Научно технический отчет

по теме

Разработка алгоритмов прогнозирования индивидуального поведения на основе визуального распознавания эмоций

договор №11320ГУ/2017 (код 0033562)

от 14.04.2017 , этап 1 , УМНИК 16-12

Исполнитель: Ивановский Леонид Игоревич

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Выражение лица – важная форма невербальной коммуникации, отражающая эмоции, настроение, чувства и физическое состояние человека. Умение правильно трактовать выражение лица позволяет лучше понимать собеседника, предугадывать его намерения, а в некоторых ситуациях даже понимать без слов.

Автоматический анализ выражения лица является перспективным направлением в области компьютерного зрения и машинного обучения. На сегодняшний день алгоритмы, анализирующие выражение лица человека, применяются в области видеоаналитики для решения таких задач, как оценка эффективности маркетинговых и рекламных компаний, анализ работы персонала при общении с клиентом, повышение интеллектуализации взаимодействия компьютерных систем и человека. Кроме того, подобные алгоритмы могут применяться в автоматических охранных системах для поиска злоумышленников и преступников, проявляющих эмоции, характерные для нарушителей правопорядка, террористов и прочих социально опасных личностей. Также совместное использование алгоритмов распознавания выражения лица и речи может быть применено в области психологии, психиатрии и криминалистики [1].

Известно, что для человека распознавание выражения лица не представляет сложности, в то время как для систем искусственного интеллекта эта задача является нетривиальной. Одна из сложностей автоматического распознавания заключается в том, что некоторые типы эмоций отличаются друг от друга, порой, всего несколькими, незначительными чертами, зависящими, в частности, от индивидуальных особенностей каждого человека. Кроме того, проявление эмоций и чувств может отличаться у людей разного возраста, пола и социального статуса [2]. Также в задачах компьютерного зрения дополнительные трудности могут возникнуть в связи с наличием шума и оптических препятствий на

изображениях или недостаточной освещенностью сцены [3]. Идеальный алгоритм распознавания выражения лица должен учитывать все эти факторы для получения точных результатов.

Перечисленные выше сложности автоматического анализа выражения лица требуют применения новых эффективных подходов к решению задачи классификации объектов на изображении с помощью систем искусственного интеллекта.

Для построения высокоэффективного алгоритма распознавания эмоций на данном этапе научно-исследовательской работы решаются следующие задачи:

- Анализ имеющихся баз данных с эмоциями.
- Сбор обучающей и тестовой базы.
- Разработка алгоритма определения лица человека.
- Программная реализация алгоритма выделения особых точек на снимке лица.
- Разработка алгоритма нормализации изображения.
- Программная реализация и оценка работы алгоритма детектирования улыбки.

Проведенные исследования служат основой для разработки программно-аппаратного комплекса для распознавания эмоций по изображению лица человека.

1. СБОР ОБУЧАЮЩЕЙ И ТЕСТОВОЙ БАЗЫ

Сбор базы данных с изображениями является важной стадией, предшествующей разработке алгоритма машинного обучения. Этот этап напрямую сопряжен с процессами обучения и тестирования классифицирующей модели. С помощью правильно подобранных обучающих и тестовых наборов изображений можно не только сконструировать робастый алгоритм машинного обучения, но и осуществить точный анализ разработанных моделей, а также верно оценить эффективность работы различных алгоритмов классификации.

На сегодняшний день, существует несколько доступных баз данных, которые могут быть использованы в задачах детектирования улыбки и определения эмоции человека по изображению его лица. Эти базы условно можно разделить на 2 большие группы: базы данных с двухмерными или трехмерными моделями лиц, построенными с помощью методов компьютерной графики (как например, Bosphorus или FERG-DB) и базы данных с реалистичными фотоснимками и кинороликами (как например, FER2013, AFEW/SFEW или MultiPie). В задачах детектирования улыбки и распознавания эмоций чаще используются базы картинок из второй группы, поскольку данные в них наиболее приближены к тем условиям, в которых будет работать алгоритм распознавания: они содержат снимки, сделанные в обстановке, приближенной к реальности. По сравнению с первой группой, в базах данных второй группы представлен более широкий диапазон эмоций человека. Такие базы содержат больше снимков, сделанных под различными углами обзора и с разной степенью освещенности сцены. Ко всему прочему, в открытом доступе имеется намного меньше наборов данных с моделями лиц, построенными с помощью компьютерной графики.

База данных Bosphorus содержит 4666 трехмерных моделей разрешения 1600×1200 пикселей 105 различных людей в возрасте от 25

до 35 лет. Среди людей, представленных в Bosphorus, 60 человек – мужчины и 45 – женщины. Трехмерные модели были сделаны под различными углами обзора, не превосходящими 90°. В этой базе представлены 6 типов эмоций: злость, отвращение, страх, счастье, грусть и удивление [4]. Примеры трехмерных моделей из базы данных Bosphorus показаны на рисунке 1.

Рисунок. 1 – Примеры моделей лиц из базы данных Bosphorus 3D Faces Database (Bosphorus).

База данных FERG-DB представляет собой набор из 55767 различных изображений 6 анимированных персонажей, 3 из которых – женщины и 3 – мужчины. В базе данных FERG-DB представлено 7 типов эмоций: злость, отвращение, радость, страх, грусть, спокойствие, и удивление. Все картинки из базы данных FERG-DB были размечены в соответствии с нужным типом эмоции [5]. Примеры картинок из базы данных FERG-DB показаны на рисунке 2.

Рисунок 2 – Примеры изображений из базы данных Facial Expression Research Group Database (FERG-DB).

База данных FER2013 содержит 35887 черно-белых снимков разрешения 48×48. Эта база записана в файле формата csv в виде таблицы, состоящей из 3 колонок: номер класса, характеризующий 1 из 7 типов представленных эмоций (0 – злость, 1 – отвращение, 2 – страх, 3 – счастье, 4 – грусть, 5 – удивление и 6 – спокойствие), записанные в строку 2304 значения яркости пикселей изображения и принадлежность картинки

к обучающей, проверочной или тестовой выборке. Тренировочный набор данных включает в себя 28709 изображений, а проверочная и тестовая выборка – по 3589 картинок соответственно [6]. Примеры изображений из базы данных FER2013 показаны на рисунке 3.

Рисунок 3 – Примеры изображений из базы данных FER2013 Database (FER2013).

База данных AFEW содержит размеченные по временным интервалам 1426 отрывков из 54 голливудских фильмов. Видеопоследовательности охватывают большой возрастной диапазон людей (от 1 до 70 лет), а также разнообразный фокус и расположение лиц на кадрах. Для каждого временного интервала, имеется разметка, хранящаяся в файле формата xml. Она содержит информацию об именах и возрасте актеров, возрасте их персонажей и эмоциях, представленных на данном временном интервале. База изображений AFEW включает также подмножество данных SFEW. Оно представляет собой набор из 700 отдельных цветных кадров базы AFEW. База изображений SFEW содержит 700 картинок 95 различных актеров. В этом подмножестве данных представлено 7 типов эмоций: злость, отвращение, страх, грусть, счастье, удивление и спокойствие [7]. Примеры кадров из базы данных AFEW показаны на рисунке 4.

