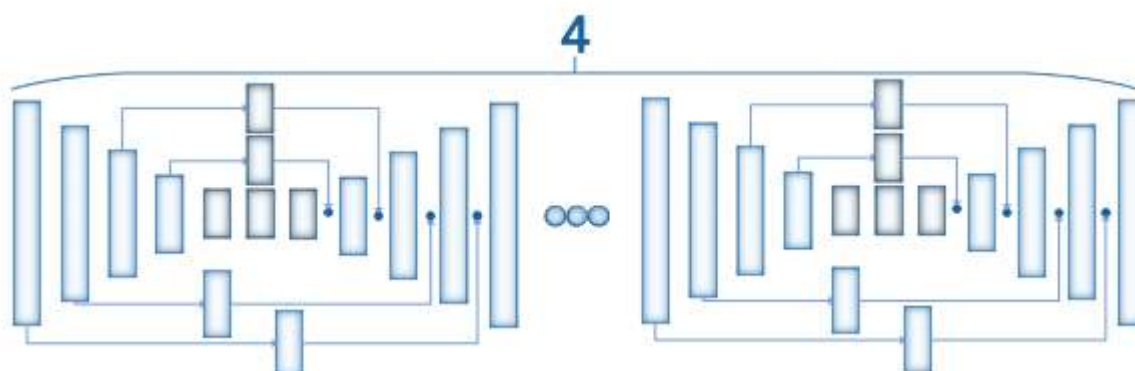


Рис. 14 Сеть FAN

Сеть предсказывает тепловую карту ключевых точек лица. Авторы обучали сеть предсказывать как двухмерные, так трехмерные ключевые точки.

4.2.2 Метод на основе двухэтапной сети

Метод [33] предложен Эдрианом Булатом и Георгисом Цимиропулосом в 2016 г.

Данный метод предлагает разделить нахождение 3d точек лица на два этапа. На первом оцениваются X и Y координаты ключевых точек лица с помощью сети ResNet-152 [34], и предсказываются тепловые карты [37] для каждой ключевой точки. На втором шаге оценивается Z координата точки, на вход сети второго этапа подаются полученные тепловые карты. Таким образом, сеть предсказывает как двухмерные, так трехмерные ключевые точки.

4.2.3 Метод на основе выравнивания лица и 3DMM

Метод [35] предложен Сонгджанг Ли и Хонгаи Ли в 2019 г. Он состоит из трех частей. Для начала сверточной нейронной сетью предсказывается положение 2d ключевых точек, затем к изображению применяются трансформации поворота, масштабирования так, чтобы все лица в датасете имели одинаковую форму, затем применяется еще одна сверточная сеть, которая находит ключевые точки на новом изображении, далее ключевые точки уточняются методом 3DMM [36], который также преобразует двухмерные точки в трехмерные и позволяет строить трехмерную поверхность лица.

4.3 Сравнение методов определения положения ключевых точек лица

Достоинства и недостатки рассмотренных методов приведены в таблице 4.

Таблица 4. Достоинства и недостатки методов определения ключевых точек лица.

Название метода	Преимущества	Недостатки
метод на основе бустинга деревьев решений	+ высокая скорость работы на CPU[15]	- неточная работа при сильных наклонах и поворотах головы[15]
Метод FAN	+ получение двухмерных и трехмерных ключевых точек	- не позволяет получить трехмерную поверхность лица [31]

Метод на основе двухэтапной сети	+ получение двухмерных и трехмерных ключевых точек + простота реализации	- Работает хуже на датасете 300VW-A (AUC = 65.1), чем метод FAN (AUC = 72.1) [31]
Метод на основе выравнивания лица и 3DMM	+ получение двухмерных и трехмерных ключевых точек + трехмерная поверхность лица	- сложность реализации

5. Анализ алгоритмов классификации.

В этом разделе рассматриваются методы классификации.

