Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Физико-технический факультет

Кафедра Вычислительной техники и электроники

УДК 004.021			
Бакалаврская работа з	ащищена	Допус	стить к защите в ГЭК
«»2	2019 г.	«»	2019г.
Оценка			дрой д.т.н., проф.
Председатель ГЭК д.т	.н., проф.		В. Н.
Седалищев			
С.П.	Пронин		
ПРОГРАМ	ІМА ОЦЕНКИ ВЛИЯНІ	щоме ки	ИОНАЛЬНОГО
воздействия с п	ОМОЩЬЮ ФОТОИЗО	БРАЖЕНИ	Я НА СОСТОЯНИЕ
	РЕБЕНКА		
]	пояснительная з	АПИСКА	
к выпускной і	КВАЛИФИКАЦИОНН MP 09.04.01.556. 0°		ТЕ БАКАЛАВРА
	обозначение документ	a	
Студент группы ——	556	подпись	И.В. Дудик
Руководитель работы	ст. пр., к.фм.н.		В.В. Белозерских
-	должность, ученое звание	подпись	и.о., фамилия
Консультанты:			
Нормоконтролер	доцент к.фм.н.		А.В. Калачев
	должность, ученое звание	подпись	и.о., фамилия

РЕФЕРАТ

Объем работы, листов	47
Количество рисунков	33
Количество используемых источников	30
Количество таблиц	1
Количество приложений	2

ПРОГРАММА ОЦЕНКИ ВЛИЯНИЯ ЭМОЦИОНАЛЬНОГО ВОЗДЕЙСТВИЯ С ПОМОЩЬЮ ФОТОИЗОБРАЖЕНИЯ НА СОСТОЯНИЕ РЕБЕНКА

Описана разработка программы, оценивающей психоэмоциональное состояние детей. Для определения эмоционального состояния используется способ психометрического анализа, который основывается на лицевых данных. Для реализации был выбран метод на основе программы EmoDetect, в которой реализованы технологии компьютерного зрения и метод обнаружения объектов Виолы Джонса.

Поиск лица на изображении реализуется с помощью метода главных компонент. Сравнение лица тестируемого человека с композитными лицами происходит с помощью сверточной нейронной сети.

Был произведен анализ существующих способов определения эмоций и алгоритмов распознавания лица на изображении.

Реализована и протестирована программа оценки уровня эмоционального состояния детей

Содержание

ВВЕДЕНИЕ
ГЛАВА І. ОБЗОР СУЩЕСТВУЮЩИХ СПОСОБОВ РЕШЕНИЯ
ПОСТАВЛЕННОЙ ПРОБЛЕМЫ
1.1. Анализ решений для определения психоэмоционального состояния
человека
1.1.1. Программно-аппаратный комплекс -нейрокомпьютерный интерфей7
1.1.2. Система контроля психоэмоционального состояния человек
VibraImage
1.1.4. Система распознавания эмоций EQ-Radio
1.1.5. Система распознавания эмоций человека EmoDetect
1.1.7. FaceReader - приложение для автоматического анализа выражений
лица14
1.1.8. Microsoft Oxford Project Emotion Recognition
1.2. Выбор способа определения эмоционального состояния человека 1
ГЛАВА 2. АНАЛИЗ АЛГОРИТМОВ РАСПОЗНАВАНИЯ ЛИЦ19
2.1. Метод главных компонент или Principal Component Analysis 19
2.2. Метод гибкого сравнения на графах
2.3. Активные модели внешнего вида (Active Appearance Models, AAM) 24
2.4. Алгоритм SURF
2.5. Гистограмма направленных градиентов (Histograms of Orientee
Gradients, HOG)
2.6. Свёрточные нейронные сети
2.7. Метод Виолы-Джонса
2.9. Выбор алгоритма для реализации способа определения эмоций 30

	•
ГЛАВА 3. РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ПРОГРАММЫ	37
3.1. Размещение ребенка перед компьютером	37
3.2. Программа обучения нейросети Trainer	39
3.3. Detector	41
3.4. Процесс обучения нейросети	42
3.5. Интерфейс программы	44
ЗАКЛЮЧЕНИЕ	46
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	47
ПРИЛОЖЕНИЕ 1	51
ПРИЛОЖЕНИЕ 2	55

ВВЕДЕНИЕ

В современном мире ІТ-технологии проникают во все сферы жизни человека. Идет тенденция на развитие автоматизированных систем анализа, а именно, систем распознавания лиц и компьютерного зрения. Данные технологии имеют широкое применение как в развлекательных сферах деятельности, например, замены лица на фотографиях или создание 3D-модели лица, так и в сфере обеспечения безопасности, системах слежения за объектом, диагностики в медицине и психологии. Особый интерес ученых, исследующих особенности человеческого поведения, вызывает область распознавания эмоций.

В настоящее время отмечается рост количества эмоциональных и психических отклонений у детей, которые необходимо выявлять как можно быстрее для возможности их своевременной коррекции еще на ранних стадиях развития. Низкая степень проникновения ИТ в область деятельности накладывает ограничения на скорость реакции отклонения. Помочь быстро решить данную задачу возможно только с Ученые использованием современного развития науки техники. вынуждены применять в своей работе старые программные продукты. Многие из существующих ресурсов отсутствуют в открытом доступе, что затрудняет исследовательский процесс в целом. Следовательно, существует необходимость создания программ и программных комплексов автоматизированных систем тестирования, которые бы шли в ногу с современными исследованиями, позволяли бы ученым быстро И обрабатывать полученные данные.

Целью моей работы является разработка нового программного продукта для ускоренного тестирования и комплексной оценки уровня психоэмоционального состояния детей.

Таким образом, цель данной работы влечет за собой ряд задач:

• проведение анализа проблемы для выявления основных критериев оценивания;

- выбор методики тестирования;
- выбор механизма реализации (средства и способы);
- выбор программных и аппаратных средств (средства разработки);
- реализация и тестирование программного продукта в соответствии с выбранным алгоритмом.

ГЛАВА І. ОБЗОР СУЩЕСТВУЮЩИХ СПОСОБОВ РЕШЕНИЯ ПОСТАВЛЕННОЙ ПРОБЛЕМЫ

1.1. Анализ решений для определения психоэмоционального состояния человека

1.1.1. Программно-аппаратный комплекс -нейрокомпьютерный интерфейс

Нейрокомпьютерный интерфейс - это программно-аппаратный комплекс, с помощью которого производятся исследования когнитивных процессов мозга, а также регистрирование изменения уровня внимания, бодрствования, эмоционального состояния человека. [1] Данный способ исследования мозговой активности обладает высоким временным разрешением и позволяет регистрировать быстрые изменения. Это, в свою очередь, соответствует высокой скорости обработки информации мозгом.

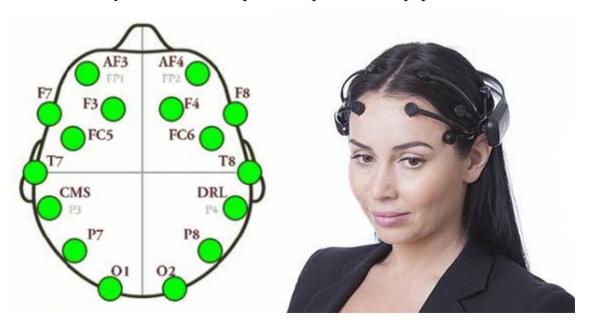


Рис.1.1 Шлем с датчиками для проведения эксперимента.