Рисунок 4 – Примеры кадров из базы данных Acted Facial Expressions in the Wild (AFEW).

База данных Multi-Pie содержит более 750000 цветных изображений разрешения 640×480 пикселей 337 людей разных этнических групп. 69,7% из запечатленных на снимках людей – мужчины и 30,3% – женщины. Изображения были сделаны под углами обзора, не превосходящими 90°, и с разной степенью освещенности сцены. В базе

Multi-Pie представлены 6 типов эмоций: спокойствие, улыбка, удивление, заинтересованность, отвращение и крик. Эта база не содержит никакого дополнительного файла с разметкой данных на классы, однако класс для каждой картинки в соответствии с выражением лица определяется расположением файла в соответствующей директории [8]. Примеры изображений из базы данных Multi-Pie показаны на рисунке 5.

Рисунок 5 – Примеры изображений из базы данных CMU Multi-Pie Faces Database (Multi-Pie).

Среди огромного числа баз данных с изображениями лиц людей, выбор был сделан в пользу базы Multi-Pie, поскольку именно в ней для каждого из типов эмоций содержится большое количество картинок, сделанных в различных условиях.

Формирование выборки осуществлялось последовательно. Сначала для случайно отобранной картинки из базы данных Multi-Pie, с углом обзора камеры, не превышающим 45°, с помощью вычислительно эффективного PICO-алгоритма, вырезался участок размером 128×128 с изображением лица. Затем этот самый участок переводился в черно-белый формат и сохранялся, в качестве элемента будущей выборки.

Рисунок 6 – Схема преобразования данных для выборки.

В результате описанных ранее преобразований, были сформированы 2 выборки из 70000 снимков для задачи детектирования улыбок и 210000

снимков для задачи распознавания эмоций соответственно. Каждый класс был представлен набором из 35000 картинок.

Сгенерированное множество снимков в свою очередь разбивалось на тренировочную и тестовую выборку в соотношении 80/20. Эти подмножества данных не содержали одинаковых изображений. Более того, снимки с одним и тем же человеком не находились одновременно и в тренировочном, и в тестовом наборе данных. Все изображения были размечены в соответствии с классом, которому они принадлежали. Их разметка картинок хранилась в текстовых файлах формата txt. Как и для базы данных FER2013, она представляла собой таблицу, состоящую из 2 колонок: название файла с изображением и номер класса, характеризующий один из представленных в базе Multi-Pie выражений лица человека (1 из 6 классов для задачи классификации эмоций и 1 из 2 классов для задачи детектирования улыбок соответственно).

Сгенерированные подобным образом данные использовались для процессов обучения и тестирования алгоритма детектирования улыбки. Планируется также, что эти данные будут использоваться и при разработке алгоритма распознавания эмоций человека.

2. ДОРАБОТКА АЛГОРИТМА ОПРЕДЕЛЕНИЯ ЛИЦА ЧЕЛОВЕКА

Задача детектирования лица человека на изображении состоит в следующем: на заданной картинке или кадре видеопоследовательности определить наличие или отсутствие лица и при положительном ответе на этот вопрос, найти границы квадратной рамки, целиком включающей нужный участок снимка.

Среди существующих на сегодняшний момент алгоритмов детектирования лиц классическим считается метод Виолы-Джонса. Этот метод использует каскадный классификатор. Каждый классификатор поочередно пытается определить лицо на снимке с помощью скользящего окна с заданной заранее длиной шага. Длина шага влияет на точность алгоритма. Скользящим окном последовательно рассматриваются участки изображения заданного размера, и в соответствие им ставится признаковое описание, на основании которого принимается решение, содержит ли область объект заданного типа или нет. Такой способ построения модели позволяет быстро и эффективно отбрасывать картинки, не содержащие лицо человека [3]. Схема работы алгоритма Виолы-Джонса показана на рисунке 7.

Рисунок 7 – Схема работы алгоритма Виолы-Джонса.

Алгоритм Виолы-Джонса делает обнаружение объектов на снимках практически осуществимым, поскольку этот метод позволяет достаточно

быстро обрабатывать изображения, что особенно важно во время апробации алгоритма на устройствах с ограниченной операционной мощностью. Реализованный в виде функции метод Виолы-Джонса содержится в составе кросс-платформенной библиотеки компьютерного зрения OpenCV, что позволяет свободно использовать этот алгоритм для любых целей. Тем не менее, на сегодняшний день требуются еще более быстрые и качественные детекторы лиц, в связи с чем у алгоритма Виолы-Джонса появилось множество модификаций.

PICO-алгоритм является модификацией классического алгоритма детектирования Виолы-Джонса. Этот алгоритм представляет собой метод визуального обнаружения объектов на основе каскадного ансамбля оптимизированных бинарных решающих деревьев фиксированной глубины (см. рисунок 8). Каждое решающее дерево из каскадного ансамбля в своих внутренних узлах использует сопоставление интенсивностей пикселей значению некоторой логической функции, что позволяет очень быстро обрабатывать области изображения и детектировать нужные участки на нем. После выполнения проходов по всем деревьям каскадного ансамбля выходы суммируются и получается итоговое пороговое значение для определения лица человека на изображении [9].

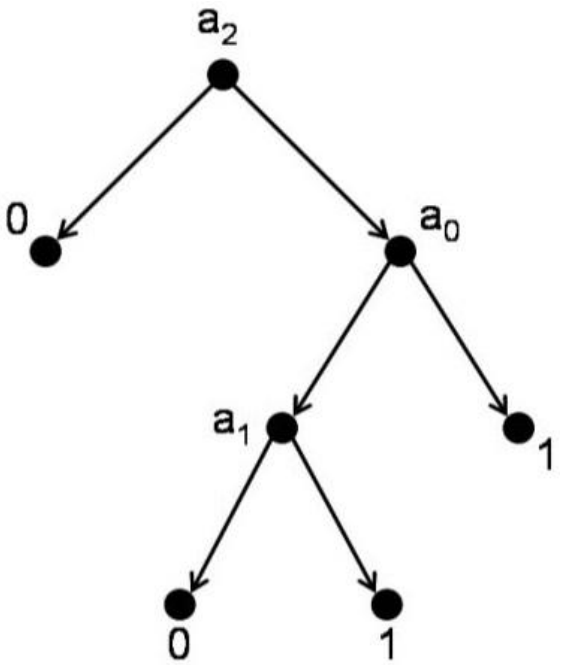

Рисунок 8 – Схема бинарного решающего дерева.

Процесс построения деревьев для PICO-алгоритма осуществляется с помощью жадного алгоритма. Параметры PICO-алгоритма задаются заранее, до начала его обучения. Эмпирическим путем были получены значения этих параметров для детектирования на фотоснимке лица человека, выражающего различные типы эмоций.