5.1 Логистическая регрессия

Логистическая регрессия - метод построения линейного классификатора, позволяющий оценивать апостериорные вероятности принадлежности объектов классам [18]. Алгоритм применяется для прогнозирования вероятности возникновения некоторого события по значениям множества признаков. Для этого вводится зависимая переменная y , принимающая значения 0 и 1 и множество независимых переменных x_1, \dots, x_n на основе значений которых требуется вычислить вероятность принятия того или иного значения зависимой переменной.

В логистической регрессии строится линейный алгоритм классификации вида:

$$a(x, w) = \text{sign} \left(\sum_{j=1}^n w_j f_j(x) - w_0 \right) = \text{sign} \langle x, w \rangle$$

где w_j – вес j -го признака, w_0 – порог принятия решения, $w=(w_0, \dots, w_n)$ – вектор весов, $\langle x, w \rangle$ – скалярное произведение признакового описания объекта на вектор весов.

Задача обучения линейного классификатора заключается в том, чтобы по выборке настроить вектор весов w . В логистической регрессии для этого решается задача минимизации эмпирического риска с функцией потерь специального вида:

$$Q(w) = \sum_{i=1}^m \ln(1 + \exp(-y_i \langle x_i, w \rangle)) \rightarrow \min_w$$

Задача минимизации может быть решена аналитически с помощью метода наименьших квадратов [19] (медленного, но точного метода), а также итеративно с помощью алгоритма градиентного спуска [20] (быстрого, но менее точного метода).

5.2 Метод опорных векторов

Основная идея метода заключается в построении гиперплоскости, разделяющей объекты выборки оптимальным способом [21]. Алгоритм работает в предположении, что чем больше расстояние между разделяющей гиперплоскостью и объектами разделяемых классов, тем меньше будет средняя ошибка классификатора.

Важнейшим понятием метода является отступ — характеристика, оценивающая, насколько типичным представителем класса является объект. Чем меньше значение отступа, тем ближе объект подходит к границе классов и тем выше становится вероятность ошибки. Отступ представляет из себя следующую формулу:

$$M_i(\vec{w}, b) = y_i(\langle \vec{w}, \vec{x}_i \rangle - b),$$

где \vec{x}_i – вектор признаков, \vec{w} и b – параметры гиперплоскости. y_i – целевой класс

В случае линейно разделимой выборки метод сводится к задаче оптимизации в терминах квадратичного программирования:

$$\begin{cases} \|\vec{w}\|^2 \rightarrow \min_{w,b} \\ M_i(\vec{w}, b) \geq 1, \quad i = 1, \dots, \ell \end{cases}$$

Она решается с помощью множителей Лагранжа

На практике линейно разделимые выборки практически не встречаются: в данных возможны выбросы и нечеткие границы между классами. В таком случае поставленная выше задача не имеет решений. В этом случае для каждого объекта отнимают от отступа некоторую положительную величину ξ_i , но требуют чтобы эти введенные поправки были минимальны. Это приведёт к следующей постановке задачи, называемой также SVM с мягким отступом:

$$\begin{cases} \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^{\ell} \xi_i \rightarrow \min_{w,b,\xi} \\ M_i(\vec{w}, b) \geq 1 - \xi_i, \quad i = 1, \dots, \ell \\ \xi_i \geq 0, \quad i = 1, \dots, \ell \end{cases}$$

Ей эквивалентна следующая задача безусловной минимизации:

$$\frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^{\ell} (1 - M_i(\vec{w}, b))_+ \rightarrow \min_{w,b},$$

где C - параметр подбираемый в процессе кросс-валидации [22].

Одним из методов ее решения является метод градиентного спуска.