В основу работы нейро-интерфейса положен принцип метода электроэнцефалографии. Во время проведения эксперимента, испытуемый сидит в
кресле. На голову ему надевают шлем с датчиками, которые в пассивном
режиме регистрируют малейшие изменения активности когнитивных
процессов. В процессе эксперт наблюдает изменения состояния мозговой

активности на мониторе компьютера, и наблюдает, какая из частей мозга активизируется при предъявлении того или иного стимула испытуемому.

1.1.2. Система контроля психоэмоционального состояния человека VibraImage

Эта система контроля психоэмоционального состояния человека предназначена ДЛЯ регистрации, исследования анализа И человека, количественного определения психоэмоционального состояния уровней эмоций, распознавания лжи, психофизиологической диагностики выявления потенциально опасных людей. [2] Главная дистанционного задача этой системы состоит в выявлении людей, которые могут быть предположительно опасными и агрессивными. Это необходимо для того, чтобы обеспечить безопасность в различных охраняемых объектах, аэропортах и т.д.

Процесс оценивания психологического состояния человека строится на основе вестибулярно-эмоционального рефлекса, с помощью программной визуализации вибро-изображения.



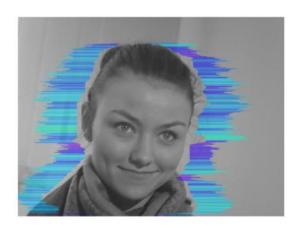


Рис.1.2 Частотная составляющая и внешнее виброизображение на основе частотной и амплитудной составляющей.

Ввод изображения объекта осуществляется с любого источника видео, например, цифровой телевизионной камеры, а программное обеспечение обрабатывает полученную информацию и предоставляет интерфейс для сохранения полученных результатов.

1.1.4. Система распознавания эмоций EQ-Radio

EQ-Radio -ЭТО система распознавания эмоций, основанная исключительно на беспроводных сигналах. передает радиосигнал и анализирует его отражения от тела человека, чтобы определить его эмоциональное состояние. [3] Эта система имеет более высокую точность распознавания по сравнению с другими существующими системами, например, с «Emotion API» компании Microsoft. Основным средством, лежащим в основе EQ-Radio, является новый алгоритм для извлечения отдельных сердечных сокращений из беспроводного сигнала с точностью, сопоставимой с мониторами ЭКГ на теле человека. Полученные удары затем используются для компьютерно-зависимых функций, которые питают классификатор эмоций машинного обучения.

С помощью этого метода непосредственно измеряется взаимодействие эмоций и физиологических сигналов, при этом не требуя от пользователя ношения датчиков на своем теле.

EQ-Radio классифицирует эмоциональное состояние человека в соответствии с известной моделью возбуждения и валентности в одну из четырех основных эмоций: гнев, грусть, радость и удовольствие.

Архитектура системы EQ-Radio состоит из трех компонентов, которые работают конвейерно, как показано на рисунке 1.5:

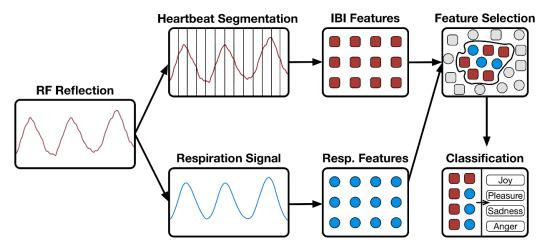


Рис.1.3 Архитектура EQ-Radio.

Архитектура системы EQ-Radio состоит из трех компонентов:

- Радио FMCW, которое передает РЧ-сигналы и фиксирует их отражения от тела человека.
- Алгоритм извлечения биений, который принимает захваченные отражения в качестве входных данных и возвращает последовательность сегментов сигнала, что соответствует индивидуальному сердцебиению человека.
- Подсистема классификации эмоций, которая вычисляет эмоционально-значимые характеристики из полученных физиологических сигналов, т.е. характер дыхания и сердцебиения человека, и использует эти функции для распознавания эмоционального состояния человека.

EQ-Radio способно распознавать эмоции субъекта со средней точностью 72,3%. [4] Данное устройство так же применимо в сферах развлечения, в быту, в медицине.

1.1.5. Система распознавания эмоций человека EmoDetect

Система распознавания эмоций EmoDetect [5] позволяет оценить психоэмоциональное состояние человека путем определения шести базовых эмоций: счастье, удивление, печаль, злобу, страх, отвращение.

Программа EmoDetect содержит в себе принципы кодирования лицевых движений FACS Action Units, разработчиком которой является Пол Экман - профессором Калифорнийского Университета в Сан-Франциско.

В ходе исследования человеку показывают на мониторе различные стимулы. Перед испытуемым расположена камера, снимающая видео. Система обрабатывает видео. Происходит это следующим образом: захватывается изображение человека, выражение лица которого должно быть классифицировано. Затем лицо извлекается из изображения с помощью Метода Джонса для обнаружения объектов. Затем алгоритмы извлечения элементов извлекают соответствующие элементы изображения Обученные классификаторы обрезанного лица. сопоставляют эти характеристики с соответствующими эмоциями. Код реализуется с помощью OpenCV, библиотеки компьютерного зрения с открытым исходным кодом.

Далее, эмоции фиксируются с привязкой времени демонстрации и результаты выгружаются в форме графиков и таблиц.

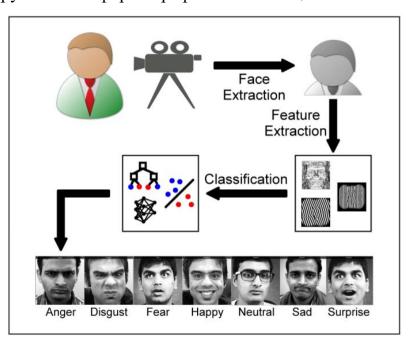


Рис. 1.4 Обзор системы.

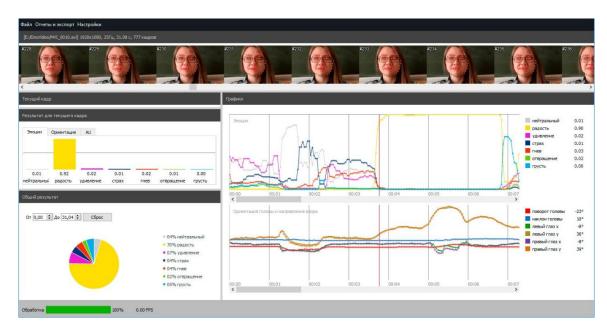


Рис.1.5 Основное окно программы EmoDetect.

Для определения эмоций необходимо ручное закрепление опорного кадра и с помощью него выбор нейтрального изображения, которое в дальнейшем программа будет принимать за точку отсчета и фиксировать эмоцию с соответствующими поправками. (Рис.1.8) Это является главным преимуществом программы ЕтоDetect перед существующими аналогичными продуктами.

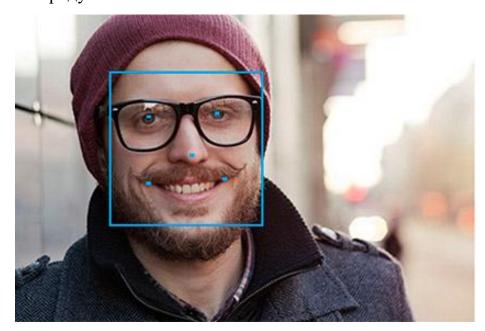


Рис.1.6 Закрепление опорного кадра - базового выражения лица испытуемого.

