МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КубГУ»)

Факультет физико-технический
Кафедра физики и информационных систем

Курсовой проект

ОЦЕНКА НЕВЕРБАЛЬНОГО ПОВЕДЕНИЯ ЧЕЛОВЕКА

Работу выполнил _____ Александров Дмитрий Николаевич
(подпись)

Направление магистерской подготовки 03.04.02 Физика

Руководитель магистерской программы «Информационные процессы и системы»
канд. физ.-мат. наук, доцент _____ Л.Р. Григорьян
(подпись, дата)

Научный руководитель
канд. физ.-мат. наук _____ М.С. Коваленко
(подпись, дата)

Нормоконтролёр
канд. физ.-мат. наук _____ М.С. Коваленко
(подпись, дата)

Краснодар
2018

РЕФЕРАТ

Курсовой проект**,** 49 с., 18 рис., 3 табл., 15 источников.

РАСПОЗНАВАНИЕ ЛИЦ, КОМПЬЮТЕРНОЕ ЗРЕНИЕ, МЕТОД СОПОСТАВЛЕНИЯ С ЭТАЛОНОМ, МЕТОД ОПОРНЫХ ВЕКОТОРОВ, СВЁРТОЧНЫЕ НЕЙРОННЫЕ СЕТИ, МЕТОД ВИОЛЛЫ-ДЖОНСА, ПРИМИТИВЫ ХААРА, MLP, OPENCV, PYTHON, МЕТОД ОБРАТНОГО РАСПОСТРАНЕНИЯ ОШИБКИ.

Объектом исследования данной работы является оценка невербального поведения человека.

Целью работы является создание программы для оценки невербального поведения человека.

В результате выполнения работы проведен обзор современных систем распознавания эмоций, рассмотрены основные методы распознавания лиц и эмоций, с указанием преимуществ, недостатков и оценкой эффективности. Разработан алгоритм распознавания лиц и эмоций, реализованный в программе.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Человек каждую секунду испытывает эмоциональные реакции. Не имеет значение какие они – положительные или отрицательные, важно то, что мозг человека в каждый момент времени вырабатывает определенные гормоны, заставляющие человеческое тело говорить, писать и двигаться определенным образом. Что такое эмоциональная реакция сказать никто не может. Одни учёные утверждают, что эмоции – это продукт физиологических процессов, другие, что физиологические процессы – следствие эмоций. Однако важно одно – человек не способен полностью скрыть происходящую эмоциональную реакцию. В связи с этим всегда была идея «прочесть» внутреннее состояние человека и его характер по его поведению. Данное направление получило название профайлинг – совокупность психологических методов и методик оценки и прогнозирования на основе анализа наиболее информативных частных признаков, характеристик внешности, невербального и вербального поведения. Изначально данные методы не были представлены общественности. Затем, в конце XX века, благодаря книге Юлия Фаста о невербальном общении (1970) и книге Аллана Пиза «Язык телодвижений» (1981) стали доступны данные о «языке тела».

Наряду с чисто психологическими методами оценки поведения человека, широкое распространение получили приборы снимающие показания биометрических данных. Одним из таких приборов является полиграф. Считается, что полиграф невозможно обмануть и с помощью него всегда можно узнать правду. Но полиграф определяет правду относительно. Правда – это то, что истинно для человека, а поверить можно во всё что угодно. А ложь – попытка скрыть то, что ты помнишь и знаешь. Таким образом остаётся возможность обмануть полиграф, да и применение его сильно ограничено лабораторными условиями. Следовательно, необходимо разрабатывать системы, способные определять отношение объекта к чему-либо независимо от места проведения испытаний, независимо от того знает

объект о проводимых испытаниях или нет, независимо от каких-либо внешних влияний. В связи с этим, в последнее время появилось новое, стремительно развивающееся направление – распознавание эмоций человека [1].

Однако на сегодняшний момент не существует системы, полностью реализующей анализ всех средств передачи эмоциональных реакций.

Тема данной работы является оценка невербального поведения человека. В качестве оценки невербального поведения человека выбрана фиксация эмоционального выражения лица человека. В работе рассматривается рынок систем компьютерного зрения, ориентированных на распознавание эмоций, резюмированы итоги обзора. Далее в работе описаны основные методы и подходы, используемые как при обнаружении лиц, так и при распознавании эмоций. В заключительной, практической части работы представлен готовый алгоритм решение, с описанием методов и основных используемых модулей – программа, обнаруживающая на видеоряде лицо и позволяющая распознавать эмоции.

Целью данной работы является создание программы для оценки невербального поведения человека.

Для достижения цели необходимо решить следующие задачи:

- произвести обзор современных систем компьютерного зрения, используемых в сегменте распознавания эмоций;

- рассмотреть современные подходы и методы обнаружения лица и распознавания эмоций;

- разработать алгоритм, используемый в программе для оценки невербального поведения человека.

Полученные в данной работе результаты могут быть использованы в анализе систем компьютерного зрения, а также могут найти реальное применение для проектирования систем распознавания эмоций.

1 Обзор рынка систем компьютерного зрения

В настоящее время существует довольно больше количество готовых программных продуктов в той или иной степени решающих задачу обнаружения и распознавания эмоций. Технологии компьютерного зрения, как и собственно технологии остальных интеллектуальных информационных систем (анализ введённого текста, распознавание речи, автоматизированный анализ большого количества данных, системы распознавания биометрических данных и т.д.) уже почти повсеместно интегрированы в повседневную жизнь. Так, камеры современных смартфонов умеют не только распознать лицо во время съёмки, фокусируя на найденном лице объектив камеры, но и фиксируют жесты рук и воспринимают человеческую речь для того, чтобы сделать снимок. Последним «словом» в смартфонах стала представленная от Apple в сентябре 2017 года технология Face ID, сканирующая с помощью фронтальной камеры лицо человека для разблокировки смартфона. А в автомобилях премиального сегмента происходит фиксация жестов рук для управления мультимедийной системой автомобиля, позволяя, например, переключить трек или изменить уровень громкости [2].

К счастью, технологии компьютерного зрения используются не только в развлекательных целях, найдя применение, например, в медицинском анализе изображений, контроле автоматизированных транспортных средств, системах безопасности, системах визуального контроля (например, считывание штрих/QR-кодов).

Большим ответвлением в компьютерном зрении является распознавание эмоций. Ниже буду рассмотрены наиболее популярные программные продукты в данной сфере.

Ниже будут рассмотрено несколько систем распознавания эмоций от разных компаний. Некоторые изначально специализировались исключительно на компьютерном зрении, а некоторые, такие как Microsoft работают в данной сфере сравнительно недавно.

1.1 Программа FaceReader

Программа может верно интерпретировать следующие выражения лица: счастливое, грустное, сердитое, удивленное, испуганное, недовольное, нейтральное.

Также, FaceReader способен по лицам людей определять их возраст, пол и этническую принадлежность. Не нуждается в обучении и дополнительной настройке.

В программе реализованы технологии компьютерного зрения. В частности, это метод Active Template, заключающийся в наложении на изображение лица деформируемого шаблона. Действие метода показано на рисунке 1.

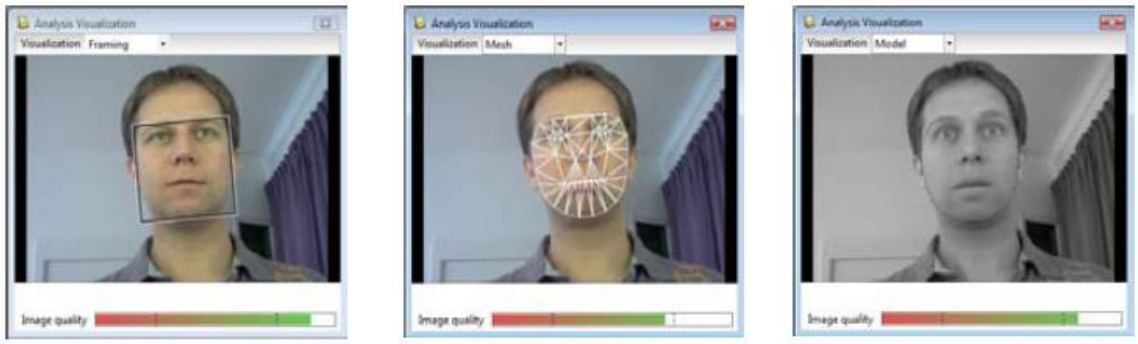


Рисунок 1 – Метод Active Template

Также, реализован метод Active Appearance Model, с помощью которого можно создавать искусственную модель лица с учетом контрольных точек и деталей поверхности, и сравнивать ее с образцами, заложенными в память.

Классификация происходит методами нейронных сетей с тренировочным набором в 2 000 фотографий.

Возможности программы:

- средний процент распознавания эмоций равен 89%. Для некоторых эмоций он выше, для некоторых ниже;

- наклон лица может быть любым в плоскости, его система обнаружит;
- программа работает с загружаемым видео в форматах с кодеками MPEG1, MPEG2, XviD, DivX4, DivX5, DivX6, DV-AVI и AVI, причем определять эмоции можно поэтапно, либо полностью при просмотре всего видео. Также, FaceReader может работать со статичными изображениями, а также в реальном времени, если у пользователя подключена веб-камера;
- программа прекрасно визуализирована: всегда можно посмотреть гистограммы, диаграммы, процент выражаемых эмоций. А на таймлайне видны проявления микровыражений в определенный промежуток времени.

Недостатки программы:

- FaceReader не натренирован для распознавания детей до 5ти лет;
- если человек в очках, то распознавание эмоций неточное, либо классификация не ведется;
- люди с разным цветом кожи по-разному воспринимаются системой, программа не до конца адаптирована;
- повернутое лицо не детектируется.

