

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

РАБОТА ПРОВЕРЕНА

Рецензент,
Технический директор
ООО "Компания СГ-групп"
_____ Д.В. Гапонюк
“ ___ ” _____ 2018 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой,
д.ф.-м.н., профессор
_____ Л.Б. Соколинский
“ ___ ” _____ 2018 г.

**РАЗРАБОТКА СИСТЕМЫ РАСПОЗНАВАНИЯ
ПСИХОЛОГИЧЕСКОГО ПОДТИПА ЧЕЛОВЕКА
ПО КЛАССИФИКАЦИИ К.Г. ЮНГА НА ОСНОВЕ
АНАЛИЗА МИМИКИ С ПОМОЩЬЮ НЕЙРОСЕТЕВЫХ
ТЕХНОЛОГИЙ**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 02.03.02.2018.115-018.ВКР

Научный руководитель
к.ф.-м.н, доцент кафедры СП
_____ В. Голодов

Автор работы,
студент группы КЭ-401
_____ В.А. Петров

Ученый секретарь
(нормоконтролер)
_____ О.Н. Иванова
“ ___ ” _____ 2018 г.

Челябинск-2018

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	8
1.1. Типология К.Г. Юнга.....	8
1.2. Автоматизированный анализ мимики.....	9
1.3. Программные продукты для работы с мимикой.....	10
Выводы по главе 1.....	12
2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ	14
2.1. Предварительная обработка видеофайла.....	14
2.2. Обнаружение на изображении лица человека	15
2.3. Выравнивание лица.....	16
2.4. Многослойная нейронная сеть.....	17
2.5. Сверточные нейронные сети.....	19
2.6. Рекуррентные нейронные сети	20
Вывод по главе 2	22
3. ПРОЕКТИРОВАНИЕ	23
3.1. Топология нейронной сети.....	23
3.2. Требования к системе	23
3.3. Варианты использования системы	25
3.4. Архитектура системы	26
Вывод по главе 3	27
4. РЕАЛИЗАЦИЯ	28
4.1. Программные средства реализации	28
4.2. Реализация предобработки видеофайла	29
4.3. Реализация и обучение нейронной сети	29
4.4. Реализация приложения	31
Вывод по главе 4	32
5. ТЕСТИРОВАНИЕ	33
Вывод по главе 5	34
ЗАКЛЮЧЕНИЕ	35

ЛИТЕРАТУРА.....	36
ПРИЛОЖЕНИЯ.....	39
Приложение 1	39
Приложение 2	40

ВВЕДЕНИЕ

Каждый человек с рождения обладает определенным типом психики, который остается неизменным до конца его жизни. Имея информацию о психологическом типе можно оценить потенциал человека и найти пути его самореализации, профессионального выбора и роста. Также с высокой долей вероятности объективно воспринимать возможности и способности людей, что позволяет строить гармоничные отношения между ними. В результате диагностики психологического типа можно проанализировать индивидуальные особенности человека и определить стратегию взаимодействия с другими людьми и коллективом в целом [13]. Практическая область применения типологии К.Г. Юнга широка и лежит в областях: проблем личности, личностной совместимости, профориентации, коммуникации и многих других [9].

Актуальность работы

Для определения психологического типа человека требуются обученные классифицированные специалисты в области соционики, что требует затрат ресурсов на их профессиональную подготовку. Точность анализа зависит от множества субъективных качеств специалиста и требует затрат времени и усилий специалиста. Поэтому задача автоматизации и объективизации определения психологического типа человека является актуальной. Разработка программной системы поможет преодолеть вышеописанные трудности, что позволит повысить эффективность процедуры определения психологического типа человека по классификации К.Г. Юнга.

Цель и задачи исследования

Основной целью данной работы является разработать настольное приложение, которое позволяет по видеофрагменту с записью мимики определить психологический подтип человека (этик, логик) в классификации К.Г. Юнга.

Для достижения поставленной цели необходимо решить следующие

задачи.

1. Произвести обзор приложений и литературы по предметной области.
2. Разработать и реализовать алгоритм предобработки видеофрагмента.
3. Подготовить обучающую и тестовую выборки.
4. Разработать архитектуру нейронной сети.
5. Обучить и протестировать нейронную сеть.
6. Разработать и протестировать настольное приложение для распознавания психологического подтипа человека (этик, логик) в классификации К.Г. Юнга.

Структура и объем работы

Работа состоит из введения, пяти глав, заключения, списка литературы и приложения. Объем работы составляет 38 страниц, объем библиографии – 32 источника, объем приложения – 2 страницы.

В первой главе приводятся теоретические сведения предметной области, а также осуществляется обзор существующих подходов к программному анализу мимики человека.

Во второй главе представлены этапы предобработки видеофайла, а также приводятся теоретические сведения о искусственных нейронных сетях и их разновидностях топологий, применяющихся для решения поставленной задачи.

В третьей главе описывается топология нейронной сети, а также архитектура и требования к настольному приложению.

В четвертой главе описаны средства разработки, представлена программная реализация алгоритма предобработки видеофайла, приложения и нейронной сети. Также описана обучающая и тестовая выборка. Представлен процесс обучения нейронной сети.

В пятой главе описывается тестирование настольного приложения и нейронной сети.

В заключении проводится оценка успешности решения поставленных задач и приводятся направления для дальнейшей работы.

Приложение 1 содержит листинг основных функций предварительной обработки видеофайла.

Приложение 2 содержит листинг основного метода классификации психологического подтипа в настольном приложении.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Типология К.Г. Юнга

Типология Юнга – система типологии личности, основанная на понятиях психологической установки, которая может быть экстравертной либо интровертной и на преобладании той или иной психической функции – мышления, чувства, ощущения или интуиции [14].

Типология была создана швейцарским психиатром К.Г. Юнгом и опубликована в его работе «Психологические типы» в 1921 году [25].

Целью разработки типологии была не простая классификация людей на категории, Юнг преследовал следующие цели:

- 1) создать инструмент, который поможет упорядочить большое количество разнообразного опыта в некоем подобии координатной шкалы;
- 2) помочь на практике с помощью классификации пациента и самого психолога выбрать наиболее эффективные методы лечения.

Типология Юнга базируется на двух установках экстраверсия-интроверсия и на четырех функциях: мышление (логика), чувство (этика), интуиция, ощущение (сенсорика). По Юнгу каждый человек обладает всеми четырьмя психологическими функциями, однако эти функции не развиты в одинаковой степени. Обычно одна функция доминирует, давая человеку возможность для достижения каких-либо социальных целей. Другие функции в сравнении с доминирующей неизбежно отстают.

Все функции делятся на два класса: «рациональные» лежащие в сфере разума (мышление и чувство) и «иррациональные», которые, соответственно лежат за пределами разума (ощущение и интуиция). Доминирование какой-либо функции из класса требует подавление противоположной (чувство исключает мышление, интуиция – ощущение и наоборот).

Для более точного описания человеческой психики Юнг также вводит понятие «вспомогательной» или дополнительной функции, что приводит к тому, что к доминирующей функции добавляется вспомогательная функция из противоположного класса. В соответствии со всеми приведенным

критерии образуются 16 психологических типов (таблица 1), по которым Юнг и предлагает классифицировать людей.

Табл. 1. Распределение психологических типов

	Рациональный класс	Иррациональный класс
Экстраверсия	Гюго (этик-сенсорик)	Дон-Кихот (интуит-логик)
	Гамлет (этик-интуит)	Жуков (сенсорик-логик)
	Джек (логик-интуит)	Наполеон (сенсорик-этик)
	Штирлиц (логик-сенсорик)	Гексли (интуит-этик)
Интроверсия	Робеспьер (логик-интуит)	Дюма (сенсорик-этик)
	Максим (логик-сенсорик)	Есенин (интуит-этик)
	Драйзер (этик-сенсорик)	Бальзак (интуит-логик)
	Достоевский (этик-интуит)	Габен (сенсорик-логик)

В данной работе рассматриваются только психические функции рационального класса и выделение доминирующей функции происходит независимо от того, является ли она главной или вспомогательной в классификации по К.Г. Юнгу. Таким образом 16 психологических типов делятся на два подтипа: «логики» с доминирующей функцией мышления и «этики» с доминирующей функцией чувства.

Классификация подтипа в дихотомии логик-этик производится на основе анализа мимики человека, так как мимика является одним из основных критериев, который на практике учитывается специалистами для определения психотипа человека и в частности для определения доминирующей функции рационального класса.

1.2. Автоматизированный анализ мимики

В ходе изучения литературы было выявлено, что большинство автоматизированных систем осуществляет анализ выражения человеческого лица по основным эмоциям [12]. Одной из фундаментальных работ в этой области, является работа [32], в которой авторы представляют наиболее широко используемую систему для классификации выражений лица чело-

века с помощью кодирования лицевых движений.