Преимущества:

- В программе реализовано три независимых алгоритма распознавания эмоций: с вводным изображением, нейронная сеть без вводного изображения, нейронная сеть с вводным изображением.
- Построение графиков зависимостей эмоции от кадра и зависимости ориентации головы от кадра, на которых фиксируются интенсивность, с которой изменяется эмоция человека во времени и формируется отчет о результатах обработки изображения или видео.
- Реализована возможность записи видео с камеры и поддержка операционной системой всех форматов видео.

Недостатки:

• Как и многие другие из существующих программ для определения эмоций, EmoDetect не определяет лицо, если оно повернуто в профиль.

В 2017 году Российская компания NtechLab, разработчик известного сервиса FindFace, позволяющего найти человека по фотографии среди миллионов пользователей социальных сетей, стала победителем первого в мире чемпионата по распознаванию эмоций людей при помощи компьютерных алгоритмов EmotionNet Challenge.

Участникам чемпионата предстояло распознать 16 основных эмоций. Каждая эмоция определятся по уникальной вариации сокращения лицевых мышц. Если действие одних мышц хорошо видно, то других — едва заметно даже для человека.

Основная трудность с определением эмоций по выражению лица в естественных условиях была связана с тем, что лица не всегда были

1.1.7. FaceReader - приложение для автоматического анализа выражений лица

FaceReader - программа для определения эмоционального выражения лица. Оно может в режиме онлайн быстро и точно определить по выражению лица шесть основных эмоций: счастье, грусть, злость, удивление, испуг, отвращение + спокойствие и презрение. [6] Кроме того, он может обнаружить состояние лица (левый и правый глаз открыт или закрыт, рот открыт или закрыто и брови подняты, нейтральны или опущены), направление взгляда испытуемого и отслеживать ориентацию головы. FaceReader также определяет, является ли эмоциональное состояние человека положительным или отрицательным, и распознает, насколько активен человек. Программа может также указать пол человека, возраст, этническую принадлежность, количество лицевых волос (борода или усы), и носит ли очки человек или нет.

FaceReader может использоваться в самых разных областях исследований:

- Психология как люди реагируют на определенные раздражители, например, в когнитивных исследованиях.
- Образование наблюдение за выражением лица учащихся может способствовать развитию образовательной системы.
- Взаимодействие человека и компьютера выражения лица могут предоставить ценную информацию о пользовательском опыте.
- Исследование рынка поведение потребителей, например, как люди реагируют на дизайн нового рекламного ролика.

Программа не требует каких-либо дополнительных настроек и обучения. В программе реализованы технологии компьютерного зрения.

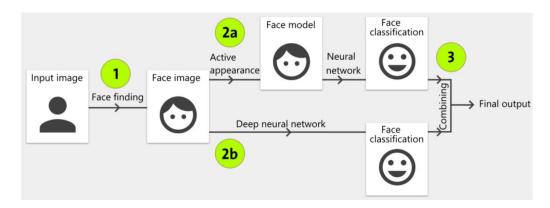


Рис.1.7 Этапы классификации выражения лица в программе FaceReader.

FaceReader классифицирует выражения лица в несколько этапов:

- 1. Определение лица положение лица на изображении определяется с помощью метода, называемого алгоритм каскадного классификатора Виолы Джонса, разработанный для поиска лица на изображении.
- 2. FaceReader использует два метода, которые работают бок о бок. В более старых версиях FaceReader используется только верхний метод (2a). Используя этот метод в сочетании с Deep Face классификация (2b) делает анализ более надежным и точным.

Преимущества программы:

- FaceReader может анализировать выражение лица с точностью до 95%. Для некоторых выражений точность будет выше, для других ниже.
- На распознавание эмоций не влияет поворот лица или наклон головы.
- Программа работает с многими форматами видеофайлов, такими как MPEG1, MPEG2, DivX4, DivX5, DivX6, DV-AVI. В преимущество к этому, распознавание эмоций может производиться как покадрово, так и полностью при просмотре всего видео. Приложение так же работает и со статичными изображениям и собирает данные с камеры устройства в реальном времени.
- Возможности визуализации: наличие гистограмм, диаграмм, отображение накладываемой сетки, процентов выражаемых эмоций.

Недостатки программы:

- При работе с лицами детей 5-го возраста, успех работы программы низкий, т.к. процент распознавания эмоций значительно падает.
- Очки на лице могут мешать классификации и значительно уменьшить процент производительности.
- Поза, движения и вращения испытуемого ограничены. Например, если лицо повернуто в профиль, программа не будет его распознавать. Тестируемый должен стоять или сидеть и смотреть спереди в камеру (под углом <40%). Так же, FaceReader требует строгих условий освещения.

1.1.8. Microsoft Oxford Project Emotion Recognition

Проект является одним из самых новых среди разработок Microsoft Oxford в области распознавания эмоций. Принцип работы Emotion Recognition [7] заключается в следующем: технологии, опираясь на машинные алгоритмы, анализируют загруженную фотографию на наличие лиц, затем определяют восемь эмоций (гнев, презрение, отвращение, страх, счастье, спокойствие, грусть, удивление).

Т.к. проект находится в тестовом режиме работы, его результаты по большей части могут получаться менее точными, чем у других существующих программ. Разработчики разрешили бесплатно в доступной форме использовать АРІ сервис, что в свою очередь, сделало его широко распространенным.

Недостатком данного приложения является то, что определение эмоций может происходить только на статичных изображениях. Так же, программа не распознает лица, которые находятся под углом >40%.

На данный момент доступна только демонстрационная версия приложения.

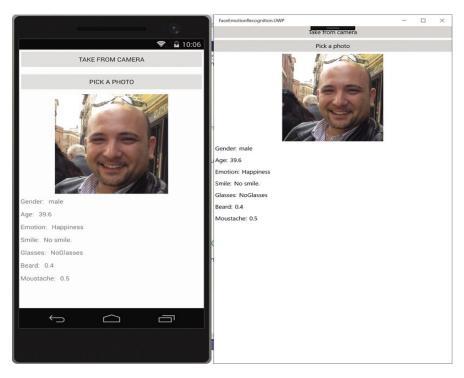


Рис.1.8 Распознавание лиц и эмоций в кросс-платформенном приложении Xamarin. Forms (Android-устройство слева, настольная версия Windows 10 справа).

1.2. Выбор способа определения эмоционального состояния человека

В ходе анализа выше рассмотренных мною способов определения психоэмоционального состояния людей, я убедилась в том, что на данный момент существует большое количество различных решений для оценки эмоционального состояния человека. Все способы имеют свои преимущества и недостатки. Можно сделать вывод, что оценка психологического типа с помощью программно- аппаратных комплексов или датчиков дает наиболее точные результаты, по сравнению с другими программными методами. Тем не мене, этот процесс является наиболее трудоемким и затратными. Использование датчиков тела является громоздкой процедурой, что делает этот подход непригодным для регулярного использования.

Оценка психического состояния путем тестирования или опроса требует, во-первых, много времени на ее реализацию, во-вторых, проведено мало исследований, где с помощью тестов эксперты точно и быстро составляли психоэмоциональный портрет человека.