1.2 Продукты Affective Computing Research Group

Компания Розалинды Пикард, Affectiva, известна в первую очередь поставляемыми носимыми биосенсорами Q-Sensor. Но не только этим богата компания. Есть огромный опыт внедрения технологий среди Affective computing, или эмоциональных вычислительных систем, разработки идут с 1995 года. Проектов очень много. Это самая старейшая группа разработчиков, занимающаяся данными технологиями [3].

Одним из способов распознавания эмоционального состояния по лицу в данных разработках является запись в реальном времени с последующим компьютерным анализом — методами сравнения с заложенными образцами (SURF и на основе SIFT-дескрипторов), а также вейвлет-методами. Работа

данных методов применяется в такой программе как Pupeteer, оценивающей поведение и эмоциональное состояние учеников.

В ходе эксперимента по данному методу шесть базовых эмоций компьютер определяет с 96-процентной точностью.

Решение примечательно еще тем, что распознает вкупе с эмоциями движения головы, такие как кивок или качание, мотания из стороны в сторону. Используются процессы Байесовского машинного обучения для классификации эмоций, а также для вычисления статистики и вычисления смешанных состояний, когда нельзя точно выразить, какая именно эмоция превалирует.

1.3 Проект Microsoft API Emotion Recognition

API распознавания эмоций принимает данные о выражении лица на изображении в качестве входных, а затем возвращает точные сведения об эмоциях на каждом лице на изображении, а также ограничивающую рамку для лица с помощью API распознавания лиц. Если пользователь вызвал API распознавания лиц, в качестве необязательных входных данных можно отправить лицо в рамке.

Распознаваемые эмоции: гнев, презрение, отвращение, страх, счастье, нейтральное выражение, грусть и удивление. Эти эмоции знакомы в разных культурах и распознаются в зависимости от определенной мимики.

Бесплатная версия продукта доступна с ограничениями в 20 распознаваний в минуту и в 10000 распознаваний в месяц [4].

Среди плюсов данной системы стоит отметить то, что её возможно включить в какой-либо программный продукт, при этом обработка изображений и распознавание эмоций происходит в облачном сервисе Microsoft. Результат работы программ изображён на рисунке 2.

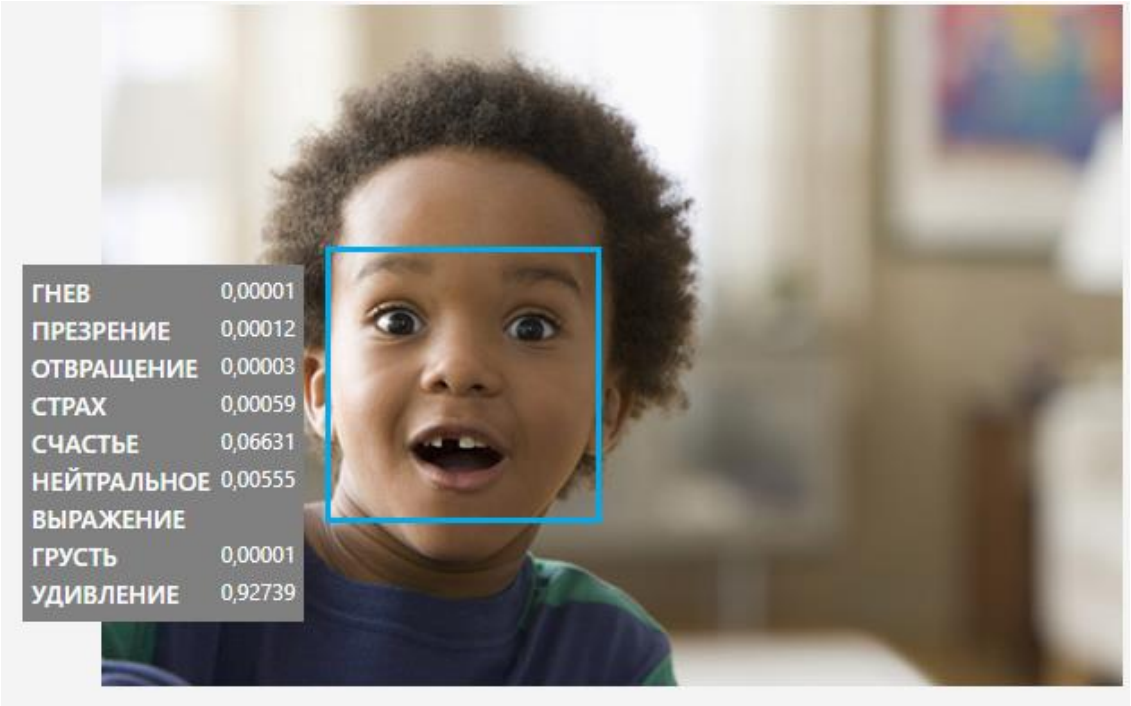


Рисунок 2 – Результат работы Microsoft API Emotion Recognition

1.4 Итоги обзора систем

Рассмотрены решения лишь некоторых крупных компаний. Остальные предоставляют свои продукты, предназначенные немного для иных задач, но разрабатываемые ими системы так или иначе интересны, потому что могут быть легко усовершенствованы до уровня распознавания эмоций. Это программы и решения, выполняющие такие задачи как:

- верификация лица (системы безопасности и контроля доступа);
- трекинг и отслеживание лица (системы видеонаблюдения);
- сравнение людей по образу и подобию своему (системы поиска);
- анимация лица и его преобразование (системы морфинга);
- преобразование лица в 3D – модели (системы моделирования);
- определение расы, возраста и пола человека (системы гендерной классификации);

Причем большинство компаний, разрабатывающих данные программы, предоставляют свой инструментарий (SDK – Software Development Kit) любому разработчику.

Но всё же ключевым критерием применения таких продуктов является точность распознаваний эмоций. Явного лидера среди рассмотренных программ по точности распознаваний выявить нельзя.

2 Современные методы распознавания эмоций

В настоящее время разработано большое количество разнообразных методов распознавания эмоций. Однако, у всех методов существует ряд общих проблем:

- все методы распознавания эмоций ориентированы на фронтальные изображения;

- особенности окклюзии: наличие бороды, усов, очков или шляпы вводят в заблуждение алгоритмы распознавания, либо вообще не позволяя распознать лицо;

- условия съёмки: к данному пункту можно отнести как качество съёмки самой камеры, так и качество освещения.

Одной из первых успешных попыток систематизировать лицевые эмоции является система кодирования лицевых движений (СКЛиД), разработанная Полом Экманом и Уоллесом Фризеном в 1978 году. С помощью СКЛиД можно закодировать выражение лица, создавая модель из определенных единиц действия, а также фиксированного промежутка времени для воспроизведения того или иного выражения лица. Система кодирования имеет фиксированные номера дескрипторов, имеющие существенную разницу с единицами действия.

Основой для СКЛиД являются двигательные единицы (ДЕ), которые позволяют закодировать любое автоматически возможное выражение лица. В проявлении эмоций участвуют не все мышцы, а лишь их небольшая часть. Основу каждой из базовых эмоций составляют 5-7 ДЕ, которые позволяют закодировать наиболее часто встречающиеся прототипы.

Например, формула (1) для удивления:

$$1 + 2 + 5B + 26. \quad (1)$$

Латинские буквы от A до E, стоящие после чисел, определяют интенсивность движения от минимальной до максимальной соответственно.

Хотя СКЛиД и требует хорошо обученных экспертов, существует положительный опыт интегрирования данной системы в компьютеры. Но всё же данный метод не получил широкого применения в технологиях компьютерного зрения. Ниже рассмотрены методы, позволяющие добиться более высокой эффективности распознавания эмоций.

2.1 Распознавание лиц на основе матрицы изменения яркостей

В задачах детекции и распознавания лиц ключевым моментом является введение меры близости на изображениях. При этом мера близости по возможности должна удовлетворять аксиомам метрики и, кроме того, соответствовать особенностям восприятия зрительной системы человека. При анализе изображения лица, человек не обращает внимания на искажение яркостной составляющей изображения на наличие теней и бликов, а нализирует характер изменения яркости, придавая большее значение качественным различиям, а не количественным [5].

Для решения задач детекции и идентификации лиц, удобно осуществлять переход от исходного изображения I к матрице изменения яркостей M^I , элементами которой являются пары чисел, соответствующие знаками частных производных от яркости исходного изображения в каждой точке. Выражение перехода с помощью формулы (2):

$$M^I_{t,j} = |sgnI'_x|_{t,j}, sgnI'_y|_{t,j}|. \quad (2)$$

Представление изображения в виде матрицы изменения яркостей обеспечивает устойчивость алгоритмов распознавания к изменению условий освещения.

В качестве меры близости для изображений, удовлетворяющей описанным выше требованиям, используется метрика Хэмминга (3) для матриц изменения яркостей:

$$\rho_w(I,J) = \sum_{t,j} w_{t,j}[M^I_{t,j} = M^J_{t,j}], \quad (3)$$

где $w_{i,j} \geq 0$ – весовые коэффициенты.

На рисунке 3 представлены шаблон лица, используемый для детекции лиц на изображениях, и соответствующая ему матрица изменения яркостей.

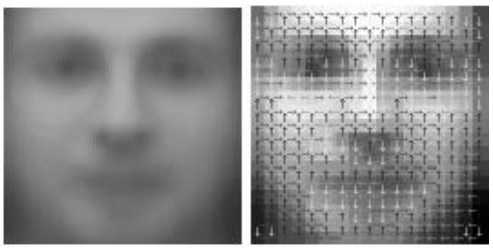


Рисунок 3 – Шаблон лица и визуальное представление соответствующей ему матрицы изменения яркостей

2.2 Метод сопоставления с эталоном (Template Matching)

В данном подходе процесс распознавания разбивается на части, соответствующие отдельным чертам лица. Каждая фотография, поступающая на вход распознающей системы, должна представлять собой фронтальное изображение лица человека с определенным для конкретной базы данных количеством масок, представляющих основные для идентификации регионы лица (например, глаза, нос, рот и нижняя часть лица). Кроме того, расположения данных масок должны быть одинаково нормализованы (например, относительно положения глаз) для всех изображений в базе данных.