5.3 Дерево решений

Дерево решений — алгоритм $a(x) = (V_{\text{внутр}}, v_0, V_{\text{лист}}, S_v, \beta_v)$,

задающийся деревом, где:

$V = V_{\text{внутр}} \cup V_{\text{лист}}$ — множество вершин, $v_0 \in V$ — корень дерева;

$S_v: D_v \rightarrow V_v$ — функция перехода по значению предиката в множество

детей вершины v ;

$\beta_v: X \rightarrow D_v$ — предикат ветвления, $v \in V_{\text{внутр}}$ и $|D_v| < \infty$;

Для листьев $v \in V_{\text{лист}}$ определена метка класса $y_v \in Y$ [23].

Это логический алгоритм классификации. Он заключается в последовательном дроблении выборки на две части до тех пор, пока в каждой части не окажутся объекты только одного класса. Разделение производится по предикату β , который выбирается из множества элементарных предикатов таким образом, чтобы максимизировать информационный выигрыш:

$$Gain(\beta, U) = \Phi(U) - \Phi(U_1, \dots, U_{|D_v|}) = \Phi(U) - \sum_{k \in D_v} \frac{|U_k|}{|U|} \Phi(U_k) \rightarrow \max_{\beta \in B},$$

где U - разбиение, а $\Phi(U)$ - мера неопределенности.

Примерами мер неопределенности распределения являются:

- Энтропия:

$$\Phi(U) = - \sum_i^N p_i \log_2 p_i,$$

где N — количество возможных классов, и p_i — вероятность объекта принадлежать i -ому классу.

- Критерий Джини:

$$\Phi(U) = \sum_{i \neq j} p_i p_j = \sum_i p_i * (1 - p_i),$$

максимизацию этого критерия можно интерпретировать как максимизацию числа пар объектов одного класса, оказавшихся после разбиения в одном множестве.

5.4 Нейронная сеть

Полносвязная нейронная сеть представляет собой последовательность узлов (нейронов), объединенных в слои [24]. Нейроны — единицы, получающие и передающие информацию, каждый из них имеет свой вес. Вектор признаков умножается на вектор весов слоя, полученное значение передается функции активации.

Функции активации определяют какая информация будет передаваться последующим слоям. Нелинейные функции активации позволяют сетям решать нетривиальные задачи с использованием малого числа узлов. На рис. 15 представлены основные функции активации.

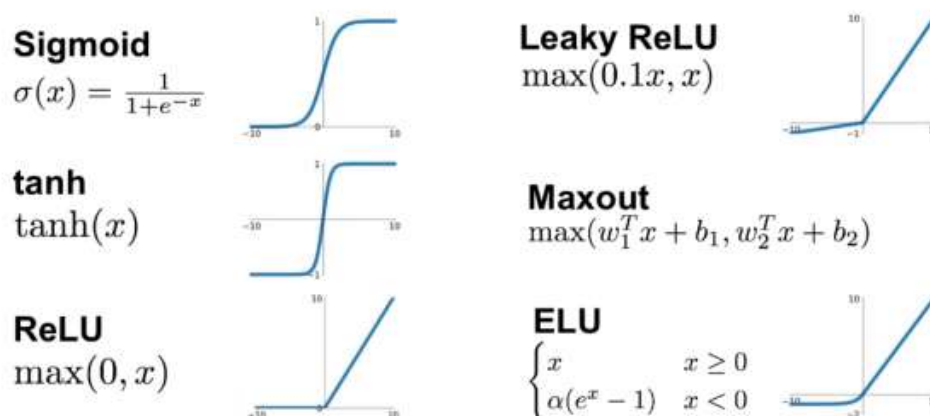


Рис. 15. Функции активации

Нейронная сеть может состоять из большого количества слоев, пример двухслойной сети представлен на рис. 16.