Наиболее универсальным и подходящим способом для реализации поставленных мною задач является способ психометрического анализа, который основывается на лицевых данных, т.к. он вполне просто реализуем, не требует наличия дорого и сложного оборудования, и при этом могут достаточно точно и быстро определять эмоциональное состояние человека.

Исходя из изложенных ранее выводов, для решения поставленной задачи мною был выделен способ распознавания эмоций по методу программы FaceReader. Однако, данный метод имеет существенный недостаток для моей задачи- процент получения успешного результата во время выполнения программы значительно падает при работе с детьми 5-6 лет. [8] Поэтому в альтернативу был выбран метод на основе программы ЕтоDetect, в которой реализованы технологии компьютерного зрения и метод обнаружения объектов Виолы Джонса.

ГЛАВА 2. АНАЛИЗ АЛГОРИТМОВ РАСПОЗНАВАНИЯ ЛИЦ

2.1. Метод главных компонент или Principal Component Analysis

Анализ основных компонентов является старейшим и наиболее известным из методов многомерного анализа. Впервые он был введен Пирсоном (1901 г.) и независимо разработан Хотеллингом (1933 г.). [9] Так же этот анализ известен как преобразование Кархунена-Лоэва.

Цель анализа основных компонентов состоит в том, чтобы уменьшить большую размерность пространства данных (наблюдаемые переменные) до меньшей внутренней размерности пространства признаков (независимых переменных), которые необходимы для экономичного описания данных.

Основная идея использования этого анализа для распознавания лиц состоит в том, чтобы выразить большой вектор пикселей, построенный из изображения лица, в компактные основные компоненты пространства признаков, т.е. векторы малой размерности. Вначале весь обучающий набор лиц преобразуется в одну общую матрицу данных, каждая строка которой представляет из себя один экземпляр изображения лица.[10]

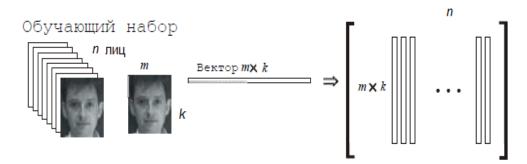


Рис. 2.1 Преобразования обучающего набора лиц в одну общую матрицу.

Собственные векторы ковариационной матрицы получены из набора изображений лица (векторов). 2-D изображение лица может быть представлено как 1-D вектор путем объединения каждой строки (или

столбца) в длинный тонкий вектор. Изображения центрируются по среднему значению путем вычитания среднего изображения из каждого вектора изображения. Далее вычисляется матрица ковариации. Для этой матрицы собственные рассчитываются векторы, соответствующие ненулевым собственным значениям ковариационной матрицы. Они сортируются от высокого к низкому согласно их соответствующим собственным значениям и остаются только первые к векторов по правилу, представленному на рисунке 2.2.

Principal Component Analysis

Covariance Algorithm

- Lero mean: $x_{ij}=x_{ij}-\mu_i$ $\mu_i=\frac{1}{n}\sum_{j=1}^n x_{ij}$ Unit variance: $x_{ij}=\frac{x_{ij}}{\sigma_j}$ $\sigma_i=\frac{1}{n-1}\sum_{j=1}^n (x_{ij}-\mu_i)^2$
- Covariance matrix: $\Sigma = XX^T$
- 4. Compute eigenvectors of Σ : W^T
- 5. Project X onto the k principal components

Рис.2.2 Алгоритм РСА.

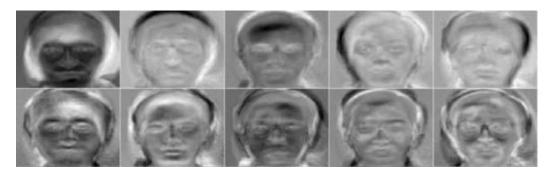


Рис. 2.3 Пример первых десяти собственных векторов (собственных лиц), полученных на обучаемом наборе лиц.

Предложенный метод был протестирован в базе данных лиц Oracle Research Laboratory (ORL), содержащей 400 изображений (40 классов).. Результаты теста показали уровень распознавания 99,50%.[11]

Так же, использование анализа в среде FPGA показало, что алгоритм главных компонент повышает точность распознавания лица, когда изображения имеют различную экспрессию и освещенность. Архитектура смогла выполнить распознавание лица за 11 мс для базы данных с 1000 изображений лица.[12]

При тестировании алгоритма рекомендуется использовать стандартный набор тестовых данных, чтобы исследователи могли напрямую сравнивать результаты. В настоящее время используется ряд баз данных, некоторые наборы данных лица, часто используемые исследователями, представлены в таблице 2.1 [13]

Типичные базы данных с наборами данных лица

Таблица 2.1

База	Количес	Условия	Разреш	Количе
данных	тво предметов		ение	ство
			изображения	изображений
База	116 (63	Выражени	768×57	3288
данных AR	мужчины и 53	я лица	6	
	женщины)	Освещени		
		e		
		Время		
		окклюзии		
FERET	1199	Выражени	256×38	1451
(технология		я лица	4	
распознавания лица)		Освещени		
		e		

		Время		
База данных МІТ	16	Ориентац ия головы	120×12 8	433
		освещение		
		Масштаб		
Йельская	15	С/без	320×24	165
база для лица		очков	3	
		Освещени		
		e		
		Выражени		
		я лица		

2.2. Метод гибкого сравнения на графах

Данный метод заключается в эластичном сопоставлении графов, описывающих изображения лиц. [14] Лица представляются в виде графов с узлами, расположенными в опорных точках (глаза, нос, уголки губ и т.д.). Края помечаются двумерными векторами расстояния, причем каждый узел содержит набор из 40 комплексных вейвлет-коэффициентов Габора в разных масштабах и ориентациях. Они называются «джетами», а распознавание основано на помеченных графах. Сопоставление графа упругого сгустка (ЕВGM) использует граф модели для представления лица человека и кодирует локальный внешний вид с помощью «вейвлет-струй».[15]

Джет Габора в основном представляет собой узел на упругой сетке, обозначенный кружками на данном изображении, как показано на рисунке 2.4

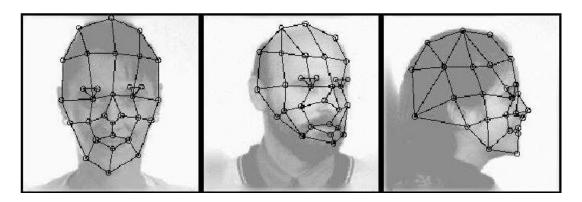


Рис.2.4 Граф на основе антропометрических точек лица.

Эти узлы описывают поведение изображения вокруг данного пикселя. Когда изображение свернуто с помощью фильтра Габора, полученный результат может быть использован для обнаружения формы и для выделения элементов с использованием обработки изображения. Свертка объединяет функции вместе и выражает степень совпадения функций. Сходство, основанное на результате работы фильтра Габора на каждом узле, и является основой распознавания.

Достоинства метода:

• метод очень эффективен к изменениям ракурса лица

Недостатки:

- низкая скорость работы
- высокая вычислительная сложность алгоритма
- данный алгоритм не эффективно использовать в системах, которые требуют высокую скорость работы, приблеженную к системам, которые работают в реальном времени.