Во время процесса распознавания, когда части входного изображения по очереди сравниваются с частями изображения, хранящегося в базе, используется вектор, отражающий результат сравнения в баллах (один балл за каждую совпавшую черту лица) и вычисляемый путем нормализованной взаимной корреляции (впрочем, методы сравнения могут быть разными). После чего входное изображение классифицируется в соответствии с максимально набранными баллами. Имеются также некоторые разновидности

данного подхода, например, с изменяющимися в процессе сравнения эталонами [6].

2.3 Активные модели внешнего вида

Активные модели внешнего вида (Active Appearance Models, AAM) — это статистические модели изображений, которые путем разного рода деформаций могут быть подогнаны под реальное изображение. Данный тип моделей в двумерном варианте был предложен Тимом Кутсом и Крисом Тейлором в 1998 году. Первоначально активные модели внешнего вида применялись для оценки параметров изображений лиц.

Активная модель внешнего вида содержит два типа параметров: параметры, связанные с формой (параметры формы), и параметры, связанные со статистической моделью пикселей изображения или текстурой (параметры внешнего вида). Перед использованием модель должна быть обучена на множестве заранее размеченных изображений. Разметка изображений производится вручную. Каждая метка имеет свой номер и определяет характерную точку, которую должна будет находить модель во время адаптации к новому изображению.

Процедура обучения AAM начинается с нормализации форм на размеченных изображениях с целью компенсации различий в масштабе, наклоне и смещении. Для этого используется так называемый обобщенный Прокрустов анализ.

Из всего множества нормированных точек затем выделяются главные компоненты с использованием метода PCA.

Далее из пикселей внутри треугольников, образуемых точками формы, формируется матрица, такая что, каждый ее столбец содержит значения пикселей соответствующей текстуры. Стоит отметить, что используемые для обучения текстуры могут быть как одноканальными (градации серого), так и многоканальными (например, пространство цветов RGB или другое). В случае

многоканальных текстур векторы пикселов формируются отдельно по каждому из каналов, а потом выполняется их конкатенация. После нахождения главных компонент матрицы текстур модель AAM считается обученной.

Эффективность метода: до 88%.

2.4 Метод опорных векторов

Метод опорных векторов (SupportVectorMachine, SVM) был предложен российскими учеными В. Вапником и А. Червоненкисом. Данный метод является линейным классификатором и применяется для решения задач классификации и регрессионного анализа.

Применение метода SVM к решению задачи классификации объектов на изображениях можно описать следующим образом. Дано множество точек в пространстве, которые сгруппированы по двум классам. Для того, чтобы разделить множество точек на классы можно провести разделяющую гиперплоскость. С точки зрения классификации необходимо найти оптимальную гиперплоскость, расстояние от которой до каждого класса будет максимально. Чем больше расстояние от разделяющей гиперплоскости до каждого класса, тем меньше ошибка классификации. Вектора, расположенные в пространстве ближе всего к гиперплоскости, называются опорными векторами.

Необходимо найти классифицирующую функцию f(x), которая принимает разные значения для векторов разных классов. Для определения данной функции используется обучающий набор данных $(x_1 y_1,) \dots (x_n y_n)$. Функция, при которой ожидаемая ошибка классификации принимает минимальное значение, является наиболее оптимальной функцией [7].

Классифицирующая функция имеет следующую формулу (4):

$$f(x) = sign(\langle w, x \rangle + b), \quad (4)$$

где x – объект, принадлежащий пространству;

w – вектор нормали к гиперплоскости;

$\langle , \rangle$ – скалярное произведение;

b – вспомогательный параметр.

Объекты, для которых функция *f(x)* принимает значения равные 1, -1 относятся к разным классам. Необходимо выбрать параметры *w* и *b* таким образом, чтобы расстояние до каждого класса было максимальным.

На практике часто возникают случаи, когда данные невозможно разделить линейным способом. Для решения подобной ситуации выполняется преобразование исходного пространства x в пространство более высокой размерности с помощью специального отображения (5):

$$\varphi: R^n \to X. \quad (5)$$

Отображение выбирается таким образом, чтобы в полученном пространстве множество данных было линейно разделимо. Основная сложность заключается в том, что с увеличением размерности пространства, увеличивается сложность процесса классификации.

Достоинством метода SVM является то, что классификация объектов производится на небольшом наборе данных. Благодаря данному методу можно построить классификатор, способный минимизировать верхнюю оценку ожидаемой ошибки классификации, даже для тех объектов, которые неизвестны и их не было в начальном наборе данных. Суть метода опорных векторов, непосредственно в задаче обнаружения лиц, сводится к поиску гиперплоскости в пространстве признаков, которая будет являться границей между пространством изображений лиц и тех изображений где лица отсутствуют.

К недостаткам метода можно отнести то, что в процессе классификации используется только некоторая часть образцов, находящаяся на границах области классификации. Достоинством же является то, что классификация методом опорных векторов производится на небольшом общем наборе данных.

Эффективность метода может достигать до 90%.

2.5 Метод на основе ключевых точек

Алгоритм разделен на два этапа: получение координат ключевых точек и классификация эмоций на их основе.

Входными данными для метода является набор изображений лиц, в котором каждому изображению соответствует файл разметки, содержащий координаты ключевых точек, выбранных человеком. По этим данным строится две статистические модели:

Модель формы – параметрическая линейная модель, описывающая возможные варианты положения ключевых точек. Формой при этом называется вектор координат ключевых точек (6):

$$s = (x_1, y_1, x_2, y_2, \dots, x_n, y_n,)^T. \quad (6)$$

Модель текстуры – сходная модель, но описывающая уже возможные вариации интенсивности пикселов. Соответственно текстурой называется вектор всех пикселов внутри внешнего контура формы (7):

$$s = (t, t_2, \dots, t_n,)^T. \quad (7)$$

Поскольку на разных изображениях количество пикселов внутри внешнего контура формы может быть разным, перед созданием модели текстуры все изображения лиц приводятся к единой усредненной форме посредствам кусочно-аффинного преобразования: множество точек формы триангулируется, а затем каждый симплекс обычным аффинным преобразованием транслируется в новые координаты.

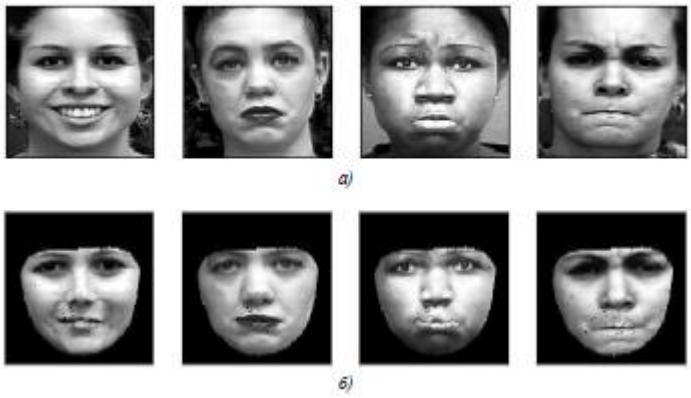
Эффективность метода – 89,4%.

2.6 Метод на основе информации о текстуре

Существует возможность определения (вручную или компьютерными средствами) выражения лица человека даже в отсутствие информации о положении и форме его ключевых элементов.

Другими словами, предполагается, что эмоции на лице кодируются не только положением ключевых точек, но и другими признаками. Проверить это предположение, впрочем, не составляет труда. Для этого достаточно преобразовать лица, выражающие эмоции, к усредненной форме, а затем попытаться определить изначальное выражение. На рисунке 4 изображен ряд лиц, к которым было применено такое преобразование.

Если присмотреться, то можно заметить, что все ключевые точки бровей, глаз, носа, губ, а также подбородка находятся в идентичном положении. Данные изображения получены путем триангуляции по ключевым точкам и применения кусочно-аффинного преобразования к снимкам лиц, выражающих эмоции. Несмотря на выравнивание положения ключевых элементов лица, эмоции на нем все еще легко угадываются по морщинам, теням и прочим элементам текстуры, что полностью подтверждает гипотезу.



а – исходные изображения лиц;

б – соответствующие изображения, приведенные к средней форме

Рисунок 4 – Примеры приведения к средней форме

Как и в методе на основе положения ключевых точек, преобразование к усредненной форме позволяет сформировать для каждого изображения соответствующий вектор текстуры (8):

$$t = (t_1, t_2, \dots, t_n)^T, \quad (8)$$

где n – количество пикселов внутри внешнего контура формы.

Этот вектор можно напрямую использовать в качестве вектора признаков при классификации. В качестве классификатора можно применять SVM.

В итоге получаем следующий алгоритм.

На этапе обучения:

1. Изображения подвергаются предварительной обработке, т.е. с помощью моделей активного образа на них определяются положения ключевых точек, а затем кусочно–афинным преобразованием лица на изображениях приводится к усредненной форме.

2. Пикселы полученных изображений вместе с соответствующими метками эмоций используются для обучения методом SVM.

На этапе применения:

1. Преобразование изображений классифицируется с помощью ранее обученной модели;

2. Преобразованные изображения классифицируется с помощью обученной ранее модели.

Эффективность метода: 86,3%.