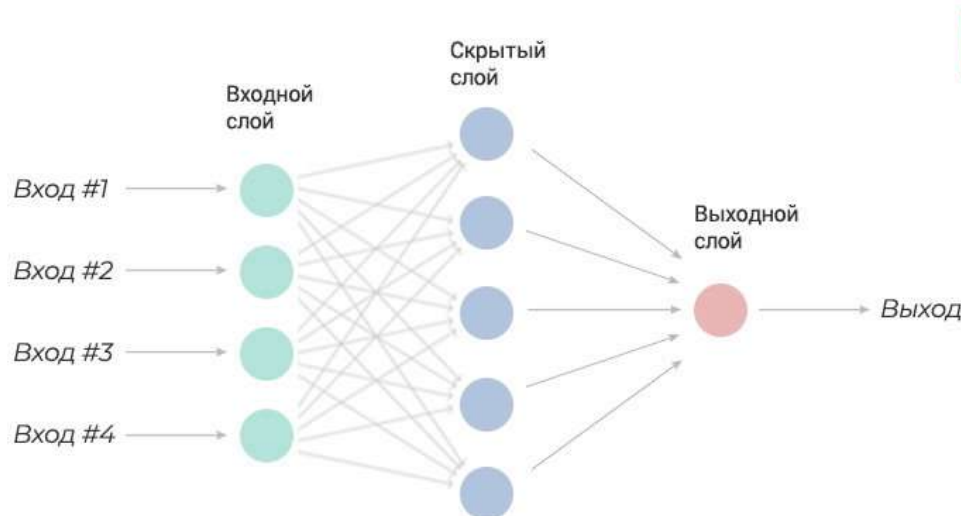


Рис. 16. Двухслойная сеть

Полученное на выходе сети значение, как в случае с логистической регрессией, передается в функцию потерь. Для задачи классификации в качестве функции потерь используют логистическую функцию:

$$\sum_i (-y_i \log a_i - (1 - y_i) \log(1 - a_i)) \rightarrow \min$$

где a_i - предсказанное значение, y_i - истинное значение.

Задача ее минимизации решается с помощью градиентного спуска.

5.5 Сравнение методов классификации

Достоинства и недостатки рассмотренных методов приведены в таблице 5.

Таблица 5. Достоинства и недостатки методов классификации.

Название метода	Преимущества	Недостатки
логистическая регрессия	+ устойчивость к шуму + простота реализации + простые методы регуляризации	- показывает хороший результат только для линейно разделимой выборки[25]
метод опорных векторов	+ принцип оптимальной разделяющей гиперплоскости приводит к	- неустойчивость к шуму - необходимо подбирать константу C при помощи кросс-валидации

	<p>максимизации ширины разделяющей полосы, а следовательно, к более уверенной классификации</p> <p>+при некоторых модификациях подходит для классификации на линейно неразделимой выборке</p>	
дерево решений	<p>+ работа с непрерывными и дискретными признаками</p> <p>+ интерпретируемость результатов</p> <p>+ мультиколлинеарность признаков не влияет на качество модели</p>	- склонность к переобучению
нейронная сеть	<p>+ способна восстанавливать сложные зависимости</p>	<p>- сложность в подборе параметров сети</p> <p>- трудность интерпретации результатов работы</p>

6. Анализ методов композиции алгоритмов машинного обучения.

Композиционные методы — это парадигма машинного обучения, в которой несколько моделей обучаются для решения одной и той же проблемы и объединяются для получения лучших результатов.

В этом разделе рассматриваются методы композиции алгоритмов машинного обучения для классификации.

6.1 Бустинг

Бустинг - это процедура последовательного построения композиции алгоритмов машинного обучения, когда каждый следующий алгоритм стремится компенсировать недостатки композиции всех предыдущих алгоритмов.

Адаптивный бустинг (AdaBoost)[26] - метод, который заключается в концентрации внимания на недообученности алгоритма. Что значит, при каждом новом предсказании алгоритм будет работать над сложными, неочевидными для предсказания подвыборками. После предсказаний каждой модели увеличиваются веса ошибочно классифицированных объектов, чтобы следующая модель лучше сработала на них.