Одной из самых ранних работ являются исследования Брунелли и Поджио [16], которые в своем эксперименте использовали 35 геометрических элементов из базы данных 47 разных людей с 4 изображениями на человека, как показано на рисунке 2.5, и достигли показателя производительности 95%

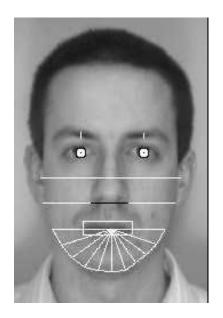


Рис.2.5 Геометрические особенности, используемые Брунелли и Поджио

2.3. Активные модели внешнего вида (Active Appearance Models, AAM)

Тим Кутс и Крис Тейлор в 1998 году предложили метод активных моделей внешнего вида, которые способны «объяснить» внешний вид лица с помощью компактного набора параметров модели. [17] AAM - это интегрированная статистическая модель изображения, которая может быть подогнана под реальное изображение, с помощью различных деформаций. [20] Этот метод включает в себя объединение модели параметров формы со статической моделью изображения (текстурой). (2.1 и 2.2)

$$x = x^{-} + Qsc (2.1)$$

$$g = g^- + Qgc (2.2)$$

Здесь х обозначает среднюю форму, g обозначает среднюю текстуру в пластинке средней формы, а Qs, Qg представляют собой матрицы, описывающие способы изменения, полученные из обучающего набора. Форма в кадре изображения, X, может быть сгенерирована путем применения подходящего преобразования к точкам (2.3)

$$x: X = St(x) \tag{2.3}$$

Текстура в кадре изображения генерируется путем применения масштабирования и смещения к интенсивностям (2.4)

$$gim = Tu(g) = (u1 + 1)gim + u21$$
 (2.4)

Параметры модели внешнего вида, с, и параметры преобразования формы, t, определяют положение точек модели в кадре изображения, X, который дает форму фрагмента изображения, который будет представлен моделью.

Под «текстурой» подразумевается образец интенсивностей или цветов на участке изображения. Чтобы построить форму модели активного внешнего вида, необходим обучающий набор с помеченными изображениями, т.е. для построения модели лица требуются изображения лиц с отмеченными точками, определяющими основные черты лица. (Рис.2.6)



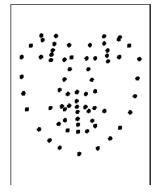




Рис. 2.6 Пример подробной разметки.

Эта форма отображает внешний контур лица, контуры рта, глаз, носа, бровей. Затем применяется анализ Прокруста для выравнивания наборов точек и построения статистической модели формы. Далее деформируется каждое тренировочное изображение так, чтобы точки соответствовали

точкам средней формы, получая участок без формы °(Рис.2.6). Наконец, корреляции между формой и текстурой изучаются для создания комбинированной модели внешнего вида.

Представленный алгоритм, не смотря на свою эффективность, имеет существенный недостаток: требует большого объема вычислений и не подходит для применения в системах реального времени.

2.4. Алгоритм SURF

Алгоритм SURF (Speeded-Up Robust Features) используется для генерации дескрипторов функций и является надежным и быстрым детектором локальных особенностей в изображении. Современный алгоритм является достаточно быстрым и эффективным, и его реализация есть во множестве математических библиотек.[18]

Векторы признаков через алгоритм SURF формируются с помощью локальных шаблонов вокруг ключевых точек, которые обнаруживаются с использованием масштабированного фильтра. Алгоритм использует разномасштабные фильтры для нахождения гессиана, т.к. гессиан инвариантен к вращению, но не инвариантен к масштабу.

Ниже приведены основные этапы определения векторов признаков SURF конкретного изображения:

- 1) Детектор ключевых точек: на этом этапе ключевые точки SURF обнаруживаются с использованием аппроксимации матрицы Гессена. Поиск, в свою очередь, тоже должен обладать инвариантностью.
- 2) Дескриптор ключевой точки: на этом этапе описываются ключевые точки и создание их дескрипторов инвариантных к масштабу и вращению, т.е. описание ключевой точки будет одинаковым, даже если образец изменится в размере или положении.

Детерминант матрицы Гессе достигает экстремума в точках максимального изменения градиента яркости. Он хорошо детектирует пятна, углы и края линий.

Для каждой ключевой точки считается направление максимального изменения яркости (градиент) и масштаб, взятый из масштабного коэффициента матрицы Гессе. Градиент в точке вычисляется с помощью фильтров Хаара.

На рис. 2.7 показаны примеры ключевых точек, обнаруженных с помощью дескриптора SURF.На данном изображении композиция имеет другой масштаб, углы обзора, и местами заслонена другим объектом. Однако, вопреки этому, ключевые точки идентифицируются достаточно точно. Диаметр круга показывает масштаб особой точки, а зеленая линия — направление градиента яркости.[19]



Рис. 2.7 Особые точки изображения в образце(слева) и на сцене(справа).

Дескриптор ключевой точки представляет собой набор из 64 (или 128) значений и вычисляется из ориентированной квадратной области локального

изображения вокруг ключевой точки. Благодаря этому, обеспечивается различие дескрипторов для ключевых точек.

Алгоритм SURF не только показывает более высокие скорости сопоставления, он также превосходит хорошо известный SIFT на окклюдированных гранях базы данных AR.[20]

2.5. Гистограмма направленных градиентов (Histograms of Oriented Gradients, HOG)

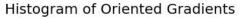
Представленный в 2005 году метод гистограммы направленных градиентов (HOG) - одна из известных функций распознавания объектов и справляется с задачей распознавания достаточно быстро и надежно.

Дескрипторы гистограммы направленного градиента (HOG) эффективны при обнаружении объектов, в частности, при распознавании человека. Эти функции были использованы для решения различных проблем, включая распознавание и отслеживание пешеходов, распознавание жестов рук для перевода на язык жестов и интерфейсы на основе жестов, распознавание лиц и распознавание частей тела.[21]

Внешний вид и форму локального объекта часто можно довольно охарактеризовать распределением локальных хорошо градиентов Особенности HOG интенсивности или краевым отклонением. рассчитываются путем взятия гистограмм направленной интенсивности краев в локальной области. [22] Дескриптор строится следующим образом: окно изображения разделяется на пространственные ячейки, каждой из которых присваивается гистограмма направлений градиентов для пикселей ячейки.

Обрабатываемое изображение обычно делает черно-белым для увеличения эффективности и точности .Так же, чтобы добиться большей инвариантности дескрипторов к освещению .требуется контрастная нормализация локальных гистограмм.





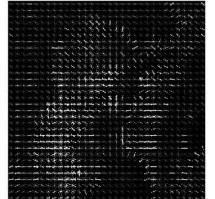


Рис. 2.8 Результат обработки изображения HOG.

Сетки дескрипторов НОG значительно превосходят существующие наборы функций для обнаружения человека. Локально нормализованные дескрипторы НОG обеспечивают отличную производительность по сравнению с другими существующими наборами функций, включая вейвлеты.

Хорошо известно, что особенности НОG устойчивы к локальным геометрическим и фотометрическим преобразованиям. Если перемещения или повороты объекта намного меньше, чем локальный размер ячейки, их влияние невелико.

Кроме того, дескрипторы HOG также используются в задачах, которые не являются ориентированными на человека, таких как классификация ориентации транспортного средства, проблема, относящаяся к автономным транспортным средствам и т.д.