2.7 Локальные бинарные шаблоны

Локальные бинарные шаблоны представляют собой бинарные коды определенной разрядности, используемые для классификации в компьютерном зрении. Локальные бинарные шаблоны впервые были предложены в 1996 г. Для анализа текстуры полутоновых изображений. При этом дальнейшие исследования показали, что ЛБШ инвариантны к изменениям в условиях освещения и поворотам изображения.

Локальный бинарный шаблон (ЛБШ) представляет собой описание окрестности пикселя изображения в двоичном представлении. Для

вычисления базового ЛБШ в некоторой точке изображения используется восемь пикселей ее окрестности, а значение интенсивности центрального пикселя принимается в качестве порога. Пиксели со значением интенсивности, большим или равным пороговому значению принимают значения, равные единице, остальные принимают значения, равные нулю. Таким образом, результатом операции является восьмиразрядный бинарный код, который описывает окрестность этого пикселя.

Для более гибкого анализа текстурных особенностей изображения используются круговая окрестность и билинейная интерполяция значений интенсивностей пикселей, которые позволяют построить расширенный ЛБШ с произвольным числом точек P и радиусом R.

Гистограмма ЛБШ. Локальный бинарный шаблон вычисляется для каждого пикселя изображения, а затем строится гистограмма, в которой каждому равномерному коду ЛБШ соответствует отдельный столбец. Также формируется еще один дополнительный столбец, который содержит информацию обо всех неравномерных шаблонах.

Изображения лиц могут быть представлены как набор всевозможных локальных особенностей, которые хорошо описываются с помощью ЛБШ. Однако гистограмма, построенная для всего изображения в целом, кодирует лишь наличие тех или иных локальных особенностей, но при этом не содержит никакой информации об их расположении на изображении. Для учета пространственного расположения изображение разбивается на подобласти, в каждой из которых вычисляется своя гистограмма ЛБШ (рисунок 5). Путем конкатенации этих гистограмм строится общая гистограмма, учитывающая как локальные, так и глобальные особенности изображения.

При таком подходе для лучшего извлечения признаков выполняется варьирование параметров оператора ЛБШ и числа разбиений изображения на подобласти [8].

Эффективность метода – в зависимости от подхода к вычислению меры различия гистограмм ЛБШ эффективность может достигать 86%.

Рисунок 5 – Разбиение изображения лица на подобласти

2.8 Распознавание лиц с помощью сверточных нейронных сетей

В 1998 году исследователи Y. LeCun, L. Bottou, Y. Bengio и P. Haffner предложили вид нейронных сетей, работающих по принципу зрительной системы человека, которые были названы сверточными нейронными сетями (СНС).

СНС представляет собой особый класс многослойного персептрона, который обладает двумерной структурой и хорошо подходит для обработки изображений с высокой степенью инвариантности к смещению, поворотам, масштабированию и другим искажениям входных данных.

Структура СНС представляет собой последовательность из двух типов слоев: сверточные и подвыборочные. Каждый слой состоит из набора плоскостей (карт характеристик), которые в свою очередь состоят из нейронов.

Каждый нейрон сверточного слоя имеет связь с небольшой группой нейронов предыдущего слоя (локальное рецептивное поле). Локальные рецептивные поля нейронов сверточного слоя частично накладываются друг на друга по принципу черепицы. Значения нейронов из локального рецептивного поля умножаются на матрицу синаптических коэффициентов, а

результат записывается в соответствующий нейрон сверточного слоя (рисунок 6).

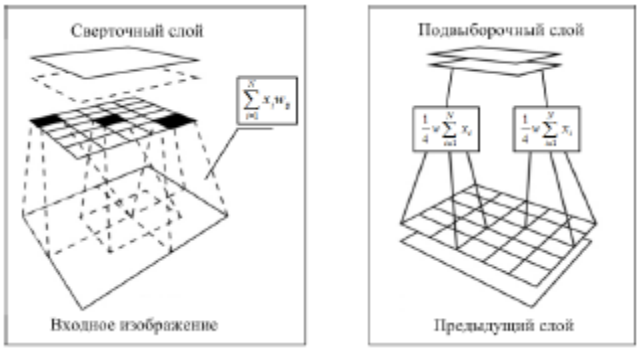


Рисунок 6 – Функционирование слоев в СНС

Следом за сверточным слоем располагается подвыборочный слой, который обеспечивает частичную инвариантность нейронной сети к изменению масштаба входного изображения. Количество плоскостей подвыборочного слоя обычно такое же, как и в предыдущем слое. В подвыборочном слое локальные рецептивные поля не пересекаются друг с другом и имеют фиксированный размер 2x2 нейрона. Каждый нейрон данного слоя вычисляет среднее значение своих четырех входов, умножает их на синаптические коэффициенты и полученный результат передает через функцию активации. Таким образом, подвыборочный слой уменьшает размерность плоскостей предыдущего слоя в два раза.

Последовательно чередуясь друг за другом, размеры плоскостей уменьшаются, но их количество увеличивается. Чередование слоев позволяет формировать различные карты характеристик, что наделяет СНС способностью к идентификации более сложных иерархических признаков. Постепенно при прохождении нескольких слоев карта признаков вырождается в вектор. Последние несколько слоев СНС представляют собой классический персептрон, состоят из обычных нейронов и предназначены для классификации выделенных признаков.

Важным свойством СНС является использование общей матрицы весовых коэффициентов для всех нейронов в пределах плоскости. Данный подход позволяет использовать меньшее число настраиваемых параметров (весовых коэффициентов) при большом количестве связей и повысить скорость процесса обучения. Таким образом, плоскости СНС представляют собой фильтры, каждый из которых осуществляет поиск индивидуальных характерных признаков входного изображения. Это позволяет сверточной нейронной сети запоминать взаимосвязь пространственно зависимых областей изображения. Характерные признаки, извлекаемые той или иной плоскостью, определяются в процессе обучения.

Если входное изображение искажено или смещено, то на выходе плоскости будет аналогично смещенный результат. Благодаря этому свойству обеспечивается устойчивость СНС к искажениям входных данных.

Недостаток использования СНС заключается в сложности настройки оптимальных параметров: количество слоев, плоскостей, нейронов, размер рецептивного поля и т.д.

Преимущества использования СНС для детектирования объектов на изображениях:

- структура СНС хорошо подходит для обработки двумерных данных.
- наслаивающиеся друг на друга рецептивные поля обеспечивают взаимосвязь пространственно зависимых областей изображения.
- повышенная устойчивость к аффинным и проекционным искажениям входных данных, шумам, изменению масштаба. Благодаря своим преимуществам на сегодняшний день СНС активно используются для обнаружения лиц и других объектов на изображениях и видео последовательностях.

Результаты: процент распознавания при наилучшем исходе – 97,4%.

3 Программа распознавания эмоций

Ранее были рассмотрены методы обнаружения и распознавания объектов. По итогу необходимо разработать программу, позволяющую обнаруживать лица и распознавать эмоции.

Были определены следующие задачи:

- программа должна обнаруживать лицо и распознавать эмоции в реальном времени;
- в качестве эмоций для распознавания определены следующие базовые эмоции: нейтрально, улыбка, грусть, удивление отвращение, злость;
- для выполнения задачи распознавания эмоций программа должна иметь два режима: режим обучения и режим тестирования (распознавания эмоций);

С учётом перечисленных задач были определены следующие инструменты:

- язык программирования Python;
- метод Виолы-Джонса в качестве метода обнаружения лиц;
- метод главных компонентов;
- использование многослойного персептрона (MLP) для распознавания эмоций на обнаруженном лице;
- библиотека OpenCV, включающая в себя выше перечисленные алгоритмы.

Но обо всё по порядку.

3.1 Язык программирования Python

Язык Python является, пожалуй, самым простым в изучении и самым приятным в использовании из языков программирования, получивших широкое распространение. Программный код на языке Python легко читать и писать, и, будучи лаконичным, он не выглядит загадочным. Python – очень

выразительный язык, позволяющий уместить приложение в меньшее количество строк, чем на это потребовалось бы в других языках, таких как C++ или Java.

Python является кросс-платформенным языком: обычно одна и та же программа на языке Python может запускаться и в Windows, и в UNIX-подобных системах, таких как Linux, BSD и Mac OS, для чего достаточно просто скопировать файл или файлы, составляющие программу, на нужный компьютер; при этом даже не потребуется выполнять «сборку», или компилирование программы. Конечно, можно написать на языке Python программу, которая будет использовать некоторые характерные особенности конкретной операционной системы, но такая необходимость возникает крайне редко, т. к. практически вся стандартная библиотека языка Python и большинство библиотек сторонних производителей обеспечивают полную кросс-платформенность [9].

Одним из основных преимуществ языка Python является наличие полной стандартной библиотеки, позволяющей обеспечить загрузку файла из Интернета, распаковку архива или создание веб-сервера посредством написания нескольких строк программного кода. В дополнение к ней существуют тысячи дополнительных библиотек сторонних производителей, среди которых одни обеспечивают более сложные и более мощные возможности, чем стандартная – например, библиотека для организации сетевых взаимодействий Twisted и библиотека для решения вычислительных задач NumPy; а другие предоставляют функциональность, которая слишком узконаправленно специализирована, чтобы ее можно было включить в стандартную библиотеку – например, пакет моделирования SimPy.

Python может использоваться для программирования в процедурном, в объектно-ориентированном и, в меньшей степени, в функциональном стиле программирования, хотя в глубине души Python – объектно-ориентированный язык программирования [10].

3.2 Метод Виолы-Джонса

Метод Виолы-Джонса – алгоритм, позволяющий обнаруживать объекты на изображениях в реальном времени. Метод был разработан Полом Виолой и Майклом Джонсом и представлен в 2001 году.

Основные достоинства метода:

- до сих пор является эффективным методом для поиска объектов на изображениях и видео;
- обладает крайне низкой вероятностью ложного обнаружения лица;
- хорошо работает и обнаруживает лица даже при наблюдении объекта под небольшим углом (до 30°).