Работы [3, 4, 11] показывают, что система кодирования лицевых движений (Facial Action Coding System, (FACS)) успешно применяется для анализа мимики и распознавания эмоций по изображению фронтального лица человека.

В статье [29] авторы приводят способ, позволяющий использовать FACS для анализа выражений лица в видеопотоке.

В статье [23] приводится метод адаптации FACS для анализа мимики с помощью нейронной сети.

Работа [27] содержит исследование по анализу эмоций человека в видеофайле с применением различных методов машинного обучения.

Авторы статьи [12] проводят примеры работ, в которых анализ мимики успешно применяется к таким прикладным задачам как предсказания засыпания водителя по морганию и движению его глаз, определения эмоциональных реакций человека по мимике, управление курсором мыши с помощью движения головы и мимики пользователя.

1.3. Программные продукты для работы с мимикой

В процессе анализа предметной области не выявлено прямых аналогов, которые позволяют по мимике определить психологический тип человека. Но приведенные ниже программные продукты, так или иначе, в своей работе используют методы для работы с мимикой.

FaceReader

FaceReader (рисунок 1) – программное обеспечение компании Noldus Information Technology (Нидерланды), способное классифицировать по изображению человеческого лица такие эмоции, как счастье, грусть, злоба, удивление, страх, отвращение и спокойствие. Кроме того, программа способна определить пол, возраст и национальную принадлежность. Приложение умеет работать с видеофайлами. Распознавание эмоций происходит следующим образом: на изображении находится лицо человека с помощью

метода Виолы-Джонса, далее создается искусственная модель лица с помощью метода Active Appearance и происходит классификация эмоции обученной искусственной нейронной сетью по созданной модели лица [17].

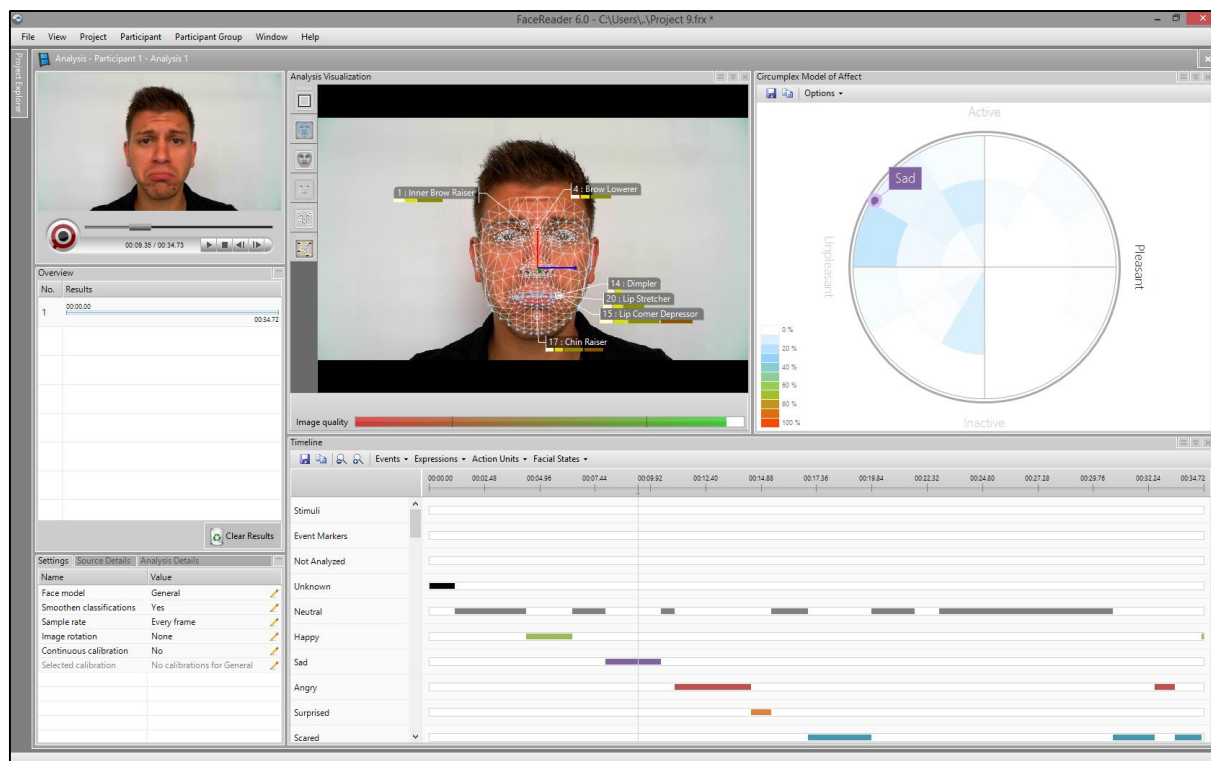


Рис. 1. FaceReader

EmoDetect

EmoDetect (рисунок 2) – программное обеспечение компании Нейроботикс (Россия), которое позволяет определить психоэмоциональное состояние человека по выборке изображений (видео или набор файлов). Классификатор позволяет определить шесть базовых эмоций: радость, удивление, грусть, злость, страх, отвращение. Имеет высокую точность как на отдельном взятом кадре, так и в динамике. Классификация эмоций осуществляется расчетом двигательных единиц по системе кодирования лицевых движений П. Экмана (Facial Action Coding System (FACS)) и их дальнейшим анализом тремя независимыми классификаторами: искусственной нейронной сетью, системой решающих правил, классификатором по взвешенной сумме признаков [16].

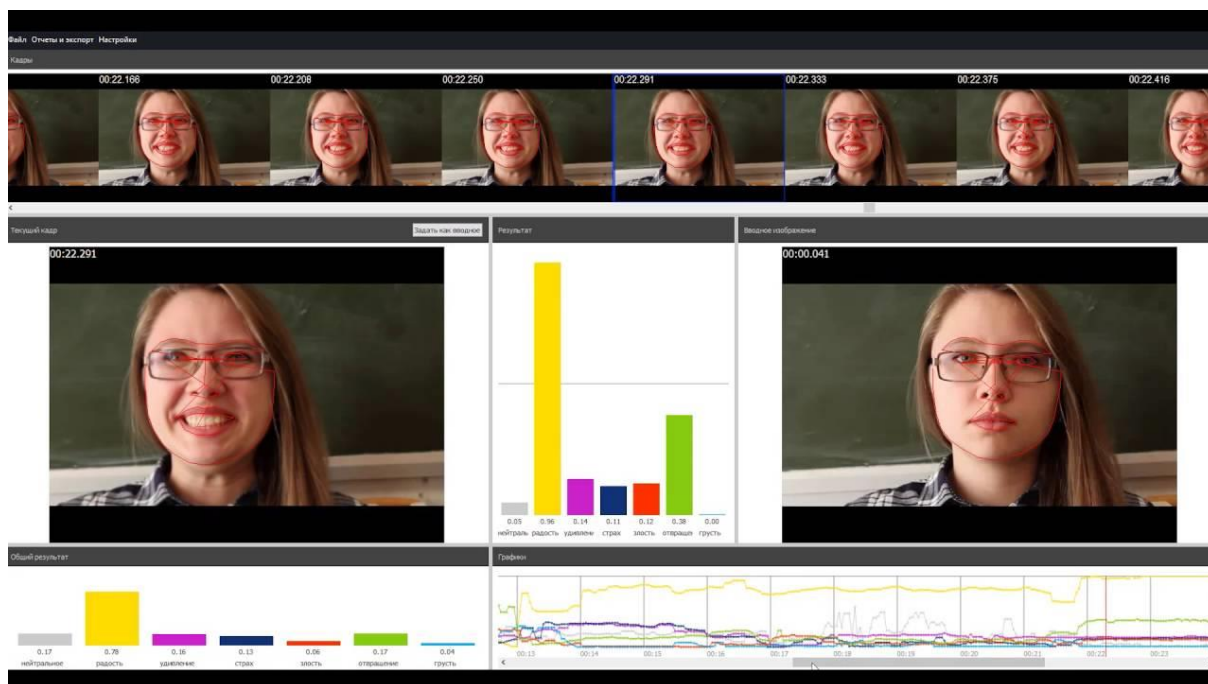


Рис. 2. EmoDetect

Facial Emotion Recognition

Facial Emotion Recognition (рисунок 3) – свободное программное обеспечение, позволяющие классифицировать шесть базовых эмоций: злость, страх, счастье, грусть, спокойствие, удивление.

Для классификации используется заранее обученная искусственная нейронная сеть, на вход которой подается изображение размером 48 на 48 пикселей с лицом человека анфас в градациях серого [28].