2.6. Свёрточные нейронные сети

Сверточная нейронная сеть (CNN) - специальная архитектура глубоких искусственных нейронных сетей с прямой связью, предложенная в 1988 году и предназначенная для распознавания изображений.[23] CNN показывет отличную производительность во многих проблемах компьютерного зрения и машинного обучения. Приложения CNN включают классификацию

изображений, семантическую сегментацию изображений, обнаружение объектов в изображениях и т.д. В категоризации изображений каждое изображение имеет основной объект, который занимает большую часть изображения. Изображение классифицируется на один из классов на основе идентичности его основного объекта, например собаки, самолета, птицы и т. д.

Сверточные нейронные сети представляют собой особый вид ANN(Artificial Neural Networks), Они хорошо работают в качестве экстрактора признаков при использовании изображений в качестве входных данных и способны работать в режиме реального времени. Это позволяет использовать необработанные входные изображения без какой-либо предварительной обработки или постобработки.

Сама архитектура нейронной сети строится на некоторых особенностях структуры коры головного мозга. Архитектура глубоких сверточных нейронных сетей (Рис.2.9) состоит из четырех частей. Первая часть автоматически обрабатывает данные. Этот этап начинается с Convolution 1, который применяет 64 различных фильтра. Следующим уровнем является Pooling 1, который понижает частоту дискретизации изображений, а затем они нормализуются на уровне LRN 1. [24]

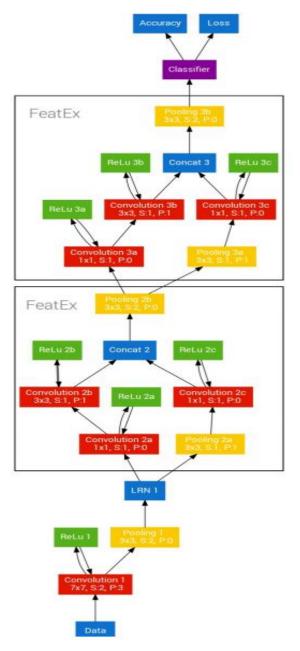


Рис.2.9 Архитектура сверточной нейронной сети.

Следующими шагами являются два блока FeatEx (Parallel Feature Extraction Block), (Puc.2.8). Они являются ядром предлагаемой архитектуры. Функции, извлеченные этими блоками, передаются на полностью связанный слой, который использует их для классификации входных данных по различным эмоциям. Описанная архитектура компактна, что делает ее не только быстрой для обучения, но и пригодной для приложений реального времени.

Для обучения, обычно, используется метод обратного распространения ошибки. Название данная архитектура получила благодаря использованию операции свертки, которая заключается в поэлементном умножении каждого фрагмента изображения на ядро свёртки и последующей записи результата в соответствующую позицию выходного изображения.

Несмотря на то, что архитектура свёрточных нейронных сетей была разработана в конце прошлого века, аппаратная составляющая, которая позволила эффективно применять их и им подобные архитектуры появилась сравнительно недавно, и с того момента свёрточные сети являются самым распространённым методом в задаче распознавания изображений.

2.7. Метод Виолы-Джонса

Метод был разработан и представлен в 2001 г. Полом Виолой и Майклом Джонсом.[25] В 2004 году в Международном журнале «Компьютерное зрение» была опубликована статья Пола Виолы и Майкла Дж. Джонса под названием «Надежное распознавание лиц в реальном времени». Алгоритм, представленный в этой статье, оказался настолько успешным, что сегодня он очень близок к тому, чтобы быть стандартом для решения задач по распознаванию лиц. Этот успех в основном объясняется относительной простотой, быстрым выполнением И замечательной производительностью алгоритма.

Виола-Джонс ограничивается полным обзором фронтальных вертикальных лиц. То есть, чтобы быть обнаруженным, все лицо должно быть направлено в сторону камеры, и оно не должно быть наклонено ни в какую сторону. Это может немного поставить под угрозу требование быть неограниченным, но, учитывая, что алгоритм обнаружения чаще всего будет следовать алгоритму распознавания, эти требования представляются вполне разумными.

Типичное входное изображение для алгоритма обнаружения лица показано на рисунке 2.10. Это изображение имеет относительно низкий контраст, содержит много различных видов текстур и, наконец, содержит много лиц.

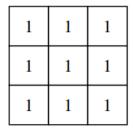


Рис.2.10 Типичное входное изображение для алгоритма.

Основной принцип алгоритма Виолы-Джонса заключается в сканировании подокна, способного обнаруживать лица по заданному входному изображению. Стандартный подход к обработке изображений состоял бы в том, чтобы изменить масштаб входного изображения до различных размеров и затем запустить детектор фиксированного размера через эти изображения. Этот подход оказывается довольно трудоемким из-за вычисления изображений разного размера. Вопреки стандартному подходу Виола-Джонс изменяет масштаб детектора вместо входного изображения и запускает детектор много раз через изображение - каждый раз с другим

размером. Пол Виола и Майкл Джонс изобрели масштабно-инвариантный детектор, который требует одинакового количества вычислений независимо от размера. Этот детектор построен с использованием так называемого интегрального изображения и некоторых простых прямоугольных элементов, напоминающих признаки Хаара.

Первым шагом алгоритма обнаружения лиц Виолы-Джонса является преобразование входного изображения в целостное изображение. Это делается путем создания каждого пикселя равным всей сумме всех пикселей выше и слева от соответствующего пикселя. Это показано на рисунке 2.11



Input image

1	2	3
2	4	6
3	6	9

Integral image

Рис.2.11 Интегральное изображение.

Это позволяет рассчитать сумму всех пикселей внутри любого данного прямоугольника, используя только четыре значения. Эти значения представляют собой пиксели в интегральном изображении, которые совпадают с углами прямоугольника во входном изображении. Это показано на рисунке 2.12.

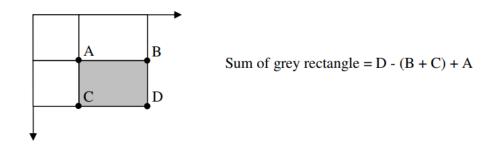


Рис.2.12 Расчет суммы

Поскольку оба прямоугольника В и С включают в себя прямоугольник А, сумма А должна быть добавлена к вычислению.

На этом этапе было продемонстрировано, как сумма пикселей в прямоугольниках произвольного размера может быть вычислена за постоянное время. Детектор лица Viola-Jones анализирует данное подокно, используя элементы, состоящие из двух или более прямоугольников. Различные типы функций показаны на рисунке 2.13.

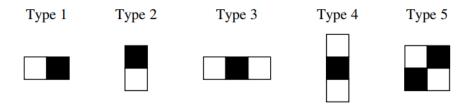


Рис.2.13 Различные типы функций

Каждый элемент приводит к единственному значению, которое вычисляется путем вычитания суммы белого прямоугольника из суммы черного прямоугольника.

Детектор с базовым разрешением 24 х 24 пикселей показывает удовлетворительные результаты. Если учесть все возможные размеры и положения элементов на рисунке 4, то можно построить около 160 000 различных вариантов расположения признаков. Далее все найденные признаки попадают к классификатору, который «выносит вердикт».[26]

2.9. Выбор алгоритма для реализации способа определения эмоций

В предыдущей главе были рассмотрены существующие методы и алгоритмы распознавания лиц. Главным недостатком алгоритмов является то, что многие из них предъявляют строгие требования к условиям съемки и слишком чувствительны к условиям освещенности, что, в свою очередь, влечет за собой получение неправильного результата. Так же некоторые методы имеют высокую вычислительную сложность, что затрудняет процесс реализации.