Адаптивный бустинг можно описать следующей последовательностью шагов:

1. Изначально всем точкам присвоены равные веса
2. Модель строится на подвыборке данных.
3. По этой модели получаются предсказания для всех данных.
4. По предсказаниям и истинным значениям вычисляются ошибки.
5. В построении следующей модели наибольшие веса присваиваются точкам данных, на предсказании которых алгоритм ошибся.
6. Веса могут быть определены по величине ошибки. А именно, чем больше ошибка, тем больше вес.
7. Этот процесс повторяется, пока функция ошибки не перестанет меняться или пока не будет достигнуто максимальное число предсказателей.

Градиентный Бустинг [26] отличается от адаптивного тем, что, в отличие от AdaBoost, изменяющего веса при каждой итерации, градиентный бустинг обучает новые модели по остаточной ошибке прошлых, двигаясь к минимуму функции потерь. Для задачи классификации в качестве функции потерь используют логистическую функцию:

$$\sum_i (-y_i \log a_i - (1 - y_i) \log(1 - a_i)) \rightarrow \min$$

где a_i - предсказанное значение, y_i - истинное значение.

Градиентный бустинг можно описать следующей последовательностью шагов:

1. Модель строится по подборке данных.
2. Эта модель делает предсказания для всего набора данных.
3. По предсказаниям и истинным значениям вычисляются ошибки.
4. Новая модель строится с учетом ошибок как целевых переменных. При этом обучение стремится найти лучшее разделение для минимизации ошибки.
5. Предсказания, сделанные с помощью этой новой модели, сочетаются с предсказаниями предыдущих.
6. Снова вычисляются ошибки с использованием этих предсказанных значений и истинных значений.
7. Этот процесс повторяется, пока функция ошибки не перестанет меняться или пока не будет достигнуто максимальное число предсказателей.

Благодаря бустингу можно понизить смещение входящих в него моделей, а разброс либо останется таким же, либо увеличится. Из-за этого, как правило, в бустинге используются неглубокие решающие деревья (3-6 уровней), которые обладают большим смещением, но не склонны к переобучению [27].

6.2 Бэггинг

Бэггинг - это технология классификации, использующая композиции алгоритмов, каждый из которых обучается независимо [28]. Результат классификации определяется путем голосования.

Идея заключается в том, что классификаторы не исправляют ошибки друг друга, в отличие от бустинга, а компенсируют их при голосовании. Базовые классификаторы должны быть независимыми, это могут быть классификаторы основанные на разных группах методов или же обученные на независимых наборах данных.

Окончательное решение о принадлежности объекта классу может приниматься, например, одним из следующих методов:

- консенсус: если все элементарные классификаторы присвоили объекту одну и ту же метку, то относим объект к выбранному классу;
- простое большинство: консенсус достижим очень редко, поэтому чаще всего используют метод простого большинства. Здесь объекту присваивается метка того класса, который определило для него большинство элементарных классификаторов;
- взвешивание классификаторов: если классификаторов четное количество, то голосов может получиться поровну, еще возможно, что одна из групп параметров важна в большей степени, тогда прибегают к взвешиванию классификаторов. То есть при голосовании голос классификатора умножается на его вес.

Бэггинг позволяет снизить процент ошибки классификации в случае, когда высока дисперсия ошибки базового метода.

6.3 Стэкинг

В основе метода лежит обучение нескольких разных слабых классификаторов, а также метамоделей для вывода предсказаний, основанных на множественных предсказаниях, возвращаемых этими слабыми моделями [29]. Обучающую выборку делят на две части. На первой обучают базовые алгоритмы. Затем получают их ответы на второй части и на тестовой выборке. Ответ каждого алгоритма рассматривается как новый признак (метапризнак). На метапризнаках второй части обучения настраивают метаалгоритм. Алгоритм работы стэкинга представлен на рис. 17.