Можно выделить несколько важных преимуществ PICO-алгоритма:

- Во время своей работы PICO-алгоритм не требует вычисления никакой дополнительной структуры данных.
- В отличие от метода Виолы-Джонса, использующего несколько различных типов функций для достижения своих результатов, PICO-алгоритм использует один и тот же тип объектов во внутренних узлах своих деревьев.
- PICO-алгоритм, не требует масштабирования признаков. Другими словами, для решения задачи детектирования лиц, нет никакой необходимости в предварительной обработке изображения (например, нормализации контрастности картинки или изменения ее размеров).

Так же, как и метод Виолы-Джонса, PICO-алгоритм имеет достаточно широкое практическое применение, при этом он является более быстрым и менее чувствительным к шуму в данных. Как и для метода Виолы-Джонса, качество работы PICO-алгоритма значительно снижается в условиях работы с данными, содержащими высокий уровень шума. Это связано с особенностями построения бинарных решающих деревьев, а именно с простотой работы двоичных тестов, содержащихся во внутренних узлах, а также чувствительности деревьев к выбросам среди данных. По мере увеличения интенсивности шума в данных, снижается также и скорость работы PICO-алгоритма. Эти неблагоприятные эффекты могут быть устранены путем применения фильтра низких частот до начала работы модели. Однако, в процессе доработки алгоритма определения лица человека, этот шаг был намеренно пропущен, поскольку для таких действий, необходим уровень шума, неестественный для современных камер видеонаблюдения.

С помощью PICO-алгоритма осуществлялось формирование выборки, которая использовалась для обучения и тестирования алгоритма детектирования улыбки. В будущем, планируется использование PICO-алгоритма с целью формирования более обширных выборок, в частности для решения задачи распознавания эмоций.

3. ДОРАБОТКА АЛГОРИТМА ВЫДЕЛЕНИЯ ОСОБЫХ ТОЧЕК НА ИЗОБРАЖЕНИИ ЛИЦА ЧЕЛОВЕКА

Особые точки это уникальные характеристики объекта, которые позволяют распознавать его и отличать от объектов другого класса на изображении. Существует несколько способов, с помощью которых можно отметить такие точки. Одни способы позволяют выделить динамические особые точки соседних кадров киноролика, другие - статические особые точки, которые остаются неизменными даже при повороте лица или изменении степени освещенности сцены. Динамические особые точки, которые на самом деле также являются статическими в течение короткого промежутка времени, нужны для того, чтобы вести объект между соседними кадрами видеопоследовательности. К таким точкам можно отнести локальные минимумы и максимумы, а также углы изображения. Статические особые точки нужны в первую очередь для определения различных типов объектов. Именно такая разновидность особых точек и нужна для составления алгоритмов классификации.

Рисунок 9 – Особые точки на изображении лица человека.

В целом, алгоритм обнаружения особых точек предназначен для обнаружения объектов абсолютно любых типов. Однако, исходя из поставленных задач, можно учитывать специфические особенности лица

человека. Это позволяет значительно повысить качество их распознавания на изображении. В качестве особых точек на лице брались центры зрачков, уголки глаз, концы бровей, нос и губы человека а также некоторые точки с контура лица.

Каждая разновидность эмоции человека характеризуется специфическим расположением на лице особых точек. Эмоциональная система кодирования лицевых движений (EmFACS) П.Экмана и У.Фризена представляет собой математическую интерпретацию степени напряжения лицевых мышц, что позволяет системе EmFACS достаточно точно описать практически любое выражение лица человека.

Для описания эмоции человека с помощью системы EmFACS используются коды из 2 групп дескрипторов:

- Двигательные единицы описывают основные движения, совершаемые отдельными мышцами. Каждой лицевой мышце соответствует определенный числовой код. Для обозначения степени интенсивности движения лицевых мышц используются латинские буквы с A по E.
- Двигательные дескрипторы представляют собой движения, совершаемые группами лицевых мышц. Они описываются с помощью последовательности двигательных единиц.

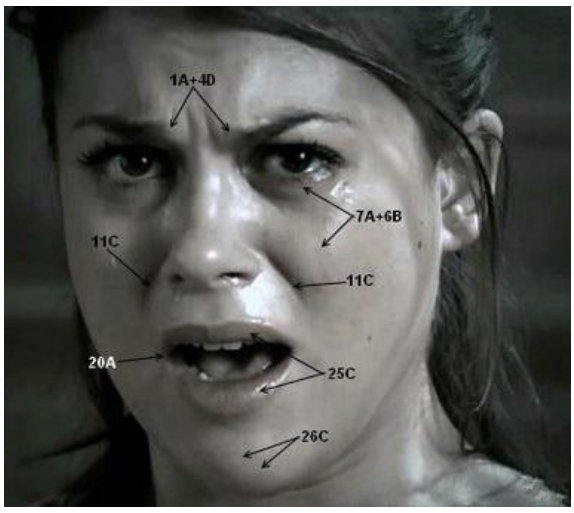

Рисунок 10 – Пример описания выражения лица человека с помощью эмоциональной системы кодирования лицевых движений (EmFACS).

Поиск особых точек для изображения лица человека осуществлялся с помощью классического алгоритма Виолы-Джонса, описанного ранее, во второй части научно-технического отчета, а также при помощи инструментов языка программирования Python. Python – это высокоуровневый, интерпретируемый язык программирования общего назначения, позволяющим реализовывать проекты любой сложности. К преимуществам Python, можно отнести также

- повышенную читаемость кода в связи с минималистичным синтаксисом языка,
- поддержку парадигм структурного, объектно-ориентированного и функционального программирования,
- большое количество легко встраиваемых готовых модулей с реализацией дополнительных функций,
- легкая переносимость кода между различными ОС.

Планируется, что информация об обнаруженных алгоритмом особых точек, будут использоваться в будущем при решении задачи классификации эмоций.

4. ДОРАБОТКА АЛГОРИТМА НОРМАЛИЗАЦИИ ИЗОБРАЖЕНИЯ

Обработка изображений является важной задачей при создании систем искусственного интеллекта. Однако во время разработки систем компьютерного зрения возникает ряд специфических проблем, таких как недостаточный уровень освещенности сцены, на которой был сделан снимок, присутствие шума на картинке, а также сложные аффинные преобразования над интересующим объектом на изображении.

Цель нормализации лица заключается в уменьшении шума, а также эффекта бесполезной и избыточной информации для улучшения процесса распознавания объектов на изображениях [10].

Существует несколько разновидностей нормализации картинок с лицами людей:

- Геометрическая нормализация для восстановления канонического вида лица осуществляется вращением, относительно центров зрачков, осевой симметрией, а также аффинным переводом, определяемым местоположением глаз. В противном случае алгоритм считает снимок уже нормализованным изображением. На сегодняшний день существует немало реализаций алгоритмов геометрической нормализации изображений.
- Яркостная нормализация осуществляется путем применения специальных фильтров, накладываемых на изображение, или же с помощью преобразований над гистограммой яркости пикселей для заданной картинки. Такой подход позволяет усреднить яркость картинки, а также справиться с данными-выбросами - пикселями с завышенным или заниженным значением яркости. Библиотека компьютерного зрения OpenCV включает в себя реализации функций по яркостной нормализации изображений.