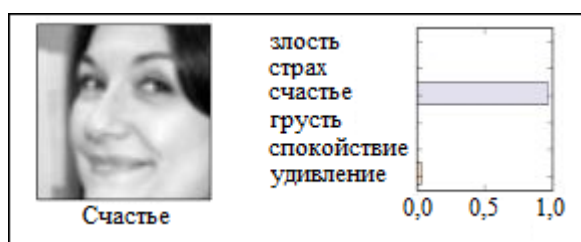


Рис. 3. Facial Emotion Recognition

Выводы по главе 1

В ходе анализа предметной области: литературы и программных систем, были выявлены методы и подходы к анализу мимики, которые могут быть использованы для определения психологического подтипа (этик-

логик), но прямых программных аналогов для решения поставленной задачи не выявлено.

Одним из широко используемым и показывающим стабильный результат оказался метод FACS в связке с машинным обучением, но на практике используются, как и более сложные методы и подходы (Active Appearance и нейронные сети), так и более простые (машинное обучение).

В работе большинства рассмотренных классификаторов эмоций прослеживается общий алгоритм работы, который заключается в том, что исходный видеофайл разбивается на отдельные кадры. На каждом кадре распознается лицо человека, на котором, следующим шагом, происходит выделение признаков и классификация с помощью методов машинного обучения, чаще всего заранее обученными искусственными нейронными сетями. Вследствие этого, в данной работе будет применен нейросетевой подход, как наиболее перспективный. При этом нейронная сеть будет разработана для решения задачи более высокого уровня абстракции – классификации психологического подтипа человека.

2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

2.1. Предварительная обработка видеофайла

Использование алгоритма классификации основанного на нейросетевом подходе требует предварительной обработки видеофайла, с целью сделать данные пригодными для передачи в нейронную сеть и улучшить качество обучения нейронной сети, выделив только полезную информацию из видеофайла.

В данной работе предобработка заключается в том, чтобы преобразовать входной видеофайл в последовательность изображений фиксированного размера в градациях серого, на которых представлена мимика человека. Последовательность должна задаваться хронологическим порядком видеофрагмента, из которого она получена.

Этапы предобработки видеофрагмента

Согласно анализу предметной области, было выявлено, что для достижения приведенного выше результата от предобработки видеофайла, необходимо рассмотреть следующие этапы.

1. Декомпозиция видеофайла на последовательность кадров.
2. Приведение изображения в градацию серого.
3. Обнаружение и выделение на изображении области с лицом человека.
4. Преобразование лица на изображении таким образом, чтобы ключевые элементы (глаза, нос, губы) всегда находились в одном и том же месте изображения.

Для декомпозиции видеофрагмента на кадры и их дальнейшего преобразования в градации серого существует большое множество открытых программных библиотек, различающихся только множеством форматов данных, с которыми может работать та или иная библиотека. Другие этапы предобработки, такие как поиск и выравнивание лица, требует применение более сложных алгоритмических подходов. Подробнее они рассмотрены ниже.

2.2. Обнаружение на изображении лица человека

Для обнаружения на изображении лица человека был выбран алгоритм, основанный на гистограмме направленных градиентов (Histogram of Oriented Gradients, HOG), который является широко распространенным и обладает высокой точностью [31]. Основной идеей алгоритма является допущение, что внешний вид и форма лица на участке изображения могут быть описаны распределением градиентов интенсивности.

Данный алгоритм работает следующим образом. Исходное изображение преобразуется в градации серого. Далее для каждого пикселя изображения определяется градиент в контексте соседних пикселей [26].

На следующем шаге изображение разбивается на небольшие квадраты 16x16 пикселей называемые ячейками. Для каждой ячейки вычисляется гистограмма направлений градиентов, с помощью участия каждого пикселя ячейки во взвешенном голосовании для девяти каналов гистограммы направлений [26]. Графическое представление результата работы этой части алгоритма приведено на рисунке 4.

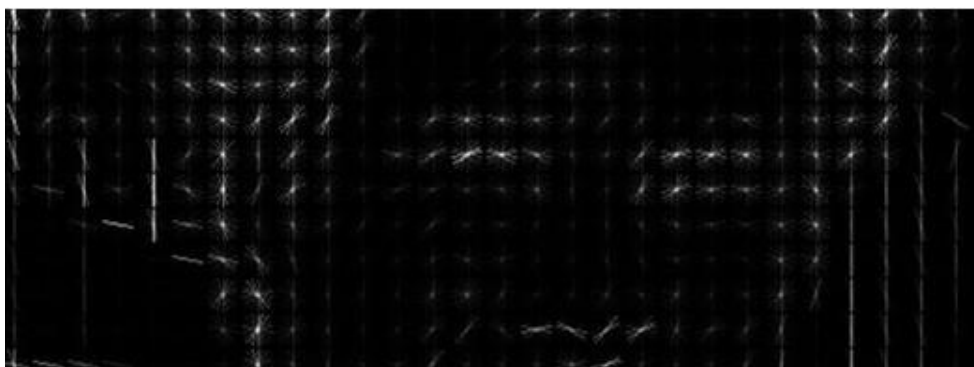


Рис. 4. Графическое представление гистограммы для каждой ячейки

Затем ячейки группируются в еще более крупные квадратные блоки – дескрипторы, где происходит нормировка гистограмм ячеек. Каждая ячейка может входить в более чем один дескриптор. Конечным шагом является подача дескриптора на предобученный SVM-классификатор [26]. Графическое представление результата работы алгоритма показано на рисунке 5.

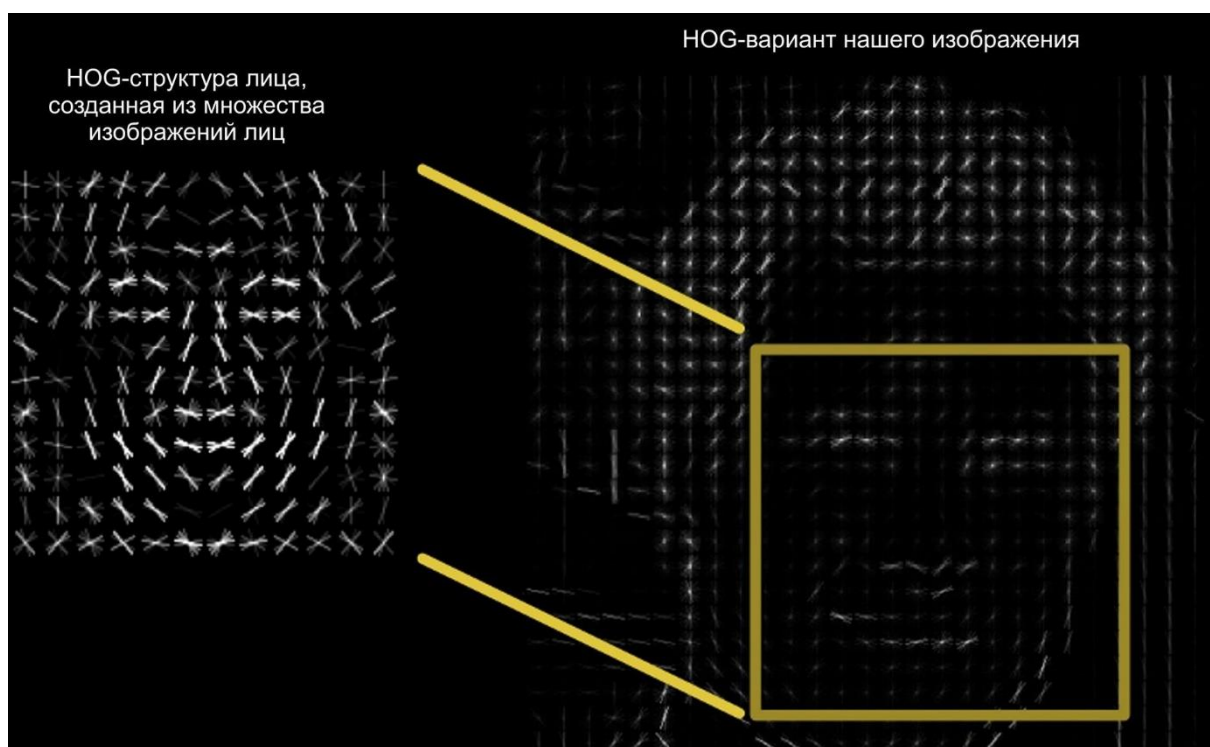


Рис. 5. Классификация дескриптора

2.3. Выравнивание лица

После обнаружения на исходном изображении лица человека, производится преобразование изображения таким образом, чтобы ключевые элементы лица, всегда находились в одном и том же месте изображения. Это преобразование необходимо, чтобы добиться расположения лица человека анфас на каждом кадре, что помогает избежать влияние поворота головы при дальнейшем анализе мимики.