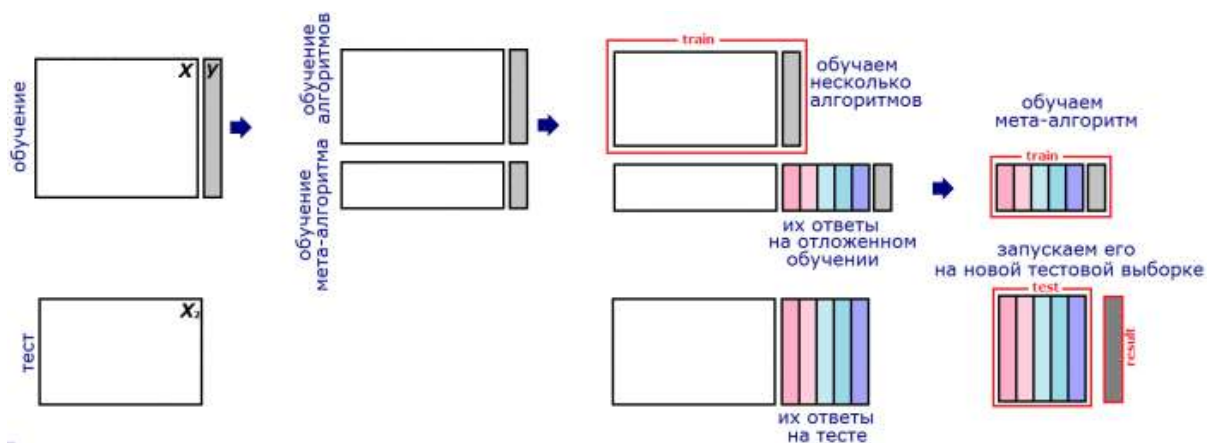


Рис. 17. Алгоритм работы стэкинга

6.4 Сравнение методов композиции алгоритмов машинного обучения

Достоинства и недостатки рассмотренных методов приведены в таблице 6.

Таблица 6. Достоинства и недостатки методов композиции алгоритмов машинного обучения.

Название метода	Преимущества	Недостатки
Бустинг	+ уменьшает смещение модели	- чувствительность в выбросам[30] - неэффективен, если модели модели имеют большой разброс
Бэггинг	+ уменьшает разброс модели	- неэффективен, если модели модели имеют большое смещение
Стэкинг	+ легок для распараллеливания	- сложно подобрать необходимые модели и параметры к ним

ЗАКЛЮЧЕНИЕ

В ходе проведения научно-исследовательской работы в рамках подготовки к написанию дипломной работы была введена система классификации лицевых движений. Были рассмотрены следующие алгоритмы:

- алгоритмы определения положения лица на фото: при высокой вероятности перекрытия другими объектами предпочтительнее использовать метод на основе сверточных нейронных сетей, однако в случае недостатка вычислительной мощности процессора и при незначительных перекрытиях и наклонах головы стоит отдать предпочтение детектору на основе гистограммы направленных градиентов (HOG);
- алгоритмы определения ключевых точек лица: при высокой вероятности наклонов и поворотов головы предпочтительнее использовать метод на основе сверточных нейронных сетей, однако в случае недостатка вычислительной мощности процессора и при незначительных перекрытиях и наклонах головы стоит отдать предпочтение методу на основе бустинга деревьев решений;
- алгоритмы определения мимики человека: метод на основе анализа соседних кадров видеопотока прост в реализации, однако работает только для видео или последовательности изображений. Метод на основе нейронной сети JAA-Net обладает высокой точностью работы, обеспечивает определение положения ключевых точек, однако имеет сложную реализацию и низкую скорость работы. Метод на основе нейронной сети ARL обладает точностью работы выше, чем у остальных рассмотренных методов, однако имеет низкую скорость работы. Метод на

основе нейронной сети DRML обладает высокой скоростью работы, однако имеет точность работы меньше, чем у остальных рассмотренных методов;