Однако, со временем компьютерная реализация этих алгоритмов значительно совершенствуется и модернизируется, поэтому рассмотренные выше методы достаточно эффективны для решения задачи распознавания лица.

Поскольку метод главных компонент является достаточно эффективным, а точность распознавания лица не требует строгих условий к освещенности и не зависит от изменения положения лица, для реализации способа отлично подходит именно он.

Так же для реализации способа оптимальным являются алгоритмы с применением свёрточных нейронных сетей.

Для реализации задачи выравнивания лица оптимальным решением является алгоритм, основанный на антропометрических точках активных моделей внешнего вида (Active Appearance Models, AAM). Работа данного алгоритма заключается в том, что создается цикл, в котором изображение смещается на несколько градусов до тех пор, пока значение точек соответствующих точек лица не будет равным относительно оси X и У.

ГЛАВА 3. РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ПРОГРАММЫ

3.1. Размещение ребенка перед компьютером

Реализация поставленных задач требует, в первую очередь, правильного размещения ребенка перед монитором и регулирования условий проведения эксперимента для получения как можно более точных результатов тестирования программы.

На рисунке 3.1 подробно представлена схема размещения ребенка перед компьютером. Ребенок сидит напротив монитора, диагональю 15,5 дюймов, расстояние обзора до которого должно быть 50 -70см (чем больше монитор, тем больше должно быть расстояние между ним и глазами ребенка). Такое расстояние позволит полностью поместить в поле зрения ребенка выводимое на экран изображение.[27] Если зрение не позволяет выдерживать это расстояние, тогда необходимо уменьшить разрешение изображения или регулировать расстояние между ребенком и монитором вручную. Уровень глаз должен находиться чуть выше центра монитора под углом примерно 15 (при наклоне головы в пределах нейтрального положения -4). [28]

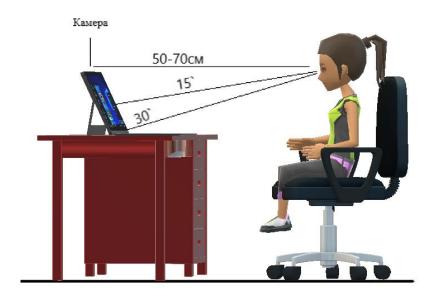


Рис. 3.1 Схема размещения ребенка перед монитором.

Ракурс обзора на лицо ребенка, сидящего прямолинейно по отношению к камере, должен быть максимально прямым. В поле зрения камеры необходимо исключить предметы, перекрывающие обзор. Лицо ребенка должно быть равномерно освещено, не должно допускаться появление резких теней на лице или одностороннего освещения, резких перепадов освещения лица.

На экран, с интервалом в 2 секунды, выводятся различные статические изображения. В качестве исходного стимульного материала будут использоваться цветные изображения базовых эмоциональных экспрессий из базы Radboud Faces Database (Langer et al., 2010) - радость, удивление, печаль, гнев, спокойствие, презрение[29], представленные на рисунке 3.2.

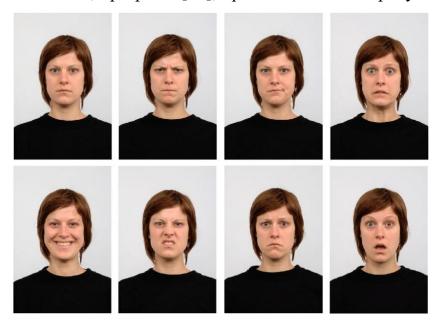


Рис. 3.2 Базовые эмоции из базы Radboud Faces Database.

Для создания программы был выбран язык программирования Руthon. Это мощный широко используемый универсальный язык программирования высокого уровня. Его концепции проектирования позволяют просто и удобно работать с кодом, а его синтаксис позволяет программистам выражать идеи в меньшем количестве строк кода, чем это было бы возможно в таких языках, как С ++ или Java.[30] Руthon имеет простой подход к объектноориентированному, императивному и функциональному программированию.

Так же он содержит динамическую типизацию и имеет обширную стандартную библиотеку.

Являясь кроссплатформенным и имея простую и достаточно мощная среду разработки IDLE, Python берет значительное преимущество перед другими языками программирования.

Так же для работы с компьютерным зрением в реальном времени была выбрана библиотека с открытым кодом OpenCV (Open Source Computer 500 функций Vision), содержащая более распознавания ЛИЦ И предоставляющая простую в использовании инфраструктуру компьютерного зрения.[29] Библиотека направлена на обеспечение основных инструментов, необходимых для решения проблем компьютерного зрения и имеет большое количество подключаемых модулей для различных задач. Библиотека свободно распространяема, удобная в использовании, написана на С, С ++ и Python, и работает под Linux, Windows и другие ОС. Помимо OpenCV была использована еще одна удобная библиотека для работы с компьютерным зрением Dlib, содержащая в себе функции и классы, которые не реализованы в OpenCV, такие как HOG-детектор для поиска лица.[31]

3.2. Программа обучения нейросети Trainer

В проекте, помимо вспомогательных скриптов, существуют два основных скрипта— Trainer, который обучает нейросеть и Detector.

TRAINER

Внутри скрипта находится несколько библиотек для работы с компьютерным зрением. Это библиотека питру, добавляющая поддержку больших многомерных массивов и матриц, вместе с большой библиотекой высокоуровневых и очень быстрых математических функций для операций с этими массивами, библиотеку машинного обучения tenserflow— для работы с нейронными сетями на основе матрицы, и все остальные вторичные библиотеки. Программа получает на вход созданный массив изображений и переводит его в матрицу.

```
insize = int(input, size):
   insize = int(input.get_shape()[1])
   W = weight_variable([insize, size])
   b = bias_variable([size])

return tf.matmul(input, W) + b
```

Рис. 3.3 Создание матрицы.

Далее на основе созданных изображений создаются наборы матриц, где каждая ячейка матрицы показывает свои свойства для определенной картинки.

```
def test(emoji_data, sess):
    logger.info('CALCULATING TESTSET ACCURACY ...')
    L = len(emoji_data.test.labels)

x = emoji_data.test.images
y = emoji_data.test.labels

accs = []

for i in tqdm.tqdm(range(0, L, 30)):
    if i+30 <= L:
        x_i = x[i:i+30].reshape(30, 48, 48, 1)
        y_i = y[i:i+30].reshape(30, len(EMOTION_MAP))
    else:
        x_i = x[i:].reshape(L-i, 48, 48, 1)
        y_i = y[i:].reshape(L-i, len(EMOTION_MAP))

accs.append(sess.run(accuracy, feed_dict={X:x_i, Y:y_i, keep_prob:1.0}))

acc = np.mean(accs)</pre>
```

Рис. 3.4 Создание матриц с критическими параметрами.