Для данного этапа используется алгоритм, основанный на оценке антропометрических точек. Идея алгоритма в том, что на лице выделяется 68-ть антропометрических точек, характеризующих различные части лица, такие как выступающая часть подбородка, внешние края глаз, внутренние края бровей и т.д. Выделение точек происходит с помощью метода описанного в работе [30]. Затем производятся аффинные преобразования изображения таким образом, чтобы точки, соответствующие глазам и нижней губе, находились в заданном месте изображения. Пример выравнивания лица показан на рисунке 6.

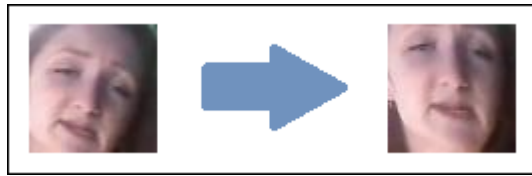


Рис. 6. Пример выравнивания лица

Аффинные преобразование позволяют сохранить параллельность и соотношение линий на изображение.

2.4. Многослойная нейронная сеть

Искусственная нейронная сеть – математическая модель, построенная по принципу организации и функционирования биологических нейронных сетей [24]. Нейронная сеть представляет собой один из методов машинного обучения и соответственно является частным методом решения задачи классификации. Базовой единицей обработки информации в нейронной сети является нейрон. На рисунке 7 показана модель нейрона, лежащего в основе искусственных нейронных сетей.

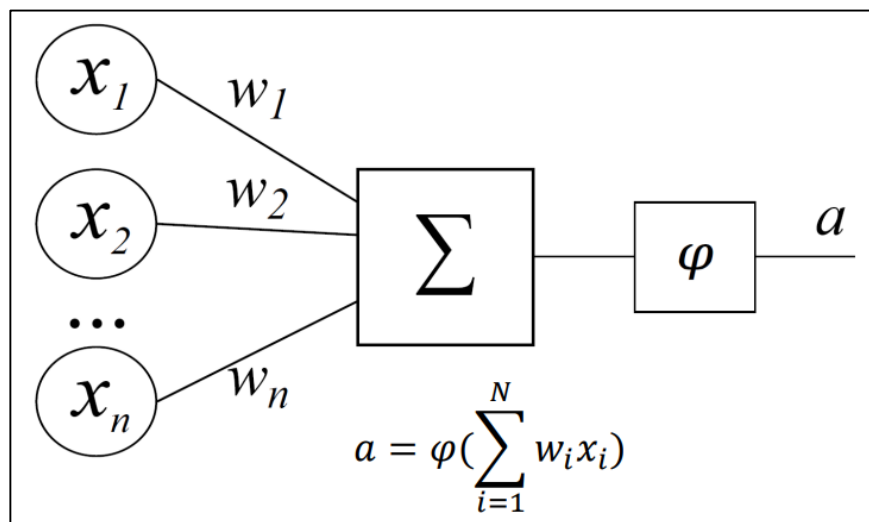


Рис. 7. Модель нейрона

Искусственный нейрон, с математической точки зрения, представляется сумматором всех входящих сигналов, применяющий к полученной взвешенной сумме некоторую простую, нелинейную функцию, непрерыв-

ную на всей области определения и называемую функцией активации [10]. Нейроны, объединенные по особенностям функционирования, имеющие один и тот же набор входов и не соединенные между собой образуют слой нейронной сети [22]. Нейронную сеть, состоящую из нескольких слоев, называют многослойной нейронной сетью прямого распространения [24].

Для корректной работы классификатора, основанного на многослойной нейронной сети, необходимо произвести обучение искусственной нейронной сети. Процесс обучения заключается в настройке значений весов в связях между нейронами с помощью алгоритма обратного распространения ошибок.

Обучение алгоритмом обратного распространения ошибки предполагает два прохода по всем слоям сети: прямого и обратного. При прямом проходе входной вектор подается на входной слой нейронной сети, после чего распространяется по сети от слоя к слою. В результате генерируется набор выходных сигналов, который и является фактической реакцией сети на входной образ. Во время прямого прохода все веса сети фиксированы. Во время обратного прохода все веса настраиваются в соответствии с правилом коррекции ошибок, а именно: фактический выход сети вычитается из желаемого, в результате чего формируется сигнал ошибки. Этот сигнал впоследствии распространяется по сети в направлении, обратном направлению связей. Веса настраиваются с целью максимального приближения выходного сигнала сети к желаемому [2].

Возможность обучения является одним из главных преимуществ нейронных сетей перед традиционными алгоритмами. В процессе обучения нейронная сеть способна выявлять сложные зависимости между входными данными и выходными, а также выполнять обобщение. Следовательно, в случае успешного обучения, нейронная сеть сможет вернуть верный результат на данных, которые отсутствовали в обучающей выборке. В данной работе топология нейронной сети содержит слои, характерные для сверточной и рекуррентной нейронной сети.

2.5. Сверточные нейронные сети

Сверточная нейронная сеть представляет собой многослойную искусственную нейронную сеть, специально созданную для распознавания двухмерных поверхностей с высокой степенью инвариантности к преобразованиям, масштабированию, искажениям и прочим видам деформации [24]. Основной идеей сверточной нейронной сети является чередование сверточных и субдискретизирующих слоев.

Сверточный слой включает в себя ядра свертки, представляющие собой матрицы весов небольшого размера, которые обрабатывают предыдущий слой по фрагментам, суммируя результаты матричного произведения для каждого фрагмента. Пример применения ядра показан рисунке 8.

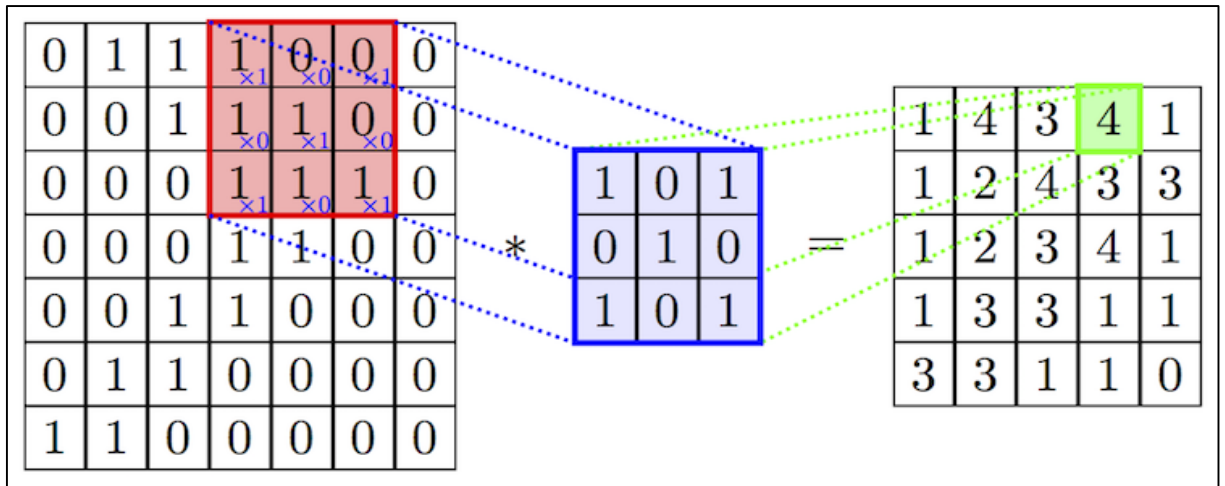


Рис. 8. Применение ядра свертки

Ядра свертки интерпретируются как графическое кодирование какого-либо признака на изображении. Тогда следующий слой, получившийся в результате операции свертки, показывает наличие заданного признака в обрабатываемом слое и его координаты, формируя тем самым карту признаков. Проход каждым ядром свертки формирует свой собственный экземпляр карты признаков, делая нейронную сеть многоканальной. Весовые коэффициенты ядра свертки заранее неизвестны и устанавливаются в процессе обучения нейронной сети. Особенностью слоя является сравнительно небольшое количество параметров, устанавливаемое при обучении [21].

Слой субдискретизации выполняет задачу уменьшение размерности сформированных карт признаков. Предполагается, что информация о факте наличия искомого признака важнее точного знания его координат, поэтому из нескольких соседних нейронов карты признаков выбирается максимальный и принимается за один нейрон уплотненной карты признаков меньшей размерности. Пример применения операции субдискретизации показан на рисунке 9. За счет данной операции, помимо ускорения дальнейших вычислений, сеть становится более инвариантной к масштабу входного изображения [21].