- алгоритмы классификации: для восстановления нелинейных зависимостей предпочтительнее использовать дерево решений, нейронную сеть, и при некоторых модификациях метод опорных векторов, однако в случае, если выборка линейно разделима, можно использовать линейную регрессию;
- методы композиции алгоритмов машинного обучения: для уменьшения смещение модели применительно использовать бустинг, но он чувствителен к выбросам, неэффективен, если модели имеют большой разброс. В случае если необходимо уменьшить разброс модели, предпочтительнее использовать бэггинг, однако он неэффективен, если модели имеют большое смещение.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Ekman P. Facial Action Coding System: A Technique for the Measurement of Facial Movement. / Ekman P., Friesen W. // Consulting Psychologists Press - 1978
2. Mahmoud Khademi, Relative Facial Action Unit Detection / Mahmoud Khademi, Louis-Philippe Morency // IEEE Transactions on Affective Computing - 2014
3. SVA [Электронный ресурс] – Режим доступа:
[https://neerc.ifmo.ru/wiki/index.php?title=Метод_опорных_векторов_\(SVM\)](https://neerc.ifmo.ru/wiki/index.php?title=Метод_опорных_векторов_(SVM))
(Дата обращения: 10.05.2021)
4. T. Baltrusaitis, P. Robinson and L.-P. Morency, “3D constrained local model for rigid and non-rigid facial tracking,” Computer Vision and Pattern Recognition (CVPR), 2012.
5. Zhiwen Shao, Deep Adaptive Attention for Joint Facial Action Unit Detection and Face Alignment / Zhiwen Shao, Zhilei Liu, Jianfei Cai, Lizhuang Ma // ECCV - 2018
6. Zhiwen Shao, Facial Action Unit Detection Using Attention and Relation Learning / Zhiwen Shao, Zhilei Liu, Jianfei Cai, Senior Member, Yunsheng Wu, and Lizhuang Ma // IEEE Transactions on Affective Computing - 2019
7. Vítor Albiero, Deep Region and Multi-label Learning for Facial Action Unit Detection / Vítor Albiero, Luciano Silva, Olga Regina Pereira Bellon// IEEE International Conference on Image Processing (ICIP) - 2018
8. Dalal N. Histograms of Oriented Gradients for Human Detection / Dalal N., Triggs B. // IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) – 2005 - С. 2.
9. Viola P., Robust Real-time Object Detection / Viola P., Jones M.// IJCV – 2001 - С. 1, 3
10. Yann LeCun, Gradient-Based Learning Applied to Document Recognition / Yann LeCun, Leon Bottou, Yoshua Bengio, Patrick Haffner // IEEE - 1998
11. Сверточные нейронные сети [Электронный ресурс] – Режим доступа:
http://neerc.ifmo.ru/wiki/index.php?title=Сверточные_нейронные_сети (Дата обращения: 10.05.2021)