– Масштабирование изображения для изменения ширины или высоты снимка осуществляется методом ближайшего пикселя, когда цвет пикселя в новой отмасштабированной картинке принимается равным цвету ближайшего к нему пикселя исходного изображения, а также с помощью методов интерполирующих функций. По сравнению с первым способом масштабирования снимка, подход, использующий интерполяцию, позволяет достичь более высокого качества изображения, однако он более сложен в реализации. Масштабирование картинок выполнялось в ходе формирования выборки для решения задачи детектирования улыбки.

Методы по нормализации изображения необходимы для работы с данными, поступающими на вход алгоритму машинного обучения с камеры видеонаблюдения. Суть этих методов заключается в определении воздействий, которым подвергнуты входные изображения, и последующем преобразовании картинок к эталонному виду. Методы нормализации позволяют осуществить предобработку снимка с целью повышения качества работы классифицирующей модели.

Для решения задачи нормализации изображения, было разработано приложение, написанное на языке Python с использованием библиотеки OpenCV. Оно представляет собой программный комплекс последовательно примененных функций из библиотеки компьютерного зрения. Данное приложение реализует процесс яркостной нормализации картинок, а также избавление от шумов. Планируется, что разработанный программный продукт будет использоваться перед применением алгоритма машинного обучения с целью улучшения качества картинки, передаваемой на вход модели.

5. ДОРАБОТКА АЛГОРИТМА ДЕТЕКТИРОВАНИЯ УЛЫБКИ

Одним из современных подходов при решении задачи классификации является использование алгоритмов глубокого машинного обучения на основе сверточных нейронных сетей. Сверточные сети нашли свое применение в области компьютерного зрения и биоинформатики. Несмотря на то, что такой алгоритм машинного обучения требует более тонкой настройки параметров, его главными преимуществами являются совместное использование параметров разных типов, а также тот факт, что значения признаков вычисляются в результате применения формирующихся по ходу обучения сверточных фильтров. Согласно последним исследованиям, сгенерированные таким образом признаки, как правило, позволяют получить наилучшие результаты во многих задачах классификации [11].

Архитектура сверточной нейронной сети для решения задачи детектирования улыбки разрабатывалась на основе алгоритма, предложенного в статье [12]. Ее авторы предлагают реализацию модели, позволяющей решать задачу предсказания возраста по изображениям лица человека. Разработанный алгоритм имеет простую реализацию, при этом он позволяет получить достаточно высокую точность предсказания: значение средней квадратичной ошибки составляет всего 3.27 ± 0.08.

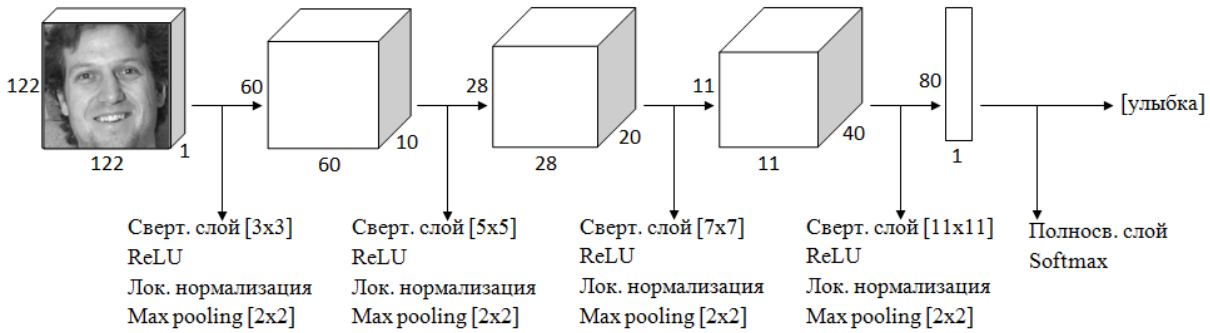

Рисунок 11 – Архитектура сверточной нейронной сети для задачи детектирования улыбки.

Как показано на рисунке 11, нейронная сеть для задачи детектирования улыбки состоит из 4 сверточных слоев (с фильтрами

размеров $3 \times 3,\ 5 \times 5,\ 7 \times 7,\ 11 \times 11$), 4 слоев с функцией активации $y = tanh(x)$, 4 слоев, реализующих процесс локальной нормализации, 3 слоев, описывающих процесс дискретизации с помощью операции max pooling (с окном размера 2×2), одного полносвязнного слоя и слоя, реализующего обобщенную логистическую функцию Softmax. По сравнению с архитектурой сети, представленной в статье [], разработанный алгоритм машинного обучения отличается тем, что для него было увеличено количество слоев свертки, активации, дискретизации и локальной нормализации. Ко всему прочему, были изменены такие параметры сверточной нейронной сети, как размер изображения, поступающего на вход модели, количество нейронов в полносвязнном слое и размеры сверточных фильтров. Значения этих параметров были получены в ходе эмпирического исследования качества работы классификатора.

Перед тем, как картинка из выборки поступала на вход сверточной нейронной сети, к ней применялась операция crop: из изображения случайным образом вырезался участок размера 122×122. Далее, эта картинка могла быть зеркально отражена относительно своей центральной вертикальной оси. Подобного рода преобразования над изображениями применялись с целью повышения робастности алгоритма машинного обучения, а также для повышения вариативности данных, поступающих на вход классифицирующей модели.

Сверточные слои являются основными элементами сверточных нейронных сетей. На этапе работы этого слоя с помощью сверточного фильтра и матричных операций преобразуются данные, поступившие на вход с предыдущего слоя нейронной сети. Результатом такой обработки является набор матриц меньшего размера – множество слоев преобразованных признаков. Условная схема работы сверточного слоя показана на рисунке 12.

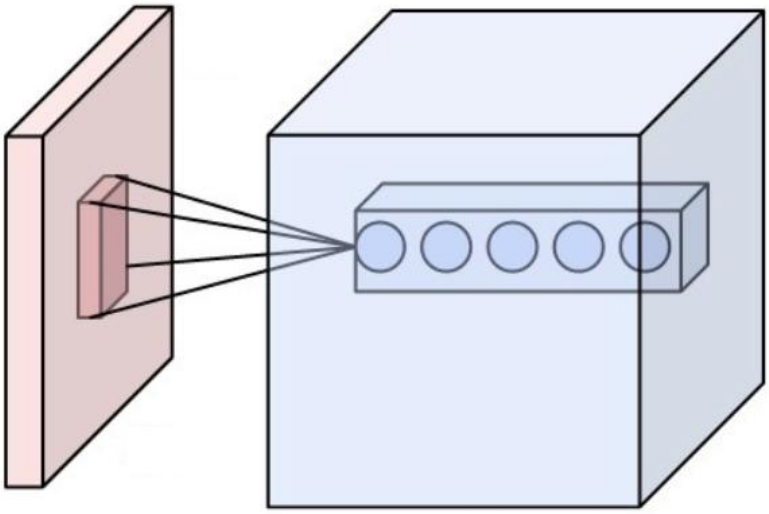

Рисунок 12 – Условная схема работы сверточного слоя.