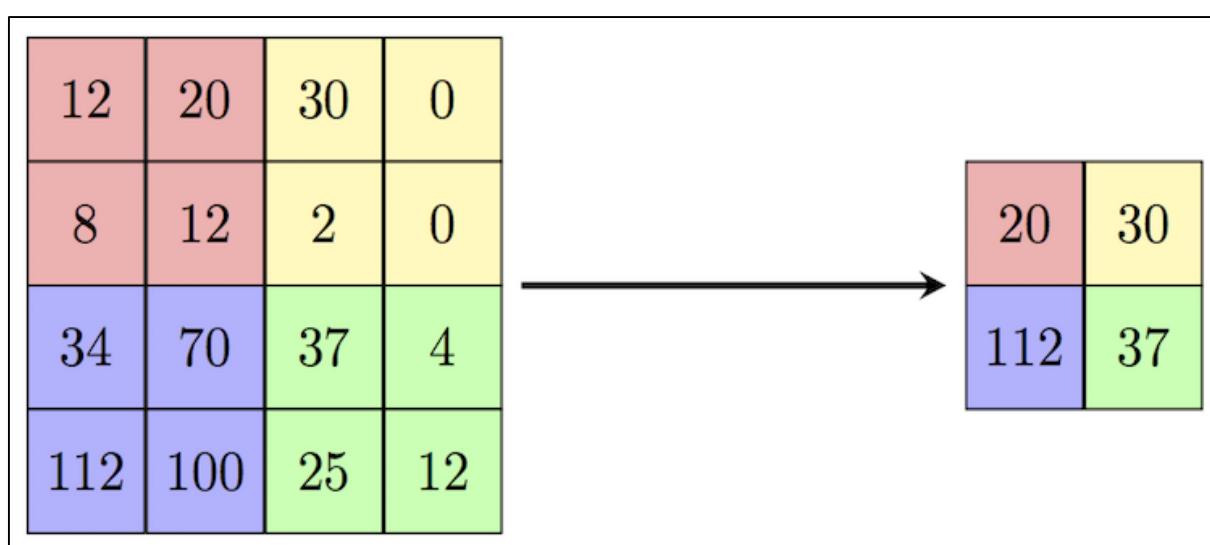


Рис. 9. Применение операции субдискретизации

Чередование сверточных и субдискретизационных слоев позволяет составлять карты признаков из карт признаков. На каждом следующем слое карта уменьшается в размере, но увеличивается количество каналов, что дает возможность распознавания сложных иерархий признаков.

2.6. Рекуррентные нейронные сети

Рекуррентные нейронные сети – вид нейронных сетей, где присутствуют обратные связи между элементами нейронной сети [24]. Благодаря этому нейронная сеть способна обрабатывать серии событий во времени или последовательные пространственные цепочки. В отличие от нейрон-

ных сетей прямого распространения, модули в рекуррентной нейронной сети способны использовать свою «внутреннюю» память для обработки последовательности произвольной длины. Одним из разновидностей рекуррентной нейронной сети, являются сети с долговременной и краткосрочной памятью (LSTM). LSTM-сеть приспособлена к задачам классификации, обработки и прогнозирования временных рядов и способна к обучению долговременным и кратковременным зависимостям.

Основным компонентом LSTM-сети является состояние ячейки. Состояние ячейки представляет собой конвейерную ленту, которая проходит через всю последовательность модулей рекуррентной нейронной сети, участвуя лишь в нескольких линейных преобразованиях. Таким образом, информация может спокойно проходить по ней, не подвергаясь изменениям. Тем не менее, информация может удаляться из состояния ячейки, этот процесс регулируется специальными структурами – фильтрами. Графическое представление структуры LSTM-сети показано на рисунке 10.

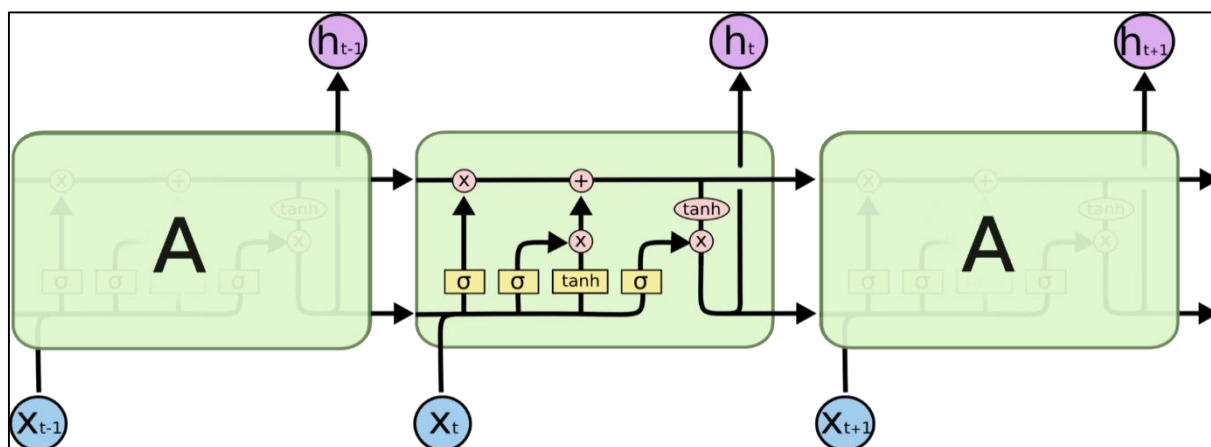


Рис. 10. Структура LSTM-сети

Фильтры способны пропускать информацию на основании некоторых условий. Они состоят из слоя сигмоидальной нейронной сети и операции поточечного умножения. Сигмоидальный слой возвращает числа от нуля до единицы, которые обозначают, какую долю каждого блока информации следует пропустить дальше по сети. В LSTM-сети три таких фильтра позволяют защищать и контролировать состояние ячейки.

Вывод по главе 2

Были рассмотрены основные этапы предобработки видеофайла, а также приведены теоретические сведения о искусственных нейронных сетях и их разновидностях топологий, применяющихся для решения поставленной задачи.

3. ПРОЕКТИРОВАНИЕ

3.1. Топология нейронной сети

Для решения поставленной задачи, была разработана топология нейронной сети, которая представлена на рисунке 11.

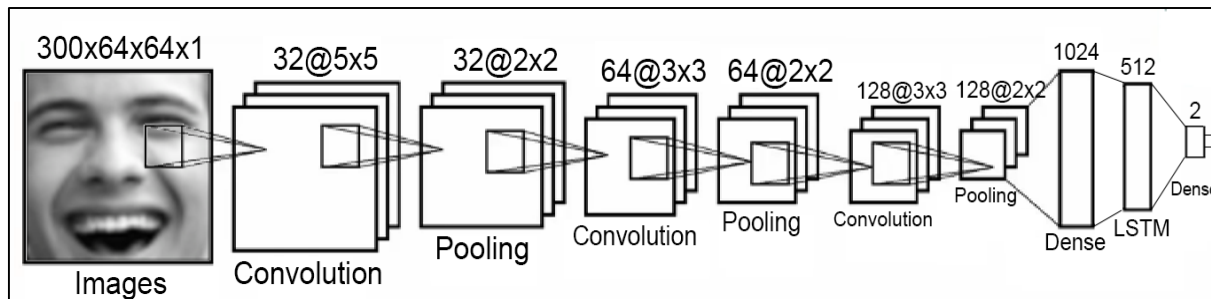


Рис. 11. Топология нейронной сети

Представленная топология нейронной сети содержит сверточные, рекуррентные и полносвязные слои и является результатом многочисленных экспериментов, с подбором количества слоёв и их параметров.

В начале расположены сверточные слои. Их задача заключается в том, чтобы получить признаковое описание изображения в виде набора карт признаков. Полученный набор карт признаков, подается на вход в полносвязный, а затем и в рекуррентный слой нейронной сети. Задача данных слоев сделать прогноз психологический подтип человека, с учетом изменений признаков во времени.

Последним слоем нейронной сети является полносвязный слой. Он отвечает за представления результата, полученного с рекуррентного слоя, в виде числового вектора, где каждое измерение содержит вероятность того, что психологический подтип человека принадлежит заданному классу.

3.2. Требования к системе

В ходе проектирования приложения были сформулированы функциональные и нефункциональные требования к разрабатываемой системе.

Функциональные требования

Функциональные требования – это требования, охватывающие пред-

полагаемое поведение системы, указывающие действия, которые система способна выполнять. В рамках разрабатываемой программной системы были выявлены следующие функциональные требования.

1. Система должна быть реализована в виде настольного приложения.
2. Система должна поддерживать выбор пользователем видеофайла в формате mp4.
3. Система должна предоставлять возможность запуска процесса классификации психологического подтипа человека на выбранном видеофайле.
4. Система должна предоставлять возможность прерывания процесса классификации психологического подтипа.
5. Система должна выводить информацию о текущем прогрессе в процессе классификации психологического подтипа.
6. Система должна поддерживать выбор пользователем модели нейронной сети, используемой для классификации психологического подтипа человека.

Нефункциональные требования

Нефункциональные требования – это требования, описывающие характер поведения системы, атрибуты качества, дизайна, производительности и прочие ограничения программного обеспечения, а также различных условий, которым она может удовлетворять.

Были выявлены следующие нефункциональные требования к разрабатываемой программной системе.