Основные недостатки:

- при угле наклона больше 30° вероятность обнаружения лиц резко падает;
- вероятность верного срабатывания также падает если ни лице присутствуют очки, усы, борода.

Основные принципы, на которых основан метод:

- интегральное представление изображений;
- признаки Хаара;
- построение классификатора на основе алгоритма бустинга;
- комбинирование классификаторов в каскадную структуру;

Рассмотрим подробнее.

Интегральное представление изображений. Интегральное представление изображений используется для того, чтобы рассчитать яркость прямоугольного участка изображения. Интегральное представление позволяет быстро рассчитывать суммарную яркость произвольного прямоугольника, причём время расчёта не зависит от площади прямоугольника [11].

Интегральное представление изображения представляет собой матрицу, совпадающую по размерам с исходным изображением. В каждом её элементе

храниться сумма интенсивностей всех пикселей, находящихся левее и выше данного элемента. Элементы матрицы рассчитываются по формуле (9):

$$I(x,y) = \sum_{x' \le x, y' \le y} i(x',y'), \quad (9)$$

где $I(x,y)$ – значение точки (x,y) интегрального изображения;

$i(x,y)$ – значение интенсивности исходного изображения.

На основе применения интегрального представления изображения вычисление признаков одинакового вида, но с разными геометрическими параметрами, происходит за одинаковое время.

Каждый элемент матрицы $I(x,y)$ представляет собой сумму пикселей в прямоугольнике от $i(0,0)$ до $i(x,y)$, т.е. значение каждого элемента $I(x,y)$ равно сумме значений всех пикселей левее и выше данного пикселя $i(x,y)$. Расчёт матрицы занимает линейное время, пропорциональное числу пикселей в изображении и его можно производить по следующей формуле (10):

$$I(x,y) = i(x,y) - I(x-1,y-1) + I(x,y-1) + I(x-1,y). \quad (10)$$

Интегральное представление имеет интересную особенность. По интегральной матрице можно очень быстро вычислить сумму пикселей произвольного прямоугольника.

Признаки Хаара. Признаки Хаара или Хаар-подобные характеристики, позволяют обнаруживать и распознавать объекты в режиме реального времени. Представляют собой результат сравнения яркостей в двух прямоугольных областях изображения.

Предположим, что задано множество объектов A и множество допустимых объектов B. Пусть $g:A \to B$ называется решающей функцией. Решающая функция g должна допускать эффективную компьютерную реализацию, по этой причине её также называют алгоритмом. Признак (*feature*) f объекта a – отображение $f:A \to D_f$, где D_f – множество допустимых значений признака. В частности, любой алгоритм $g:A \to B$ также можно рассматривать как признак. Если задан набор признаков $f_1, \ldots, f_n$, то вектор (11)

$$x = (f_1(a), \ldots, f_n(a)) \quad (11)$$

называется признаковым описанием объекта $a \in A$.

Признаковые описания допустимо отождествлять с самими объектами. При этом множество (12) называют признаковым пространством:

$$A = D_n \times \ldots \times D_{fn}. \quad (12)$$

В стандартном методе Виолы-Джонса используются прямоугольные признаки, называемые примитивами Хаара. (рисунок 7).

В расширенном методе Виолы-Джонса, представленном в библиотеке OpenCV, используются дополнительные признаки (рисунок 8).

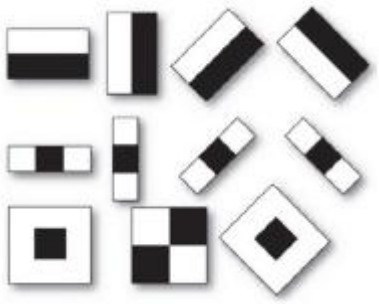


Рисунок 7 – Примитивы признаков Хаара

Вычисляемым значением такого признака (13) будет:

$$F = U - V, \quad (13)$$

где U – сумма значений яркостей точек ,закрываемых свелой частью признака;

V – сумма значений яркостей точек, закрываемых тёмной частью признака.

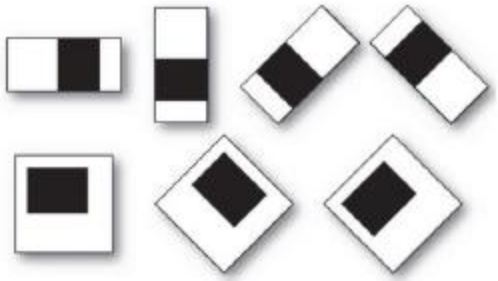


Рисунок 8 – Дополнительные признаки Хаара

Для их вычисления используется понятие интегрального изображения. Хаар-подобные признаки описывают значение перепада яркости по оси X и Y изображения соответственно.

Построение классификатора на основе алгоритма бустинга. Бустинг – комплекс методов, способствующих повышению точности аналитических моделей. Бустинг (boosting) означает дословно «усиление» «слабых» моделей – это процедура последовательного построения композиции алгоритмов машинного обучения, когда каждый следующий алгоритм стремится компенсировать недостатки композиции всех предыдущих алгоритмов. Идея бустинга была предложена Робертом Шапиро (Schapire) в конце 90-х гг. прошлого века, когда надо было найти решение
вопроса о том, каким образом имея множество плохих (незначительно отличающихся от случайных) алгоритмов обучения, получить один хороший.

В результате работы алгоритма бустинга на каждой итерации формируется простой классификатор вида, имеющий следующую формулу (14):

$$h_j(z)\begin{cases}1, & \text{если } p_j f_j(z) < p_j \theta_j \\ 0, & \text{иначе}\end{cases} \tag{14}$$

где p_j – показывает направление знака неравенства;

θ_j – значение порога;

$f_j(z)$ – вычисленное значение признака;

z – окно изображения размером 24×24 пикселей.

Полученный классификатор имеет минимальную ошибку по отношению к текущим значениям весов, задействованным в процедуре обучения для определения ошибки.

Комбинирование классификаторов в каскадную структуру. Каскадная структура повышает скорость обнаружения, фокусируя свою работу на наиболее информативных областях изображения.

Структура каскадного детектора приведена на рисунке 9. Каскад состоит из слоёв, которые представляют собой классификаторы, обученные с помощью процедуры бустинга.

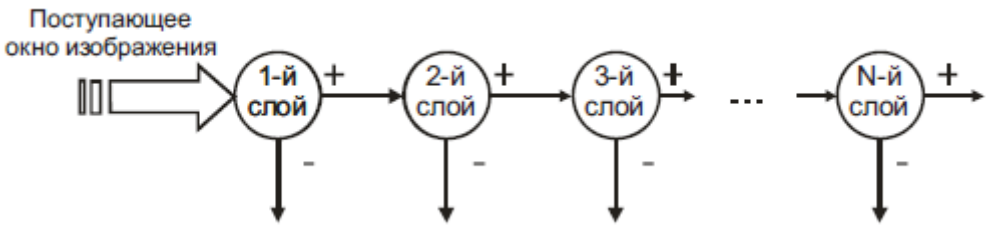


Рисунок 9 – Структура каскадного детектора

3.3 Метод главных компонент

Метод главных компонент (Principal Component Analysis, PCA) – метод сокращения размерности. Этот подход в задаче распознавания лиц был впервые применен M. Turk и A. Pentland в 1991 году и получил названия собственные лица. Он заключается в линейном ортогональном преобразовании входного вектора P размерности N в выходной вектор Q размерности M, M<N. Компоненты вектора Q являются некоррелированными, и общая дисперсия после преобразования остаётся неизменной [12].

Рассмотрим набор данных, из которого мы извлекли ровно две характеристики. Этими характеристиками могут быть значения оттенков серого в пикселях в некоторых положениях *x* и *y*. Если мы построим набор данных по этим двум признакам, данные могут лежать в пределах некоторого многомерного гауссова пространства.

PCA поворачивает все точки, пока выделенные вектора не совпадут с осями, которые представляют собой направление распространение данных. PCA считает, что данные оси позволяют предоставить данные указанные на изображении наиболее информативным образом.

Основной недостаток данного метода заключается в том, что он слишком чувствителен к применению условий освещенности и в реальных условиях в качестве похожих изображений выдаёт изображения, полученные при похожих условиях освещения, а не изображения похожих людей.

3.4 Многослойный персептрон

Многослойные персептроны (multi-layer perceptron) – искусственная нейронная сеть (ANN) используемая для преобразования набора входных данных для преобразования в набор выходных данных. В основе своей MLP является персептроном, напоминающим упрощенный биологический нейрон. Объединяя большое количество персептронов во множество слоёв, MLP производит нелинейные решения его входных данных. Кроме того, MLP возможно обучить с помощью метода обратного распространения ошибки, что делает их очень интересными для контролируемого обучения.

После того, как определен один персептрон, целесообразно объединить несколько персептронов для формирования большой сети.

Многослойные персептроны обычно состоят минимум из трёх слоёв, где первый слой имеет узел для каждой из функций набора данных, а последний слой имеет узел для каждой метки класса. Слой между ними называется скрытым слоем. Пример такой нейронной сети с обратной связью показан на рисунке 10.