Слои активации способствуют уменьшению значения ошибки сети. Этой слой позволяет применить к каждому элементу матрицы, поступившей на вход с предыдущего слоя, некоторую нелинейную функцию, активируя тем самым определенные значения признаков. Для задачи детектирования улыбок была выбрана функция типа гиперболического тангенса $y = tanh(x)$ График этой функции изображен на рисунке 13.

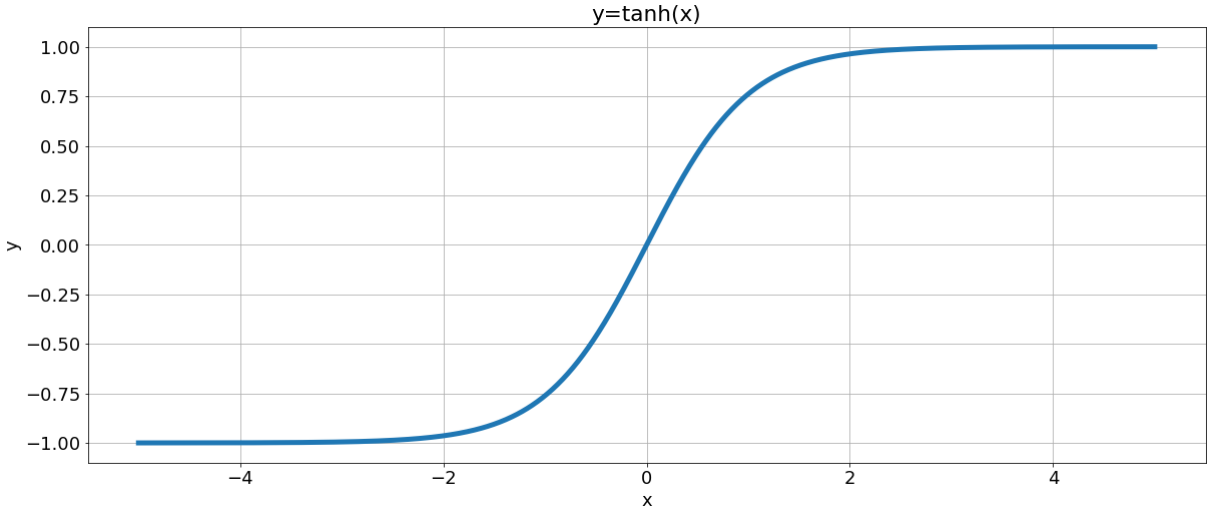

Рисунок 13 – График нелинейной функцией $y = tanh(x)$, использовавшийся в слое активации.

Слой локальной нормализации, как и слой активации, позволяет уменьшить значение ошибки сети. Однако в отличие от функции активации, это слой, реализует процесс нормализации матрицы, поступившей на вход, снижая тем самым достаточно высокие значения

признаков – выбросы, поступившие из выборки на вход алгоритму машинного обучения.

Слой дискретизации представляет собой уплотнение карты признаков, поступившей на вход с предыдущего слоя. С помощью нелинейного преобразования, затрагивающего непересекающиеся участки матрицы, группа признаков заменяется одним числовым значением, позволяя тем самым уменьшить пространственный объем признаков, теряя минимум информации. Операция max pooling характеризуется тем, что из набора данных выбирается значение признака с максимальной величиной (см. рисунок 14).

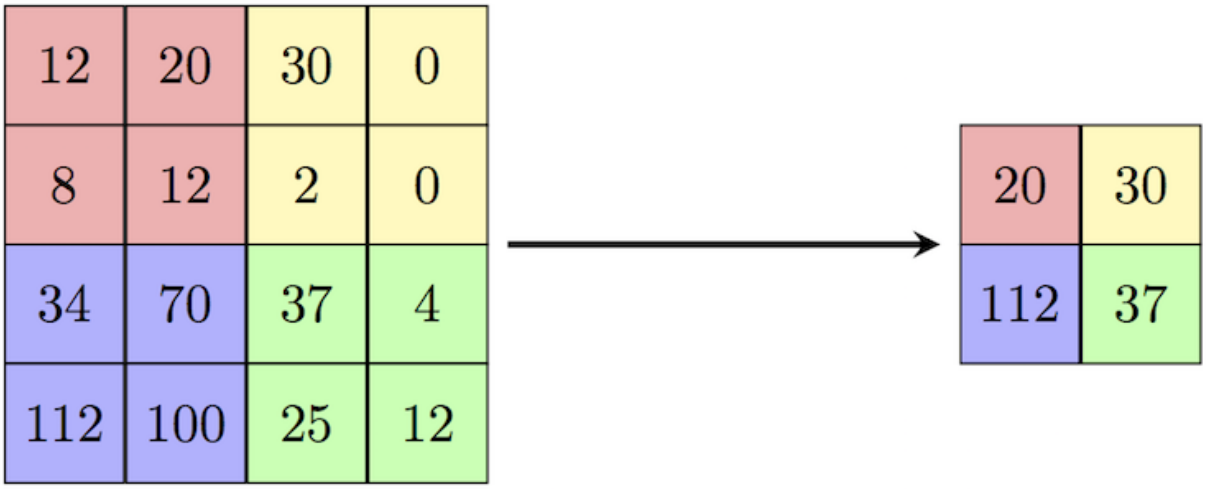

Рисунок 14 – Операция max pooling.

После того, как картинка из выборки прошла через все слои свертки, активации, дискретизации и локальной нормализации, изображение представляет собой одномерный вектор из 96 значений признаков. Этот массив подается на вход полносвязнному слою. Полносвязнный слой – двухслойная нейронная сеть, на вход которой подается некоторый набор признаков, а уже на выходе формируется вектор, с заданным количеством элементов. В случае задачи детектирования улыбки на выходе получался вектор из 2 значений признаков. Схема работы полносвязнного слоя показана на рисунке 15.

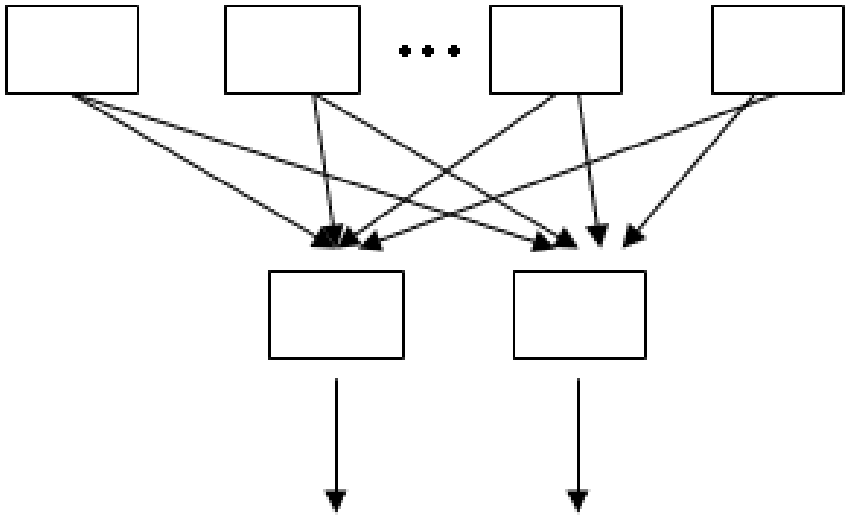

Рисунок 15 – Схема работы полносвязнного слоя.