12. Сверточные нейронные сети [Электронный ресурс] – Режим доступа:
<https://moluch.ru/conf/tech/archive/324/14834/> (Дата обращения: 10.05.2021)
13. Нейронная сеть YOLO [Электронный ресурс] – Режим доступа:
https://medium.com/@jonathan_hui/real-time-object-detection-with-yoloyolov2-28b1b93e2088 (Дата обращения: 10.05.2021)
14. Методы детекции лица [Электронный ресурс] – Режим доступа:
<https://www.learnopencv.com/face-detection-opencv-dlib-and-deeplearning-c-python/> (Дата обращения: 10.05.2021)
15. Kazemi V. One Millisecond Face Alignment with an Ensemble of Regression Trees / Kazemi V., Sullivan J. // Computer Vision and Pattern Recognition - 2014
16. Ke Sun, High-Resolution Representations for Labeling Pixels and Regions / Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, Jingdong Wang // - 2019
17. Апсемплинг [Электронный ресурс] – Режим доступа:
<http://bigwww.epfl.ch/publications/thevenaz9901.pdf> (Дата обращения: 10.05.2021)
18. Логистическая регрессия [Электронный ресурс] – Режим доступа:
<http://statistica.ru/theory/logisticheskaya-regressiya/> (Дата обращения: 10.05.2021)
19. Логистическая регрессия [Электронный ресурс] – Режим доступа:
http://www.machinelearning.ru/wiki/index.php?title=Логистическая_регрессия (Дата обращения: 10.05.2021)
20. Метод градиентного спуска [Электронный ресурс] – Режим доступа:
<https://neurohive.io/ru/osnovy-data-science/gradient-descent/> (Дата обращения: 10.05.2021)
21. SVA [Электронный ресурс] – Режим доступа:
[https://neerc.ifmo.ru/wiki/index.php?title=Метод_опорных_векторов_\(SVM\)](https://neerc.ifmo.ru/wiki/index.php?title=Метод_опорных_векторов_(SVM)) (Дата обращения: 10.05.2021)
22. Кросс валидация [Электронный ресурс] – Режим доступа:
<https://neerc.ifmo.ru/wiki/index.php?title=Кросс-валидация> (Дата обращения: 10.05.2021)

23. Дерево решений [Электронный ресурс] – Режим доступа:
https://neerc.ifmo.ru/wiki/index.php?title=Дерево_решений_и_случайный_лес
(Дата обращения: 10.05.2021)
24. Нейронная сеть [Электронный ресурс] – Режим доступа:
<https://robohunter.com/news/kak-rabotayt-neironnie-seti-o-slojnoi-sisteme-prostimislovami14200> (Дата обращения: 10.05.2021)
25. Плюсы и минусы алгоритмов машинного обучения [Электронный ресурс] – Режим доступа:
<https://www.hackingnote.com/en/machinelearning/algorithms-pros-and-cons> (Дата обращения: 10.05.2021)
26. Виды бустинга [Электронный ресурс] – Режим доступа:
<https://lambdait.ru/post/busting-s-pomoshchiu-adaboost-i-gradient-boosting> (Дата обращения: 10.05.2021)
27. Соколов Е. А., Курс «Машинное обучение», Лекция 9, Градиентный бустинг / Е. А. Соколов // ФКН ВШЭ - 2017
28. Бэггинг [Электронный ресурс] – Режим доступа:
<http://www.machinelearning.ru/wiki/index.php?title=Бэггинг> (Дата обращения: 10.05.2021)
29. Стэкинг [Электронный ресурс] – Режим доступа:
<https://dyakonov.org/2017/03/10/стекинг-stacking-и-блендинг-blending/> (Дата обращения: 10.05.2021)
30. Бустинг [Электронный ресурс] – Режим доступа:
<https://corporatefinanceinstitute.com/resources/knowledge/other/boosting/> (Дата обращения: 10.05.2021)
31. A. Bulat. How far are we from solving the 2D & 3D Face Alignment problem? (and a dataset of 230,000 3D facial landmarks) // ICCV - 2017
32. A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In ECCV, 2016
33. A. Bulat. Two-stage Convolutional Part Heatmap Regression for the 1st 3D Face Alignment in the Wild (3DFAW) Challenge // ICCV - 2016

34. ResNet [Электронный ресурс] – Режим доступа:
<https://towardsdatascience.com/review-resnet-winner-of-ilsvrc-2015-image-classification-localization-detection-e39402bfa5d8> (Дата обращения: 10.05.2021)
35. Songjiang Li. Pose-Aware Face Alignment based on CNN and 3DMM // ICCV - 2019
36. Patrik Huber, Guosheng Hu. A multiresolution 3d morphable face model and fitting framework. In Proceedings of the 11th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, pages 27–29, 2016.
37. Тепловая карта [Электронный ресурс] – Режим доступа:
<https://www.cs.uic.edu/~wilkinson/Publications/heatmap.pdf> (Дата обращения: 10.05.2021).