1. Система должна быть реализована на языке программирования Python 3.
2. Система должна работать на операционных системах Windows и Linux.
3. Для процесса классификации психологического подтипа человека должна использоваться предобученная нейронная сеть.

3.3. Варианты использования системы

Для проектирования приложения был использован язык графического описания объектного моделирования UML. Была построена модель взаимодействия актера «Пользователь» с приложением в виде диаграммы вариантов использования, которая представлена на рисунке 12.

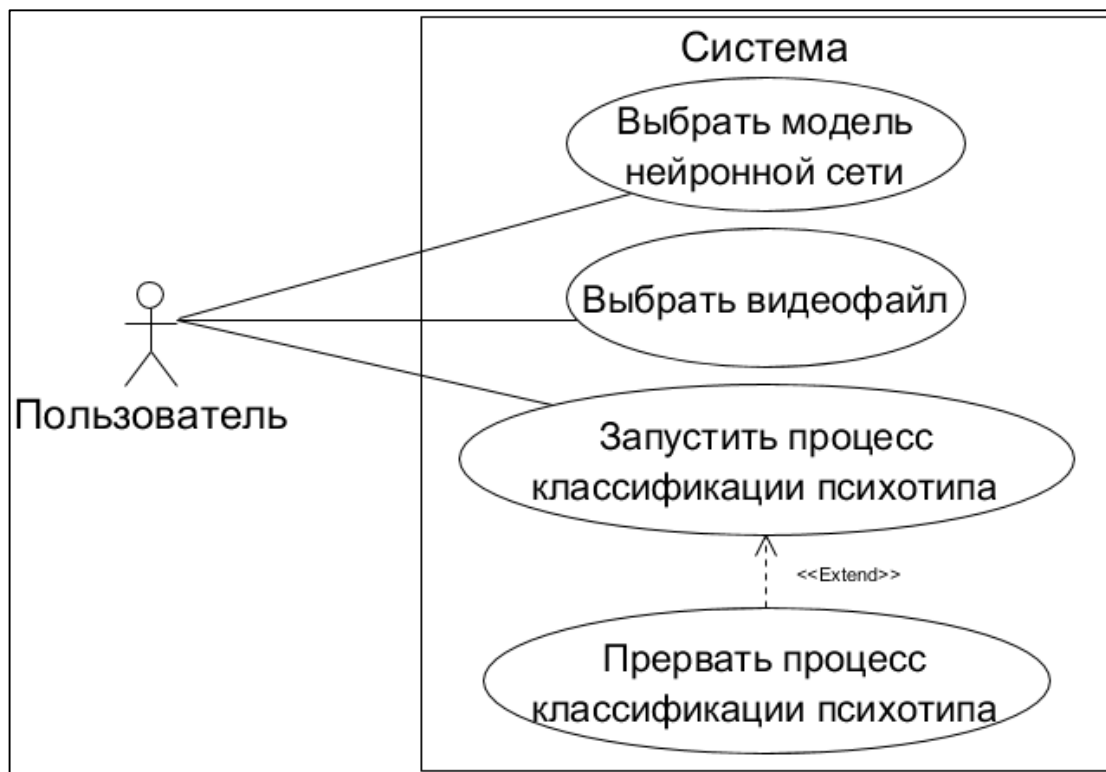


Рис. 12. Диаграмма вариантов использования

В рамках разрабатываемой системы предусмотрен один актер – пользователь. Пользователь – лицо, которое использует действующую систему для выполнения конкретной функции.

Данный актер может реализовать следующие варианты использования системы.

1. Пользователь может выбрать модель нейронной сети.
2. Пользователь может выбрать видеофайл, который доступен ему на компьютере.
3. Пользователь может запустить процесс классификации психологического подтипа человека. Если на момент запуска процесса распознава-

ния не выбран видеофайл, то система извещает пользователя об ошибке и игнорирует команду. Иначе система классифицирует психологический подтип человека на видеофайле и выводит результат.

4. Пользователь может прервать процесс классификации психологического подтипа человека, если он был до этого запущен.

3.4. Архитектура системы

В ходе проектирования системы была разработана система классов, которая представлена в диаграмме классов на рисунке 13.

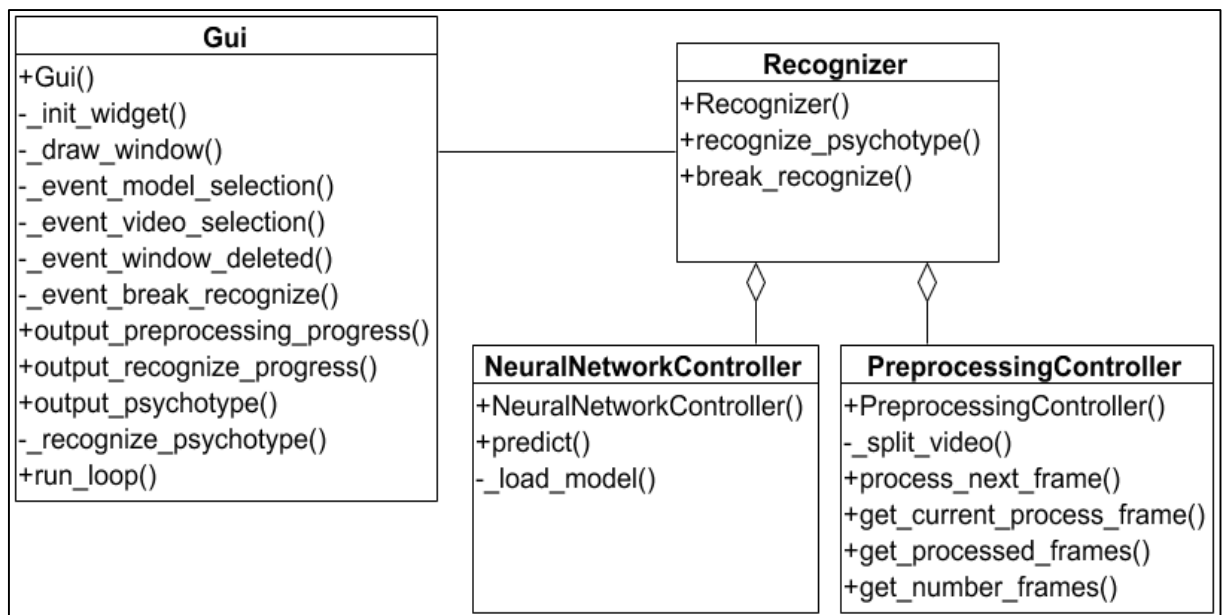


Рис. 13. Диаграмма классов системы

Ниже представлено более подробное описание классов, составляющих программную систему.

Gui – отображает интерфейс для взаимодействия пользователя с системой. Содержит в себе кнопки для выбора модели нейронной сети, загрузки видеофайла, запуска и прерывания процесса классификации психологического подтипа человека, а также диалоговое окно, для вывода сообщения. Является точкой входа в программу.

Recognizer – класс, который содержит основную логику работы системы. Обеспечивает связь между интерфейсом и логикой программы.

NeuralNetworkController – класс, предоставляющий набор методов для работы с искусственной нейронной сетью.

PreprocessingController – класс, предоставляющий набор методов для предобработки видеофайла.

Вывод по главе 3

В главе была приведена топология искусственной нейронной сети, полученная в ходе многочисленных экспериментов. А также представлены функциональные и нефункциональные требования к системе и спроектирована UML диаграмма вариантов использования и диаграмма классов к разрабатываемому настольному приложению.

4. РЕАЛИЗАЦИЯ

4.1. Программные средства реализации

Для реализации системы распознавания психологического подтипа человека по классификации К.Г. Юнга был выбран язык программирования Python 3. Процесс написания кода и его отладки производился в среде разработки PyCharm Community Edition 2017.1. Также использовалась система контроля версий Git, а в качестве удаленного репозитория был выбран веб-сервис Bitbucket.

В процессе разработки были использованы следующие программные библиотеки.

1. NumPy [18] – библиотека для языка программирования Python, добавляющее поддержку больших многомерных массивов и матриц, вместе с большой библиотекой высокоуровневых математических функций для операций с этими массивами.

2. Dlib [15] – кроссплатформенная программная библиотека, содержащая алгоритмы машинного обучения и инструменты для создания сложного программного обеспечения. Используется как в промышленности, так и в академических кругах в широком диапазоне областей, включая робототехнику, встроенные устройства, мобильные телефоны и большие высокопроизводительные вычислительные среды.

3. PIL [6] – библиотека для языка программирования Python, предназначенная для работы с изображениями.

4. OpenFace [19] – открытая библиотека для распознавания лиц, использующая глубокую сверточную нейронную сеть, которая основана на технологии FaceNet.