По полученному вектору значений признаков с полносвязнного слоя нетрудно было вычислить вероятности принадлежности исходного изображения к классу «улыбки» или «неулыбки» соответственно. Это можно сделать, с помощью обобщенной логистической функции Softmax. Значение данной функции вычисляется по формуле

$$\sigma(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}},$$

где x_i - значения признаков -мерного вектора, поданного с предыдущего слоя на вход обобщенной логистической функции, а $\sigma(x)_i$ − значения вероятностей, которые получились после применения функции Softmax. В итоге, сверточная нейронная сеть позволяет классифицировать изображение лица человека в соответствии с наиболее вероятным типом эмоций: «улыбки» или «неулыбки».

В качестве алгоритма численной оптимизации использовался алгоритм стохастического градиентного спуска. Это упрощенный метод последовательного поиска локального экстремума функции с помощью движения вдоль вектора градиента. Вектор градиента ошибок предсказаний классифицирующей модели рассчитывался на каждом шаге по отдельности для одного случайно выбранного элемента из обучающей выборки. Применение такого подхода позволяет эффективно и быстро

работать с большими объемами данных, которые имеют место в задаче детектирования улыбки. Геометрическая интерпретация метода стохастического градиентного спуска изображена на рисунке 16.

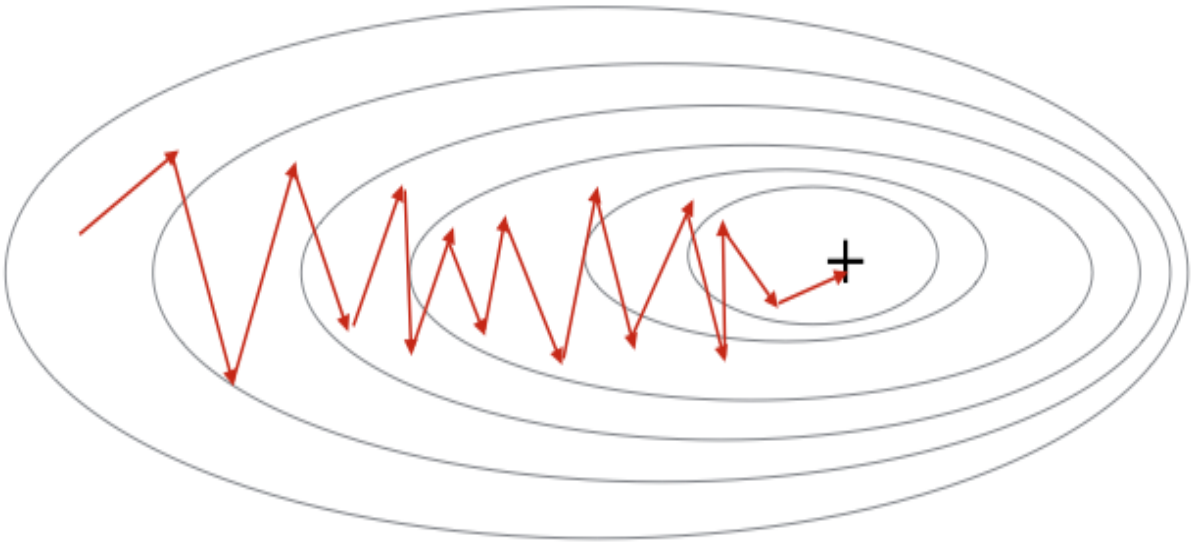

Рисунок 16 – Геометрическая интерпретация метода стохастического градиентного спуска.

Реализация архитектуры сети осуществлялась с помощью фреймворка Caffe. Эта кросс-платформенная библиотека позволяет достаточно просто, описать сверточную нейронную сеть и параметры для ее запуска в файлах формата prototxt. Библиотека Caffe содержит уже готовые реализации различных слоев сверточной нейронной сети. Главными достоинствами этой библиотеки является простота разработки моделей глубокого машинного обучения, а также тот факт, что полученными результатами, можно легко воспользоваться, встроив их в проект, написанный на C++ или Python. На сегодняшний день, фреймворк Caffe активно применяется для решения практических задач классификации [13].

Процессы обучения и тестирования разработанного алгоритма осуществлялись на суперкомпьютере NVIDIA DGX-1. Обучение на нем длилось около 40-45 минут, а тестирование занимало порядком 8-9 минут. Доля правильных ответов классификатора (A) рассчитывалась по формуле:

$$A = \frac{P}{N},$$

где P – количество изображений, по которым классификатор принял верное решение, а N – размер выборки. Для задачи детектирования улыбок доля правильных ответов классификатора составила 92,29%.

Как видно из рисунка 17, с ростом числа проделанных итераций обучения, значение функции потерь в целом убывает. Это позволяет сделать вывод о том, что алгоритм сходится. В области машинного обучения функция потерь также характеризует собой стоимость неправильно принятых решений. О хорошем качестве разработанной модели говорит и тот факт, что величина функции потерь незначительна: ее значение колеблется в интервале $[0.02, 0.17]$.

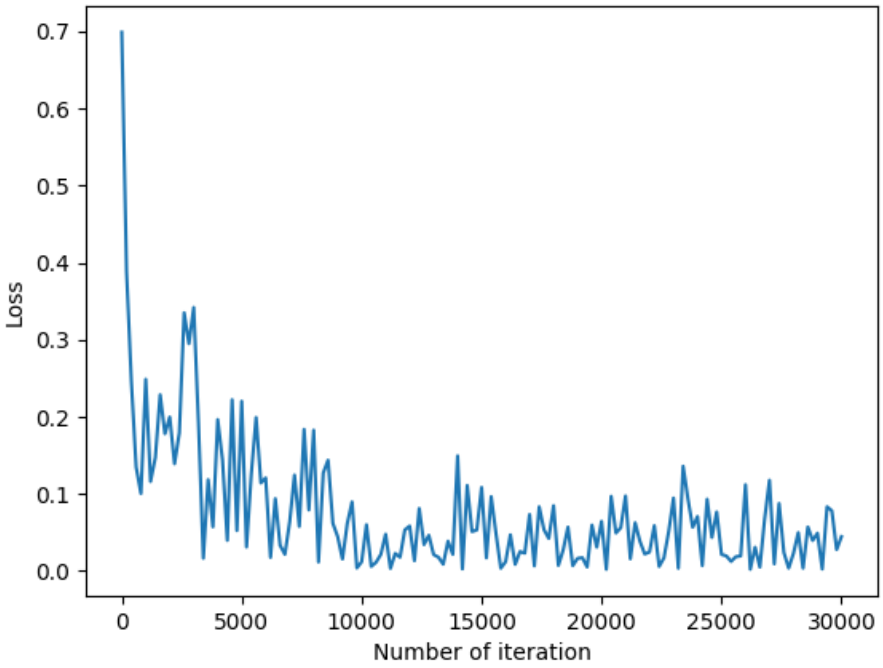

Рисунок 17 – Зависимость функции потерь от количества проделанных итераций обучения.