В нейронной сети с обратной связью подключены некоторые или все узлы входного уровня для всех узлов в скрытом слое, а некоторые или все узлы в скрытом слое связанных с некоторыми или всеми узлами выходного уровня. Обычно количество узлов во входном слое должно быть равно количеству признаков в наборе данных, потому что каждый узел представляет одну

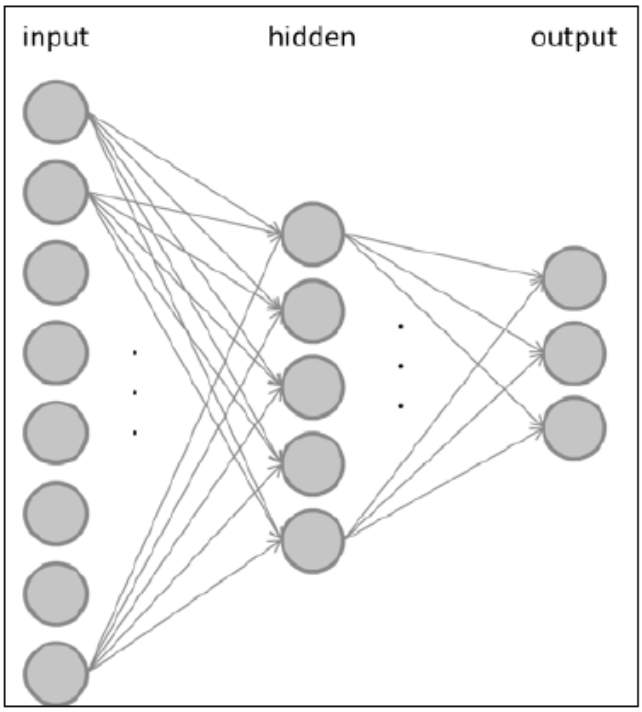


Рисунок 10 – Пример многослойной нейронной сети

функцию. Аналогично, количество узлов на выходе уровень обычно равно количеству классов данных, поэтому, когда входной образец класса i , i -й узел в выходном слое активен, а все остальные нет.

Конечно, возможно также иметь несколько скрытых слоев. Часто заранее не известно какой должен быть оптимальный размер сети.

Как правило, частота ошибок на обучающем наборе уменьшается, когда добавляется больше нейронов в сеть, как показано на следующем рисунке 12 (более тонкая, красная кривая).

Связано это с тем, что сложность модели увеличивается с увеличением нейронной сети. Однако, тоже самое нельзя сказать о частоте ошибок на тестовом наборе (синяя кривая). Вместо этого, с увеличением сложности модели ошибка теста проходит через минимум, и добавление большего количества нейронов в сеть больше не улучшит характеристики.

Таким образом, необходим размер нейронной сети оптимального диапазона, в котором сеть достигает наилучшего представления.

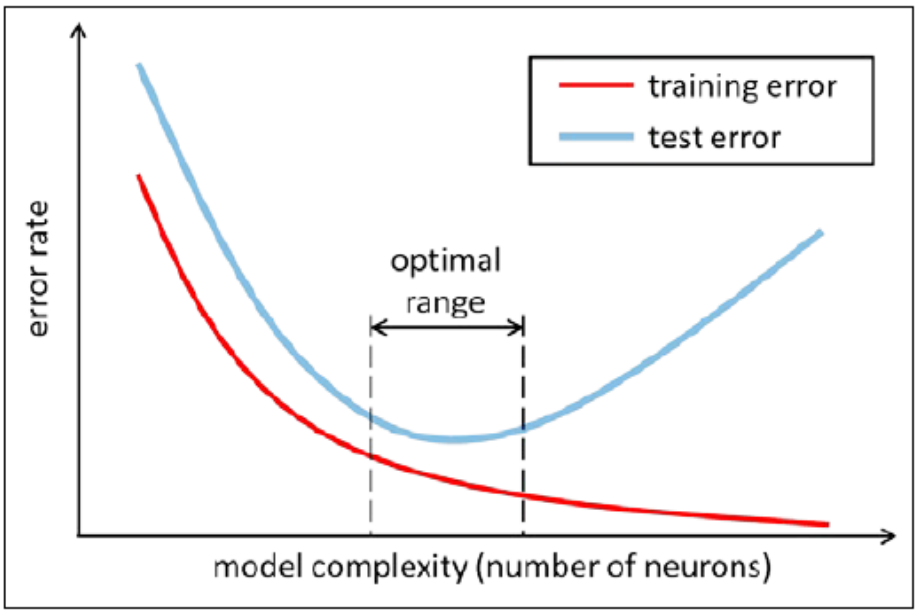


Рисунок 11 – Зависимость эффективности нейронной сети от количества нейронов

MLP учится, настраивать размер нейронной сети так, что, когда входной образец класса *i* представлен, *i* -й узел в выходном слое активен, а все остальные нет. MLP обучен посредством обратного распространения ошибки, который является алгоритмом для вычисления частных производных функции потерь по отношению к любому синаптическому весу или смещению нейрона в сеть. Эти частные производные могут затем использоваться для обновления весов и смещений в сети, чтобы поэтапно снизить общую потерю.

Функция потерь может быть получена путем представления учебных образцов в сеть и путем наблюдения за выходными данными сети. Наблюдая за тем, какие выходные узлы активны и в каких мы можем рассчитать относительную ошибку между тем, что делает выходной уровень, и тем, что он должен выполнять (т. е. функцию потерь). Затем мы вносим поправки ко всем весам в сети, чтобы ошибка со временем уменьшалась. Оказывается, ошибка в скрытом уровне зависит от выходного уровня, а ошибка во входном слое зависит от ошибки как в скрытом слое, так и выходном слое. Таким образом, в некотором смысле ошибка распространяется через сеть [13].

3.5 Библиотека OpenCV

OpenCV - библиотека компьютерного зрения с открытым исходным кодом. Библиотека написана на C и C++ и работает на компьютерах под управлением Linux, Windows, Mac OS X. Так же активно развиваются интерфейсы библиотеки для Python, Ruby, Matlab и других языков программирования.

Библиотека OpenCV была разработана с целью повышения вычислительной эффективности и с уклоном на приложения реального времени. OpenCV написана с использованием оптимизированного C и может использовать многоядерные процессоры. В дальнейшем, если потребуется автоматическая оптимизация на аппаратных платформах Intel, существует возможность покупки библиотеки IPP (Integrated Perfomance Primitives), которая состоит из процедур с низкоуровневой оптимизацией для различных алгоритмических областей. OpenCV будет автоматически использовать библиотеку IPP во время выполнения программы [14].

Одной из основных целей OpenCV является предоставление простого в использовании интерфейса, который позволит людям довольно таки быстро строить сложные приложения, использующие компьютерное зрения. Библиотека OpenCV содержит более 500 функций, которые охватывают многие области компьютерного зрения, такие как: инспекция фабричной продукции, медицина, безопасность, пользовательский интерфейс, калибровка камеры, стереозрение и робототехника. И все это благодаря тому, что компьютерное зрение и машинное обучение часто идут "рука об руку", к тому же OpenCV полностью включает в себя библиотеку общего назначения MLL (Machine Learning Library). MLL библиотека ориентирована на распознавание статических образов и кластеризацию. MLL очень полезна для задач компьютерного зрения, которые составляют основу OpenCV, но она довольно-таки обобщенная, чтобы решать конкретные проблемы машинного обучения.

3.6 Структура, принцип работы и интерфейс программы

Программа будет состоять из следующих компонентов.

face_recognize.py – основной сценарий.

FaceLayout – пользовательский макет интерфейса, работающий в двух различных режимах:

- режим обучения – в данном режиме программа собирает кадры изображений, обнаруживает на них лицо, указывается эмоция, в зависимости от выражения лица и после выхода из программы происходит сохранение образцов в файле, чтобы их в дальнейшем можно было проанализировать с помощью класса *datasets.homebrew*;

- режим тестирования – в данном режиме программа обнаруживает лицо в каждом видеокадре и предсказывает соответствующую метку класса эмоции, используя предварительно подготовленный MLP;

Main – основная функция для запуска приложения GUI.

detectors.FaceDetector – класс для распознавания лиц. Имеет следующие методы:

- *detect* – метод обнаружения лиц на изображении в градации серого, если это необходимо изображение масштабируется для повышение вероятности обнаружения лица, при успешном обнаружении метод возвращает извлеченный участок головы;

- *align_head* – метод обработки выделенной области головы с помощью кусочно-афинных преобразований, для того, чтобы лицо было центрировано;

classifiers.Classifier – абстрактный базовый класс, который определяет общий интерфейс для всех классификаторов.

classifiers.MultiLayerPerceptron – класс, реализующий MLP, используя следующие методы:

- *fit* – метод, проверяющий соответствие MLP учебным данным. В качестве входных данных принимает матрицу с данными обучения, где каждая

строка является образцом обучения, а столбцы содержат функцию значения и вектор меток;

- *evaluate* – метод оценки MLP путём его сравнения с некоторыми тестовыми данными. В качестве входных данных принимает матрицу с тестовыми данными, где каждая строка является тестовым образцом, а столбцы – значения признаков и вектор меток. Функция возвращает три показателя производительности: правильность, точность и время отклика;

- *predict* – метод прогнозирования меток классов некоторых тестовых данных. Метод полезен тем, что его можно применять к любому количеству выборки данных;

- *save* – метод, сохраняющий обученный MLP в файл;

- *load* – метод загрузки предварительно подготовленного MLP из файла;

train_test_mlp – сценарий для обучения и тестирования MLP, применяемый для записанного набора данных.

datasets.homebrew – класс, анализирующий обучающие наборы. Содержит следующие методы:

- *load_data* – метод загрузки учебного набора, разделяющий через функцию *extract_features* данные на тренировочные и на испытательные;

- *load_from_file* – метод загрузки ранее сохранённого предварительно обработанного набора данных;

- *extract_features* – метод извлечения выбранной функции;

gui – модуль, необходимый для доступа к устройству видеозахвата.

gui.BaseLayout – общий макет интерфейса.

Сначала программа обнаруживает лицо на изображении. Для обнаружения лиц используется ряд предварительно подготовленных классификаторов, в формате xml, поставляемых вместе с OpenCV. Ниже в таблице 1 перечислены классификаторы и файл xml.