5. PyAV [7] – библиотека для языка программирования Python, предназначенная для работы с видео.

6. TensorFlow [20] – открытая программная библиотека для машинного обучения, разработанная компанией Google для решения задач построения и тренировки нейронной сети с целью автоматического находже-

ния и классификации образов, достигая качества человеческого восприятия.

7. Keras [5] – открытая библиотека, представляющая собой надстройку над фреймворками DeepLearning4j, TensorFlow и Theano.

8. Tkinter [8] – кросс-платформенная графическая библиотека на основе средств Tk, написанная Стивом Лумхольтом и Гвидо ван Россумом.

4.2. Реализация предобработки видеофайла

После анализа предметной области был спроектирован и реализован следующий алгоритм предварительной обработки видеофайла.

1. Разбить видеофайла на последовательность кадров.
2. Для каждого кадра выполнить следующее:
 - a) преобразовать в градацию серого;
 - b) выделить прямоугольную область с лицом человека;
 - c) вырезать найденную область;
 - d) выровнять лицо внутри области;
 - e) преобразовать изображение к заданным размерам.

Реализация данного алгоритма представляет собой функцию, которая принимает на вход путь до видеофайла и размер итоговых изображений. Ее программный код приведен в приложении 1.

С целью уменьшения времени работы предобработки, функции была реализована в многопоточном стиле. Была добавлена возможность сериализации полученного результата работы функции, что также позволило, при повторном обращении к уже ранее обработанному видеофайлу, сократить время работы функции.

4.3. Реализация и обучение нейронной сети

Для обучения и тестирования нейронной сети была подготовлена выборка, состоящая из 262 видеофайлов с записью мимики людей. Источником послужил сайт [1]. Из данного источника были отобраны видеофай-

лы, на которых запечатлена мимика ярких представителей классифицируемых психологических подтипов. Видеофайлы был приведён к одному формату: длительность – 10 секунд, частота – 30 кадров в секунду, размер – 1280x720 пикселей, формат – mp4.

В листинге 1 представлен программный код топологии нейронной сети.

Листинг 1. Программный код топологии нейронной сети

```
model = Sequential()
model.add(TimeDistributed(Conv2D(32, (5, 5), padding='same', activation='relu'), input_shape=(None, 64, 64, 1)))
model.add(TimeDistributed(MaxPooling2D(pool_size=(2, 2))))
model.add(Dropout(0.2))
model.add(TimeDistributed(Conv2D(64, (3, 3), padding='same', activation='relu')))
model.add(TimeDistributed(MaxPooling2D(pool_size=(2, 2))))
model.add(Dropout(0.2))
model.add(TimeDistributed(Conv2D(128, (3, 3), padding='same', activation='relu')))
model.add(TimeDistributed(MaxPooling2D(pool_size=(2, 2))))
model.add(Dropout(0.2))
model.add(TimeDistributed(Flatten()))
model.add(TimeDistributed(Dense(1024, activation='relu')))
model.add(Dropout(0.25))
model.add(LSTM(512, dropout=0.1, return_sequences=False))
model.add(Dense(number_class, activation='softmax'))
```

Данная нейронная сеть состоит из трех слоев свертки, трех слоев субдискретизации, одного рекуррентного слоя и двух полносвязных слоев.

На вход нейронной сети подается 4 мерный массив, размером $N \times 64 \times 64 \times 1$, где N – количество кадров в видеофайле. На выходе получаем массив вероятностей размером равным количеству классов.

Программный код, для запуска процесса обучения нейронной сети, представлен в листинге 2.

Листинг 2. Программный код для запуска процесса обучения

```
train_data, train_answer, test_data, test_answer = \
```

```

preprocessing.get_dataset(options.PATH_TO_DATA, options.IMAGE_SIZE,
                          options.TO_TRAIN, options.NUMBER_CLASS)
model = model_4.get_model(options.IMAGE_SIZE, options.NUMBER_CLASS)
model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.Adadelta(),
              metrics=['categorical_accuracy'])
model.fit(train_data, train_answer, batch_size=options.BATCH_SIZE,
          epochs=options.NUMBER_EPOCH,
          validation_data=(test_data, test_answer), shuffle=True)
model_4.save_model(model)

```

Для обучения нейронной сети было использовано 80% видеофайлов от общей выборки, т.е. 209 видеофайлов. Обучение длилось 20 эпох.

Функция ошибки была выбрана categorical crossentropy:

$$H(p, q) = - \sum_x p(x) \log q(x)$$

где: $p(x)$ – «истинная» вероятность попадания объекта в класс x ;
 $q(x)$ – вероятность, полученная на выходе нейронной сети.

4.4. Реализация приложения

Реализация основного метода классификации психологического под-типа приведена в приложении 2. Пример графического интерфейса приведен на рисунке 14.

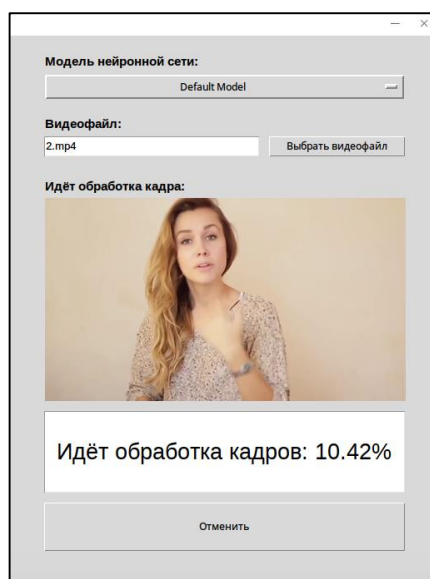


Рис. 14. Интерфейс приложения в процессе классификации психотипа

Вывод по главе 4

В данной главе описаны средства разработки, представлен алгоритм предварительной обработки видеофайла и его программная реализация. Также описана обучающая и тестовая выборка. Представлен процесс обучения нейронной сети и графический интерфейс приложения.

5. ТЕСТИРОВАНИЕ

Тестирование – это процесс, который заключается в проверке соответствия программного продукта заявленным характеристикам и требованиям, требованиям эксплуатации в различных окружениях, с различными нагрузками, требованиями по безопасности и удобству использования.

Тестирование нейронной сети

Для определения эффективности обучения нейронной сети использовалась тестовая выборка, которая составляет 20 % от общей выборки, что равняется 53 видеофайлам.

Для оценки эффективности, применялась формула categorical accuracy. Обученная нейронная сеть показала эффективность 96,23 %.

Функциональное тестирование приложения

Функциональное тестирование – это тестирование программного обеспечения, для проверки реализуемости функциональных требований, то есть способность программного обеспечения в определенных условиях решать задачи, нужные пользователям. Функциональные требования определяют, что именно делает программное обеспечение, какие задачи оно решает.

Было проведено тестирование на основе функциональных требований. Ниже представлен протокол функционального тестирования.

Тест №1. Цель: Проверить возможность выбора видеофайла.

Действие: Пользователь нажимает на кнопку «Выбрать видеофайл».

Ожидаемый результат: Пользователь увидит окно выборы видеофайла.

Тест пройден? Да.

Тест №2. Цель: Проверить возможность загрузки видеофайла.

Действие: Пользователь нажимает на кнопку «Выбрать видеофайл», выбирает видеофайл и нажимает кнопку «Открыть».

Ожидаемый результат: Видеофайл загрузился, и пользователь увидел уведомление.

Тест пройден? Да.

Тест №3. Цель: Проверить возможность классификации психологического подтипа человека.

Действие: Пользователь нажимает кнопку «Определить психотип» с выбранным видеофайлом.

Ожидаемый результат: Пользователь увидит сообщение о запуске процесса определения психотипа. В процессе пользователь будет получать информацию о текущем прогрессе. И в конце, пользователь получит результат.

Тест пройден? Да.

Тест №4. Цель: Проверить возможность прервать процесс классификации психологического подтипа человека.

Действие: Пользователь нажимает кнопку «Определить психотип» с выбранным видеофайлом, после того как начался процесс классификации, пользователь нажимает кнопку «Отмена».

Ожидаемый результат: Процесс классификации прервется, и пользователь увидит об этом сообщение.

Тест пройден? Да.

Тест №5. Цель: Проверить возможность выбора модели нейронной сети.

Действие: Пользователь нажимает на кнопку со списком моделей.

Ожидаемый результат: Пользователь увидит список доступных моделей и сможет выбрать любую.

Тест пройден? Да.

Вывод по главе 5

В главе приведены результаты оценки эффективности нейронной сети, а также представлен протокол функционального тестирования. Нейронная сеть показала высокую эффективность, а приложение успешно прошло функциональное тестирование.

ЗАКЛЮЧЕНИЕ

В рамках данной работы было разработано настольное приложение, которое позволяет по видеофрагменту с записью мимики определить психологический подтип человека (этик, логик) в классификации К.Г. Юнга.

При этом были решены следующие задачи.