В дополнение к этому, была построена матрица неточностей, которая позволяет не только оценить качество работы алгоритма машинного обучения, но и определить ошибки классификации [14]. Для задачи детектирования улыбок возможно 4 различных варианта предсказаний.

Таблица 1

Матрица неточностей

Классы		Фактический класс	
		Улыбки	Неулыбки
Предсказанный класс	Улыбки	TP	FP
	Неулыбки	FN	TN

Среди значений матрицы неточностей величины TP и TN обозначают истинно-положительные и истинно-отрицательные срабатывания, характеризуя собой те случаи, когда классификатор принимал верные решения, а величины FP и FN обозначают ложно-положительные и ложно-отрицательные срабатывания классификатора (ошибки первого и второго рода соответственно), т.е. случаи, когда классификатор принимал неверные решения. В зависимости от текущей задачи классификации, разработчики могут сделать упор в сторону уменьшения значений ошибки первого или второго рода. В случае же задачи классификации они имеют одинаковую цену.

Как видно из таблицы 2, разработанный алгоритм показал достаточно высокие результаты. По сравнению с верными решениями классификатора ошибки первого и второго рода незначительны. Однако стоит отметить тот факт, что модель чуть чаще пропускала картинки с улыбками, а не наоборот, определяла улыбку на изображении лица, где она отсутствует. Этот факт можно объяснить тем, что в классе «неулыбки» представлен более широкий спектр эмоций людей (спокойствие, удивление, спокойствие, отвращение, заинтересованность и крик).

Таблица 2

Матрица неточностей для задачи детектирования улыбки

Классы		Фактический класс	
		Улыбки	Неулыбки
Предсказанный класс	Улыбки	6590	246
	Неулыбки	410	6754

Хорошие результаты работы классифицирующей модели подтверждаются также и посчитанными для каждого класса значениями точности (P), полноты (R) и F-меры (F). Значения этих метрик качества вычислялись по следующим формулам:

$$P = \frac{TP}{TP + FP}, \qquad R = \frac{TP}{TP + FN}, \qquad F = 2\frac{P \times R}{P + R}.$$

Величина точности классификатора (P) отражает долю тех ответов модели, которые принадлежат нужному классу относительно всех объектов этого класса из выборки. Значение полноты (R) отражает долю найденных алгоритмом ответов, действительно относящихся к данному классу относительно всех объектов этого класса из выборки. F-мера представляет собой среднее гармоническое значений точности и полноты классификатора [15]. Как видно из таблицы 3 величина F-меры для каждого класса составляет около 0,95.

Таблица 3

Анализ ошибок классификатора для задачи детектирования улыбок

Анализ ошибок		Метрики качества модели		
		Точность	Полнота	F-мера
Классы	Улыбки	0,96	0,94	0,95
	Неулыбки	0,94	0,96	0,95

Ко всему прочему, была построена ROC-кривая – график оценки качества работы бинарного классификатора, отражающего зависимость долей истинных и ошибочных положительных классификаций, TPR (True Positive Rate) и FPR (False Positive Rate) (см. рисунок 18). Эти показатели называют также специфичностью и чувствительностью алгоритма соответственно. Исходя из сконструированной матрицы неточностей, величины TPR и FPR можно рассчитать по следующим формулам:

$$TPR = \frac{TP}{TP + FN}, \qquad FPR = \frac{FP}{FP + TN}$$

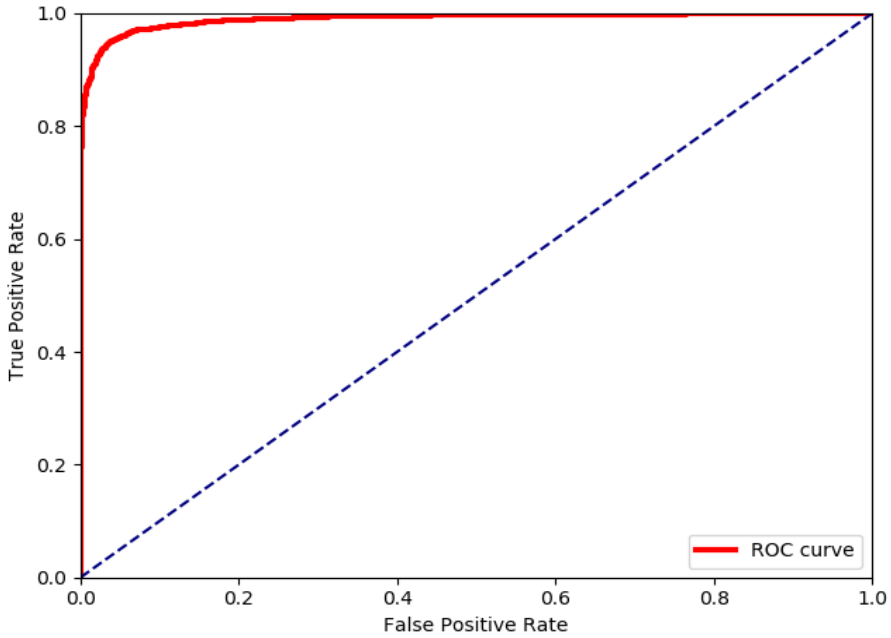

Рисунок 18 – ROC-кривая для задачи детектирования улыбки.

Для ROC-кривой была также рассчитана ее количественная характеристика – значение площади под ней, величина AUC ROC. Чем больше показатель AUC-ROC, тем лучше качество классифицирующей модели [16]. Для разработанной модели, величина AUC-ROC составила порядком 0,98. Наряду с высоким значением F-меры, это является ещё одним показателем хорошей работы алгоритма детектирования улыбки по изображению лица человека.

ЗАКЛЮЧЕНИЕ

В ходе работы над проектом были проанализированы литературные источники, исследованы основные принципы работы алгоритмов машинного обучения, а также были изучены необходимые инструменты для их разработки.

Программная реализация алгоритмов осуществлялась с помощью инструментов языков программирования Python и C#, а также библиотеки компьютерного зрения OpenCV и фреймворка Caffe для разработки сверточных нейронных сетей. В результате работы над проектом

- Были созданы высокоэффективные алгоритмы для решения задач нормализации изображения, определения лица человека на заданном снимке, выделения особых точек и детектирования улыбки.
- Для решения задачи детектирования улыбки, а также для последующей разработки алгоритма распознавания эмоций, в результате анализа имеющихся баз данных с изображениями были собраны и размечены обучающие и тестовые выборки.
- Была оценена эффективность работы алгоритма детектирования улыбки. Посчитанные из матрицы неточностей показатели точности и полноты классификатора для каждого из имеющихся классов, а также высокие значения доли правильных ответов, F-меры и площади под ROC-кривой подтверждают эффективность работы классифицирующей модели.

Таким образом, разработанную модель для детектирования улыбки уже можно встраивать в приложения искусственного интеллекта, актуальные в области NeuroNet рынка НТИ.

На втором этапе научно-исследовательской работы планируется реализация алгоритма глубокого машинного обучения для определения

эмоций человека по изображениям на основе сверточных нейронных сетей, а также его тестирование и апробация на пилотных объектах.