Таблица 1 – Классификаторы используемые в работе

Классификатор	Имя файла XML
Лицевой детектор (по умолчанию)	haarcascade_frontalface_default.xml
Лицевой детектор (быстрый Хаар)	haarcascade_frontalface_alt2.xml
Детектор левого глаза	haarcascade_lefteye_2splits.xml
Детектор правого глаза	haarcascade_righteye_2splits.xml

Изначально программа находит лицо и глаза на изображении, причём обнаружения левого и правого глаза производится отдельно. Это сделано для того, чтобы можно было также распознавать лица в очках.

Обнаружение лица описано в сценарии *detectors.py*, в классе *FaceDetector*, с помощью метода *detect*. Метод определяет грани во входном RGB-изображении, возвращая True при успехе (иначе False), рисует ограничивающий прямоугольник головы на входное изображение (кадр) и извлекает область с головой. Ниже, на рисунке 12 сценарий метода.

```
def detect(self, frame):

    frameCasc = cv2.cvtColor(
        cv2.resize(
            frame,
            (0, 0),
            fx=1.0 / self.scale_factor,
            fy=1.0 / self.scale_factor),
        cv2.COLOR_RGB2GRAY)
    faces = self.face_casc.detectMultiScale(
        frameCasc,
        scaleFactor=1.1,
        minNeighbors=3,
        flags=cv2.cv.CV_HAAR_FIND_BIGGEST_OBJECT) * self.scale_factor


    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x + w, y + h), (100, 255, 0), 2)
        head = cv2.cvtColor(frame[y:y + h, x:x + w],
                            cv2.COLOR_RGB2GRAY)
        return True, frame, head, (x, y)

    return False, frame, None, (0, 0)
```

Рисунок 12 – Иллюстрация метода detect

После того как лицо обнаружено, прежде чем применять к нему классификацию, происходит обработка предварительно извлеченного участка головы. Хотя каскад лица довольно точен, для распознавания эмоций важно чтобы лицо на изображении было центрировано. Для того, чтобы извлеченная область головы была центрирована используется метод *align_head*. Метод выравнивает область головы с использованием аффинных преобразований, обрабатывая извлеченную область головы путем вращения и масштабируя его так, чтобы лицо было расположено по центру. Метод возвращает True при успешном выполнении (иначе False). Метод может не работать из-за плохих условий освещения.

Захват изображения с видеокамеры происходит в сценарии *face_recognize.py* с помощью модуля *cv2.VideoCapture* в классе *FaceLayout*, в методе *main*. Код модуля изображен на рисунке 13.

```
def main():
    capture = cv2.VideoCapture(0)
    if not(capture.isOpened()):
        capture.open()

    if hasattr(cv2, 'cv'):
        capture.set(cv2.cv.CV_CAP_PROP_FRAME_WIDTH, 640)
        capture.set(cv2.cv.CV_CAP_PROP_FRAME_HEIGHT, 480)
    else:
        capture.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
        capture.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)

    # start graphical user interface
    app = wx.App()
    layout = FaceLayout(capture, title='Распознавание лиц')
    layout.init_algorithm()
    layout.Show(True)
    app.MainLoop()
```

Рисунок 13 – Иллюстрация модуля cv2.VideoCapture

Интерфейс программы. Макет графического интерфейса основан на методе *_create_custom_layout*. Создаётся панель для видеокадра и рядом с ней находится ряд кнопок.

Идея состоит в том, чтобы щёлкнуть одну из шести кнопок для того, чтобы указать какое из шести выражений лица указано на изображении, затем после выбора нужной эмоции нажать кнопку «сделать снимок» для моментального снимка.

Также среди кнопок присутствуют два переключателя, для выбора режима работы программы: в режиме обучения или в режиме тестирования (распознавания эмоций). Ниже, на рисунке 14 изображён интерфейс кнопок.

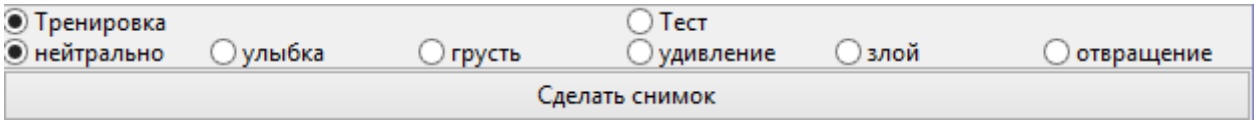


Рисунок 14 – Иллюстрация интерфейса программы

Когда происходит нажатие кнопки «сделать снимок», происходит вызов метода *_on_snapshot*. Этот метод фиксирует эмоцию, которую мы хотим записать, проверяя значения всех переключателей и присваивает соответствующую метку класса. Ниже, на рисунке 15 изображён код сценария.

```
def _on_snapshot(self, evt):

    if self.neutral.GetValue():
        label = 'нейтрально'
    elif self.happy.GetValue():
        label = 'улыбка'
    elif self.sad.GetValue():
        label = 'грусть'
    elif self.surprised.GetValue():
        label = 'удивление'
    elif self.angry.GetValue():
        label = 'злость'
    elif self.disgusted.GetValue():
        label = 'отвращение'

    if self.head is None:
        print "Лицо не обнаружено"
    else:
        success, head = self.faces.align_head(self.head)
        if success:
            print "Добавлен образец для тренировочного набора"
            self.samples.append(head.flatten())
            self.labels.append(label)
        else:
            print "Не удалось выровнять голову (не удалось обнаружить глаза)"
```

Рисунок 15 – Иллюстрация сценария _on_snapshot

После выхода из программы запускается событие *EVT_CLOSE*, которое привязано к методу *_on_exit*. Этот метод сбрасывает собранные образцы и метки классов в файл. При этом, происходит проверка в случае случайного перезаписывания данных. Если имя файла уже существует, то файл нумеруется и данные сохраняются в новый файл. После выгрузки данных в файл выпадает сообщение о том, что все элементы структуры данных правильно сохранены. Ниже, на рисунке 16 изображён код сценария.

```
def _on_exit(self, evt):

    if len(self.samples) > 0:

        if path.isfile(self.data_file):

            load_from_file, fileext = path.splitext(self.data_file)
            offset = 0
            while True:
                file = load_from_file + "-" + str(offset) + fileext
                if path.isfile(file):
                    offset += 1
                else:
                    break
            self.data_file = file

        f = open(self.data_file, 'wb')
        pickle.dump(self.samples, f)
        pickle.dump(self.labels, f)
        f.close()

        print "Сохранено", len(self.samples), "образцы для", self.data_file

    self.Destroy()
```

Рисунок 16 – Иллюстрация сценария _on_exit

Далее происходит анализ загруженных данных с помощью метода главных компонент. Проанализированные данные хранятся в *homebrew.py*.

В заключение обучается и тестируется классификатор MLP с помощью сценария *train_test_mlp.py*. Сценарий сначала анализирует набор данных из *homebrew.py* и извлекает все метки класса [15].

3.7 Тестирование и оценка эффективности распознавания

Результаты обнаружуения лица на изображении очень высоки. Программа распознаёт лицо:

- при небольших наклонах и поворотах головы;
- при наличии посторонних объектов на изображении;

- при закрытых глазах и наличии очков;
- распозаёт лица на фотографиях (например фотография модели на обложке модного журнала или фотография с телефона);
- обнаруживает нарисованное лицо на листе бумаги.

Ниже на рисунке 18 показаны результаты обнаружения лица.

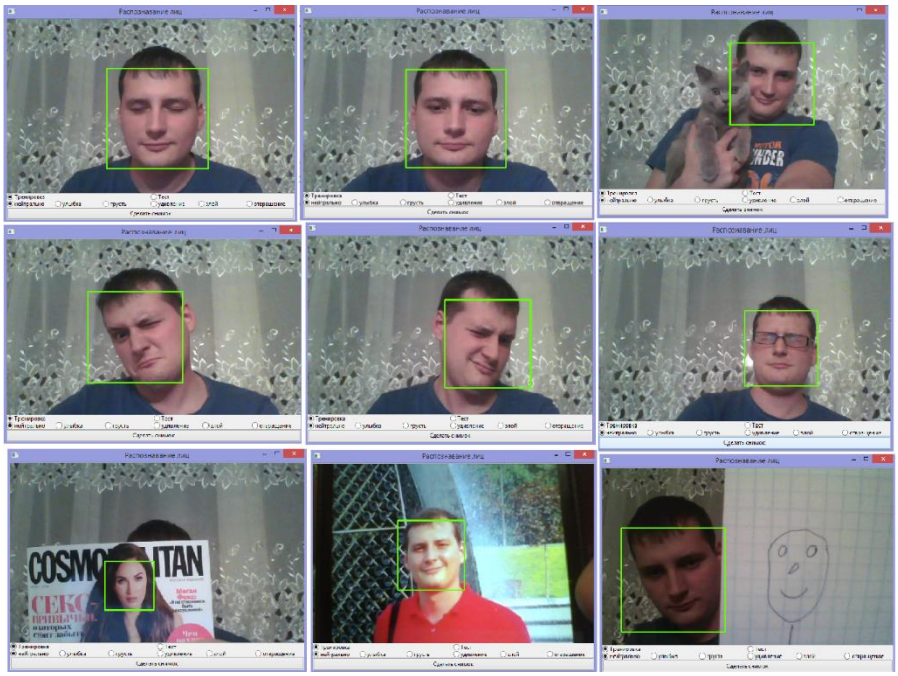


Рисунок 17 – Результат обнаружения лица на изображении

После обнаружения лица происходит выделение головы зеленой рамкой, это значит что можно приступать к следующему шагу: фиксации данных и тренировки программы для возможности автоматически распознавать эмоции на изображении.

Таже была произведена попытка вызвать ложно распознавание. Для этого на клетчатом листе были нарисованы черты лица.