1. Проведен обзор приложений и литературы предметной области.
2. Разработан и реализован алгоритм предобработки видеофрагмента.
3. Подготовлена обучающая и тестовая выборка.
4. Разработана архитектура нейронной сети.
5. Произведено обучение и тестирование нейронной сети.
6. Разработано и протестировано настольное приложение для распознавания психологического подтипа человека (этик, логик) в классификации К.Г. Юнга.

В дальнейшем планируется разработка мобильного приложения и веб-сервиса. Для этого будет проведена работа по уменьшению времени работы разработанного приложения, путем применения новых подходов на различных этапах предобработки и оптимизация существующего программного кода.

ЛИТЕРАТУРА

1. Академическая соционика. [Электронный ресурс] URL: <https://vk.com/academsoc> (дата обращения: 23.05.2018).
2. Алгоритм обратного распространения ошибки. [Электронный ресурс] URL: <http://www.aiportal.ru/articles/neural-networks/back-propagation.html> (дата обращения: 17.04.2018).
3. Бобе А.С., Конышев Д.В., Воротников С.А. Система распознавания базовых эмоций на основе анализа двигательных единиц лица. // Инженерный журнал: наука и инновации, 2016. – № 9 – С.1–16.
4. Галах Н.И., Меркулова Е.В. Использование системы кодировки лицевых экспрессий в задачах распознавания темперамента человека по изображению лица. // Сборник материалов I Всеукраинской научно-технической конференции студентов, аспирантов, молодых ученых, 2010. – Т. II. – С. 634–639.
5. Документация Keras. [Электронный ресурс] URL: <https://keras.io/> (дата обращения: 07.05.2018).
6. Документация Pillow. [Электронный ресурс] URL: <https://pillow.readthedocs.io/en/5.1.x/> (дата обращения: 07.05.2018).
7. Документация PyAV. [Электронный ресурс] URL: <https://mikeboers.github.io/PyAV/> (дата обращения: 07.05.2018).
8. Документация TkInter. [Электронный ресурс] URL: <https://wiki.python.org/moin/TkInter> (дата обращения: 17.05.2018).
9. Иванова С.А., Погорелова Н.Г. Соционика и ее практическое применение // Международный студенческий научный вестник, 2016. – № 5–1. – С. 15–19.
10. Искусственный нейрон. Структура искусственного нейрона. [Электронный ресурс] URL: <http://neuronus.com/theory/151-iskusstvennyj-nejron.html> (дата обращения: 17.04.2018).
11. Крак Ю.В., Кузнецов В.А., Тернов А.С. Анализ мимических проявлений на лице человека. // Искусственный интеллект, 2015. – № 3–4.

– С. 37–50.

12. Лугуев Т.С. Методы компьютерного анализа выражения человеческого лица. // Известия ЮФУ. Технические науки, 2013. – № 1. – С. 251–256.

13. Овсянникова Т.С., Алямкина Е.А., Бурыкин Е.С. Типология личностей и ее практическое применение в условиях развития инновационной экономики. // Вестник Московского университета имени С. Ю. Витте, 2015. – № 1. – С. 120–125.

14. Основания соционики. Введение в психологические типы. [Электронный ресурс] URL: <http://becmology.ru/blog/psychology/socionika01.htm> (дата обращения: 02.03.2018).

15. Официальный сайт dlib. [Электронный ресурс] URL: <http://dlib.net/> (дата обращения: 07.05.2018).

16. Официальный сайт EmoDetect. [Электронный ресурс] URL: <http://neurobotics.ru/psychophysiology/emodetect> (дата обращения: 03.03.2018).

17. Официальный сайт FaceReader Online. [Электронный ресурс] URL: <https://www.facereader-online.com/Technology.aspx> (дата обращения: 03.03.2018).

18. Официальный сайт NumPy. [Электронный ресурс] URL: <http://www.numpy.org/> (дата обращения: 07.05.2018).

19. Официальный сайт OpenFace. [Электронный ресурс] URL: <https://cmusatyalab.github.io/openface/> (дата обращения: 07.05.2018).

20. Официальный сайт TensorFlow. [Электронный ресурс] URL: <https://www.tensorflow.org/> (дата обращения: 24.05.2018).

21. Сверточная нейронная сеть, часть 1: структура, топология, функции активации и обучающее множество. [Электронный ресурс] URL: <https://habr.com/post/348000/> (дата обращения: 17.05.2018).

22. Слой нейронной сети | BaseGroup Labs. [Электронный ресурс]

URL: <https://basegroup.ru/community/glossary/layer> (дата обращения: 24.03.2018).

23. Тюрин А.И., Безыкорнов Д.С. Адаптация системы кодирования движений лицевых мышц к работе в нейросетевой модели. // Современные проблемы науки и образования, 2015. – № 1–2. – С. 1–8.

24. Хайкин С. Нейронные сети: полный курс. / Под ред. Н.Н. Куссуль, 2-е изд. – ООО «И.Д. Вильямс», 2016. – 1104 с.

25. Юнг К.Г. Психологические типы. / Под ред. В.В. Зеленский. – СПб: Азбука, 2001. – 736 с.

26. Dalal N., Triggs B. Histograms of Oriented Gradients for Human. // Detection IEEE, 2005. – P. 886–893.

27. Ebrahimi Kahou S., Michalski V., Konda K. Recurrent Neural Networks for Emotion Recognition in Video. // New York, USA: ACM Press, 2015. – P. 467–474.

28. Facial Emotion Recognition Facial Emotion Recognition. [Электронный ресурс] URL: <https://github.com/JostineHo/mememoji/blob> (дата обращения: 03.03.2018).

29. Hamm J., Hamm J., Kohler C. Automated Facial Action Coding System for dynamic analysis of facial expressions in neuropsychiatric disorders. // Journal of Neuroscience Methods, 2011. – No. 2 (200). – P. 237–256.

30. Kazemi V., Sullivan J. One millisecond face alignment with an ensemble of regression trees. // IEEE Conference on Computer Vision and Pattern Recognition, 2014. – P. 1867–1874.

31. Shu C., Ding X., Fang C. Histogram of the oriented gradient for face recognition. // Tsinghua Science and Technology, 2011. – No. 2 (16). – P. 216–224.

32. Valstar M.F., Patras I., Pantic M. Facial Action Unit Detection using Probabilistic Actively Learned Support Vector Machines on Tracked Facial Point Data. // IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR05 Workshops, 2005. – № 3. – P. 76–76.

ПРИЛОЖЕНИЯ

Приложение 1

Листинг основных функций предварительной обработки видеофайла.

```
def get_preprocessing_data_from_video(path_to_video, image_size):
    # Пытаемся загрузить кеш-файл
    position_last_slash = path_to_video.rfind('/')
    cache_path = CACHE_DIRECTORY + path_to_video[2:position_last_slash + 1]
    file_name=path_to_video[position_last_slash+1:path_to_video.rfind('.')]
    if check_files([file_name], cache_path):
        return library.load_file(file_name, cache_path)
    # Иначе начинаем предобработку
    images = split_video(path_to_video)
    processed_images = numpy.array(make_async_processing_images(images, im-
age_size))
    if CACHE:
        save_file(file_name, cache_path, processed_images)
    return processed_images

def make_async_processing_images(images, image_size):
    threads = ThreadPool(CPU_COUNT)
    processed_images = threads.starmap(get_transform_faces, [(image, im-
age_size) for image in images])
    threads.close()
    threads.join()
    result_images = []
    for processed_image in processed_images:
        if len(processed_image) == 1:
            result_images.append(numpy.array(processed_image[0]
.convert('L')).reshape(image_size, image_size, 1))
    return result_images
```

Приложение 2

Листинг основного метода классификации психологического подтипа.

```
def recognize_psychotype(self):
    preprocessing = PreprocessingController(self._image_size,
                                           self._path_to_video)

    while not self._is_break:
        index_frame, frame = preprocessing.get_current_processed_frame()
        number_frames = preprocessing.get_number_frames()
        self._user_gui.output_preprocessing_progress(frame, index_frame /
number_frames)
        is_there_next_frame = preprocessing.process_next_frame()
        if not is_there_next_frame:
            break

    data_blocks = preprocessing.get_processed_frames()
    if not self._is_break and len(data_blocks) == 0:
        self._user_gui.output_psychotype(None)
        return

    neural_network = NeuralNetworkController(self._file_name_model,
                                           self._image_size)

    total_result = None
    ind = 0
    while not self._is_break:
        self._user_gui.output_recognize_progress(ind / len(data_blocks))
        result = neural_network.predict(data_blocks[ind])
        if total_result is None:
            total_result = result
        else:
            total_result += result
        ind += 1
        if ind == len(data_blocks):
            break

    if not self._is_break:
        total_result /= len(data_blocks)
        number_psychotype = total_result.argmax()
        self._user_gui.output_psychotype(number_psychotype)
```