СПИСОК ЛИТЕРАТУРЫ

[1] Ахметшин Р.И., Кирпичников А.П., Шлеймович М.П. (2015) Распознавание эмоций человека на изображениях // Вестник Казанского технологического университета. 2015. №11.

[2] Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение – М.: ДМК Пресс, 2017, 652 с.

[3] Гонсалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2005. – 621 с.

[4] The Bosphorus 3D Faces Database. URL: http://bosphorus.ee.boun.edu.tr/default.aspx.

[5] Facial Expression Research Group Database. URL: http://rail.cs.washington.edu/projects/deepexpr/ferg-db.html.

[6] The FER2013 Database. URL: https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data.

[7] Acted Facial Expressions In The Wild. Static Facial Expressions in the Wild. URL: https://cs.anu.edu.au/few/AFEW.html.

[8] The CMU Multi-PIE Face Database. URL: http://www.cs.cmu.edu/afs/cs/project/PIE/MultiPie/Multi-Pie/Home.html.

[9] Markus N., Frljak M., Pandzic I.S., Ahlberg J., Forchheimer R. Object Detection with Pixel Intensity Comparisons Organized in Decision Trees. URL: https://arxiv.org/pdf/1305.4537.pdf.

[10] Struc V., Pavesic N. Image Normalization Techniques for Robust Face Recognition // Proceedings of the 8th WSEAS International Conference on SIGNAL PROCESSING, ROBOTICS and AUTOMATION, 2010, pp. 155 – 160.

[11] Николенко С., Кадурин А., Архангельская Е. Глубокое обучение. Погружение в мир нейронных сетей. – СПб.: Питер, 2018, 480 с.

[12] Niu Z., Zhou M., Wang L., Gao X., Hua G. Ordinal Regression with Multiple Output CNN for Age Estimation // IEEE CVPR, 2016.

[13] Caffe Framework [Электронный ресурс]. URL: http://caffe.berkleyvision.org.

[14] Вандер Плас Дж. Python для сложных задач: наука о данных и машинное обучение. СПб.: Питер, 2018, 576 с.

[15] Рашка С. Python и машинное обучение // ДМК Пресс, 2017, 418 с.

[16] Scikit-learn. Web: http://scikit-learn.org/stable.

Приложение А. Исходный код архитектуры сверточной нейронной сети для решения задачи детектирования улыбки.

smile_cnn.prototxt

```
name: "SmileDetection"

layer {
  name: "data"
  type: "Data"
  include {
    phase: TRAIN
  }
 transform_param {
    crop_size: 122
    mean_value: 127
    scale: 0.00390625
    mirror: false
 }
 data_param {
    source: "train_lmdb"
    batch_size: 32
    backend: LMDB
  }
  top: "data"
  top: "label"
}

layer {
  name: "data"
  type: "Data"
  include {
    phase: TEST
  }
```

```
 transform_param {
    crop_size: 122
    mean_value: 127
    scale: 0.00390625
    mirror: false
 }
 data_param {
    source: " test_lmdb"
    batch_size: 24
    backend: LMDB
  }
  top: "data"
  top: "label"
}

layer {
  name: "conv0"
  type: "Convolution"
  bottom: "data"
  top: "conv0"
  param {
    lr_mult: 1
  }
  param {
    lr_mult: 2
  }
  convolution_param {
    num_output: 10
    kernel_size: 3
    stride: 1
    weight_filler {
        type: "xavier"
    }
    bias_filler {
        type: "constant"
```

```
      value: 0
    }
  }
}
layer {
  name: "tanh0"
  type: "TanH"
  bottom: "conv0"
  top: "tanh0"
}
layer {
  name: "norm0"
  type: "LRN"
  bottom: "tanh0"
  top: "norm0"
}
layer {
  name: "pool0"
  type: "Pooling"
  bottom: "norm0"
  top: "pool0"
  pooling_param {
    kernel_size: 2
    stride: 2
    pool: MAX
  }
}

layer {
  name: "conv1"
  type: "Convolution"
  bottom: "pool0"
  top: "conv1"
  param {
    lr_mult: 1
```

```
  }
  param {
    lr_mult: 2
  }
  convolution_param {
    num_output: 20
    kernel_size: 5
    stride: 1
    weight_filler {
        type: "xavier"
    }
    bias_filler {
        type: "constant"
        value: 0
    }
  }
}
layer {
  name: "tanh1"
  type: "TanH"
  bottom: "conv1"
  top: "tanh1"
}
layer {
  name: "norm1"
  type: "LRN"
  bottom: "tanh1"
  top: "norm1"
}
layer {
  name: "pool1"
  type: "Pooling"
  bottom: "norm1"
  top: "pool1"
  pooling_param {
```

```
    kernel_size: 2
    stride: 2
    pool: MAX
  }
}

layer {
  name: "conv2"
  type: "Convolution"
  bottom: "pool1"
  top: "conv2"
  param {
    lr_mult: 1
  }
  param {
    lr_mult: 2
  }
  convolution_param {
    num_output: 40
    kernel_size: 7
    stride: 1
    weight_filler {
        type: "xavier"
    }
    bias_filler {
        type: "constant"
        value: 0
    }
  }
}
layer {
  name: "tanh2"
  type: "TanH"
  bottom: "conv2"
  top: "tanh2"
```

```
}
layer {
  name: "norm2"
  type: "LRN"
  bottom: "tanh2"
  top: "norm2"
}
layer {
  name: "pool2"
  type: "Pooling"
  bottom: "norm2"
  top: "pool2"
  pooling_param {
    kernel_size: 2
    stride: 2
    pool: MAX
  }
}

layer {
  name: "conv3"
  type: "Convolution"
  bottom: "pool2"
  top: "conv3"
  param {
    lr_mult: 1
  }
  param {
    lr_mult: 2
  }
  convolution_param {
    num_output: 80
    kernel_size: 11
    stride: 1
    weight_filler {
```

```
      type: "xavier"
    }
    bias_filler {
      type: "constant"
      value: 0
    }
  }
}
layer {
  name: "tanh3"
  type: "TanH"
  bottom: "conv3"
  top: "tanh3"
}
layer {
  name: "norm3"
  type: "LRN"
  bottom: "tanh3"
  top: "norm3"
}

layer {
  name: "fc4"
  type: "InnerProduct"
  bottom: "norm3"
  top: "fc4"
  param {
    lr_mult: 1
  }
  param {
    lr_mult: 2
  }
  inner_product_param {
    num_output: 2
    weight_filler {
```

```
      type: "xavier"
    }
    bias_filler {
      type: "constant"
      value: 0
    }
  }
}

layer {
  name: "loss"
  type: "SoftmaxWithLoss"
  bottom: "fc4"
  bottom: "label"
  top: "loss/loss"
}
layer {
  name: "accuracy/top1"
  type: "Accuracy"
  bottom: "fc4"
  bottom: "label"
  top: "accuracy@1"
  include: {
     phase: TEST
  }
  accuracy_param {
    top_k: 1
  }
}
```