Далее проведён ряд тестов, с целью оценить эффективность работы программы.

Тест №1. В качестве первого теста было сделано 12 фотографий, по 2 на каждую эмоцию. В итоге программа верно определила 5 из 6 эмоций, не распознав только удивление, перепутав его с отвращением.

Однако не все эмоции одинаково хорошо распознавались. Например, нейтральное выражение лица начинало хорошо распознаваться только лишь при легком наклоне головы вперёд, иначе эмоция на 100% не определялась, указывая что выражения лица либо злое, либо нейтральное.

Улыбка распознавалась только лишь при небольшом повороте головы вправо, а удивление, как обозначалось ранее вообще не удалось распознать.

Также была проведена оценка эффеткивности распознавания эмоций. За 30 секунд подсчитывалось количество неверных распознаваний поочерёдно, для каждой из эмоций. Далее вычислялся процент неверных распознаваний и отнимался от общего процента числа измерений. Также для проведения измерения было допущено, что за 1 секунду происходит только одно распознавание.

Т.е. оценку эффективности программы можно определить с помощью формулы (15):

$$p = \frac{I_{\text{общ}} - I_{\text{н}}}{I_{\text{общ}}}, \tag{15}$$

где $I_{\text{общ}}$ – общее количество измерений;

$I_{\text{н}}$ – количество неверных распознаваний.

В таблице 2 ниже указаны численные данные и процент ошибочно сработанных распознаваний.

Таблица 2 – Результаты первого тестирования

	нейтрально	улыбка	грусть	удивление	злость	отвращени	Сумма
Общее кол-во измерений	30	30	30	30	30	30	180
Кол-во неверных распознаваний	15	16	6	30	13	11	91
Процент верного распознавания	50,00	46,67	80,00	0,00	56,67	63,33	49,44

Как видно из таблицы процент верно распознанных эмоций весьма невелик – всего 49,44%. Связано это с недостатком данных для обучения.

Ниже на рисунке 18 показаны результаты работы программы, обученной на 12 фотографиях.

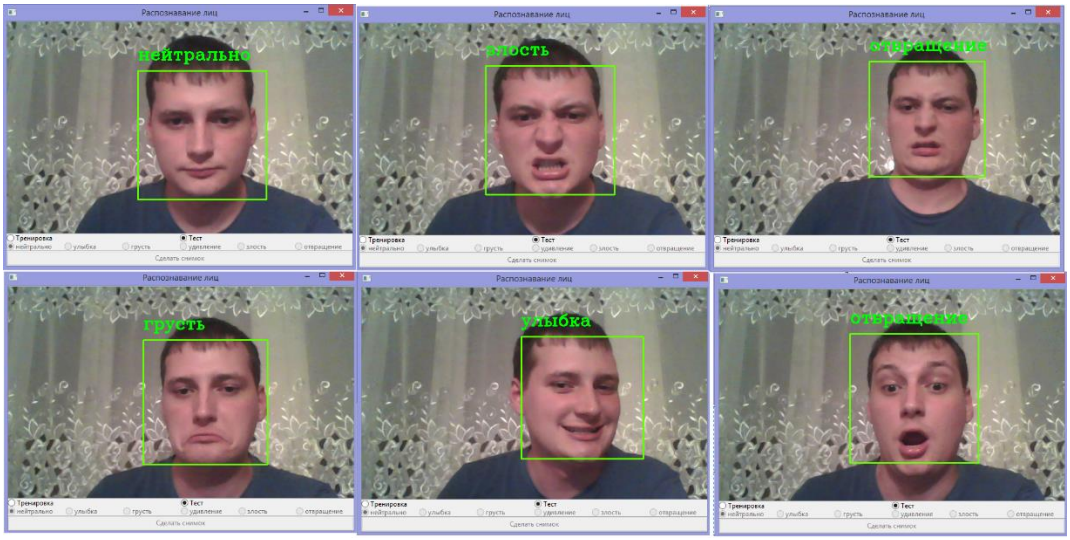


Рисунок 18 – Результат работы программы

Тест №2. Всего было сделано 30 снимков, по 5 фотографий на каждую эмоцию. Как и при первом тестировании удалость распознать 5 из 6 эмоций. На этот раз распознаванию не поддалось отвращение.

Однако, из-за вариативности исходных данных (снимки были сделаны при разных ракурсах и наклонах головы) удалось увеличить тоность распознаваний. В таблице 3 ниже указаны результаты измерений.

В данном тестировании процент распознанных эмоций уже значитлеьно выше – 57,78%. Наиболее точно распознавалась улыбка на лице – 80% верных распознаваний. Нейтральное, удивленное и злое выражение лица – попали в среднюю категорию. Весьма плохо распознавалась грусть и совсем, как уже отмечалось ранее, не удалось распознать отвращение.

Таблица 3 – Результат второго тестирования

	нейтрально	улыбка	грусть	удивление	злость	отвращение	Сумма
Общее кол-во измерений	30	30	30	30	30	30	180
Кол-во неверных распознаваний	7	6	13	8	12	30	76
Процент верного распознавания	76,67	80,00	56,67	73,33	60,00	0,00	57,78

ЗАКЛЮЧЕНИЕ

В ходе исследования получены следующие результаты:

1 Произведён обзор современного рынка систем распознавания эмоций, рассмотрены их особенности. Основные программы, рассмотренные в разделе: FaceReader, Affective Computing Research Group, Microsoft API Emotion Recognition. Среди рассмотренных программ нет явного лидера по эффективности распознавания эмоций.

2 Рассмотрены разнообразные методы распознавания эмоций, описан принцип работы, их особенности, недостатки и эффективность. В среднем эффективность рассмотренных методов 86 – 90%. Наиболее эффективным методом оказалось распознавание лиц и мимики с помощью свёрточных нейронных сетей. Точность распознавания данного метода составляет 97,4%.

3 На основе описанных методов и готовых программных средств разработан алгоритм, реализованный в программе обнаруживающей и распознающей эмоции на лице. Проведена оценка эффективности программы – тестирование, в ходе которого отслеживалось количество верно распознанных эмоций.

В результате выполнения курсового проекта освоены следующие компетенции:

- при подготовке обзора методов распознавания эмоций были поставлены задачи, решённые в последствии с помощью современных информационных технологий, что позволило освоить компетенцию ПК-1;

- в процессе построения программного кода, выполняющего распознавание эмоций, использовались знания в области компьютерных технологий и решались задачи построения алгоритмов детектирования лиц и распознавания эмоций, что позволило освоить компетенцию ПК-5;

- в ходе подготовки текста курсового проекта, разработке презентации и защиты результатов курсового проекта была освоена компетенция ОПК-1;

- при подготовке, поиске и систематизации статей, публикаций, научной литературы для курсового проекта достигнута компетенция ОК-3.

Результатом данной работы является алгоритм распознавания эмоций, реализованный в программе.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Горбунова Е. С. Реализация интеллектуальной системы распознавания эмоций с применением нейронных сетей: дис… студент 25.16.05 / Е. С. Горбунова; Уральский федеральный университет имени первого Президента России Б.Н.Ельцина – Екатеринбург, 2017. – 80 с.

2 Макаренко А. А. Методика локализации изображения лица для систем видеоконтроля на основе нейронной сети / А. А. Макаренко, В. Т. Калайда // Известия Томского политехнического университета. – 2006. – Т. 309. - № 309. – С. 113–118.

3 Исследование рынка систем распознавания эмоций – (Рус.). – URL: https://habrahabr.ru/post/133686/ [10 октября 2017].

4 API распознавания эмоций – средство определения эмоций | Microsoft Azure – (Рус.). – URL: https://azure.microsoft.com/ru-ru/services/cognitive-services/emotion/ [10 октября 2017].

5 Заболеева-Зотова А. В. Развитие системы автоматизированного определения эмоций и возможные сферы применения / А. В. Заболеева-Зотова, Ю. А. Орлова, А. С. Бобков // Открытое образование. – 2011. – Т. 2. – С. 59–62.

6 Паттерн шаблонный метод (template method) – Блог программиста – (Рус.). – URL: https://pro-prof.com/archives/1108 [14 декабря 2017].

7 Санников К. А. Разработка алгоритмов распознавания эмоционального состояния человека по изображению его лица: дис… студент 09.04.01 / К. А. Санников; Национальный Исследовательский Томский Политехнический Университет. – Томск, 2017. – 98 с.

8 Пару слов о распознавании образов / Хабрахабр – (Рус.). – URL: https://habrahabr.ru/post/208090/ [18 февраля 2018].

9 Прохоренок, Н. А. Python 3 и PyQt 5. Разработка приложений / Н. А. Прохоренок, В. А. Дронов. — СПб.: БХВ-Петербург, 2016. – 832 с.

10 Саммерфилд М. Программирование на Python 3. Подробное руководство / М. Саммерфилд. – Спб.: Символ-Плюс, 2009. – 608 с.

11 Viola P. Rapid Object Detection using a Boosted Cascade of Simple Features / P. Viola, M. Jones // Accepted Conference on Computer Vision and Pattern Recognition. – 2001. – P. 1–9.

12 Потапов А. С. Системы Компьютерного Зрения: [Учебное пособие] / А. С. Потапов. – Санкт-Петербург, 2016. – 162 с.

13 Коэльо Л. П. Построение систем машинного обучения на языке Python / Л. П. Коэльо, В. Ричарт. – М.: ДМК Пресс, 2016. – 302 с.

14 OpenCV Tutorials – OpenCV 2.4.13.6 documentation – (Eng.). – URL: https://docs.opencv.org/2.4/doc/tutorials/tutorials.html [14 марта 2018].

15 Howse J. OpenCV: Computer Vision Projects with Python / J. Howse, P. Joshi, M. Beyeler. – Birmingham, 2016. – 541 p.