Матрица строится 48х48, каждая ячейка которой имеет определенный критический параметр, который схож с конкретным изображением.

```
with tf.Session(config=config) as sess:

sess.run(tf.global_variables_initializer())

for i in tqdm.tqdm(range(STEPS)):
    x_data, y_data = emoset.train.next_batch(BATCH_SIZE)

acc, loss, _ = sess.run(
    [accuracy, cross_entropy, train],
    feed_dict={X:x_data, Y:y_data, keep_prob: KEEP_PROB}
)

if i % 20 == 0:
    logger.info('accuracy: {:.4}%, loss: {:.4}'.format(
         acc*100, loss
    ))

test(emoset, sess)
```

Рис. 3.5 Массив выяснения точности предсказаний.

```
clef model(x, keep_prob):
    C1, C2, C3 = 30, 50, 80
    F1 = 512
    convl_1 = conv_layer(x, shape=[3, 3, 1, C1])
    convl_1 pool = max_pool_2x2(convl_1)

convl_2 = conv_layer(convl_1 pool, shape=[3, 3, C1, C2])
    convl_2 pool = max_pool_2x2(convl_2)

convl_drop = tf.nn.dropout(convl_2 pool, keep_prob=keep_prob)

conv2_1 = conv_layer(convl_drop, shape=[3, 3, C2, C3])
    conv2_1 pool = max_pool_2x2(conv2_1)

conv2_flat = tf.reshape(conv2_1 pool, [-1, 6*6*C3])
    conv2_drop = tf.nn.dropout(conv2_flat, keep_prob=keep_prob)

full1 = tf.nn.relu(full_layer(conv2_drop, F1))
    full1_drop = tf.nn.dropout(full1, keep_prob=keep_prob)
```

Рис. 3.6 Формирование слоя с моделью.

3.3. Detector

```
def from_cam(sess):
    start_time = int(time.time())
     face cascade = cv2.CascadeClassifier(config parser['OPEN CV']['cascade classifier path'])
     cap = cv2.VideoCapture(0)
     while True:
         pixmap = QPixmap(f'pictures/{counter}.png')
         window.label_2.setFixmap(pixmap)
window.label_2.resize(pixmap.width(), pixmap.height())
         ret, frame = cap.read()
          gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
         s = time.time()
              s = time.time()
emotion, confidence = inference(sess, face_img_gray)
logger.critical('model inference time: {}'.format(time.time() - s))
with open(f'{name}.csv', 'a') as n:
    n.write(f"{counter};{emotion};{confidence};{time.time()}\n")
               if int(time.time()) - start time > TIME TO CHANGE PICTURE:
                 start_time = int(time.time())
                    counter += 1
          if counter > 6:
     cap.release()
```

Рис. 3.7 Основной цикл from cam- обработка текущего изображения нейросетью.

Здесь программа с помощью захвата лица, пытается найти на изображении с веб-камеры мое лицо. После нахождения лица, программа переводит его в матрицу и сравнивает эту матрицу с паттернами, которые были предварительно обучены. Далее программа делает вывод о том, какая

эмоция соответствует матрице текущего лица. Работы цикла происходит до тех пор, пока камера захватывает лицо.

3.4. Процесс обучения нейросети

На данном этапе происходит обучение нейросети для создания моделей. Обучение нейросети производилось на моем лице, путем демонстрации в камеру соответствующих эмоций, для создания композитных лиц, на основе которых в дальнейшем при использовании программы, будет сравниваться лицо пользователя и создаваться определенный эмоциональный тип.

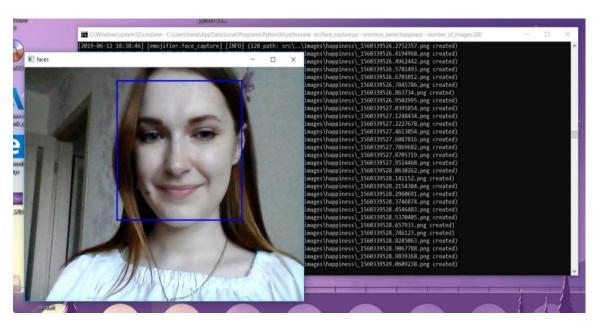


Рис. 3.8 Обучение нейросети на своем лице.

После того, как камера зафиксировала все эмоции, начинается процесс обучения. На данном этапе точность и общий процент обучения нейросети очень маленький.

```
C:\Windows\system32\cmd.exe - C:\Users\Irena\AppData\Local\Programs\Python36\python.exe src/trainer.py
                                                                                    | 100/5000 [00:37<28:58, 2.82it/s][2019-06-12 17:53:18] [emojifier.mode
        (accuracy: 23.33%, loss: 2.227)
                                                                                    | 120/5000 [00:44<29:09, 2.79it/s][2019-06-12 17:53:26] [emojifier.mode
                                                                                    | 140/5000 [00:52<29:26, 2.75it/s][2019-06-12 17:53:33] [emojifier.mode
        (accuracy: 33.33%, loss: 1.896)
                                                                                    | 160/5000 [00:59<28:28, 2.83it/s][2019-06-12 17:53:40] [emojifier.mode
                                                                                    | 180/5000 [01:06<28:27, 2.82it/s][2019-06-12 17:53:47] [emojifier.model
        (accuracy: 36.67%, loss: 1.438)
                                                                                    | 200/5000 [01:13<30:55, 2.59it/s][2019-06-12 17:53:54] [emojifier.mode
        (accuracy: 30.0%, loss: 1.673)
                                                                                    | 220/5000 [01:20<29:10, 2.73it/s][2019-06-12 17:54:02] [emojifier.mode
        (accuracy: 46.67%, loss: 1.399)
                                                                                    | 240/5000 [01:28<28:39, 2.77it/s][2019-06-12 17:54:09] [emojifier.mode
         (accuracy: 50.0%, loss: 1.428)
                                                                                    260/5000 [01:35<28:11, 2.80it/s][2019-06-12 17:54:16] [emojifier.mode]
                                                                                    280/5000 [01:43<32:02, 2.46it/s][2019-06-12 17:54:25] [emojifier.model
        (accuracy: 50.0%, loss: 1.349)
                                                                                    | 300/5000 [01:52<40:55, 1.91it/s][2019-06-12 17:54:34] [emojifier.mode
    FO] (accuracy: 56.67%, loss: 1.339)
                                                                                    | 320/5000 [02:03<34:22, 2.27it/s][2019-06-12 17:54:45] [emojifier.mode
                                                                                    | 340/5000 [02:10<28:04, 2.77it/s][2019-06-12 17:54:52] [emojifier.mode
    FO] (accuracy: 66.67%, loss: 0.9299)
                                                                                    | 360/5000 [02:18<27:19, 2.83it/s][2019-06-12 17:54:59] [emojifier.mode
        (accuracy: 66.67%, loss: 0.9582)
```

Рис. 3.9 Начало обучения.

На следующем этапе нейросеть уже полностью обучена, и далее запускается основной интерфейс программы.

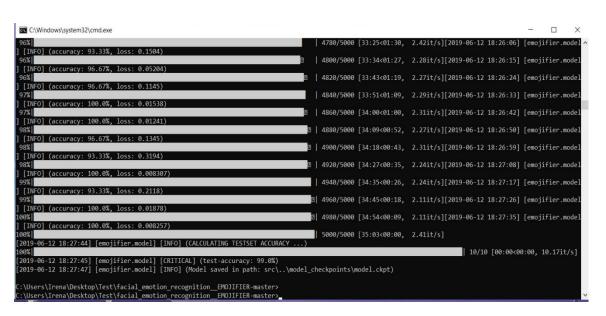


Рис. 3.10 Конец обучения.

3.5. Интерфейс программы

Интерфейс программы достаточной простой. В левом нижнем углу есть окошко, куда записывается имя испытуемого. После нажатия кнопки start начинается демонстрация картинок, которые меняются с интервалом в 2 секунды. Картинки демонстрируют 6 основных эмоций: радость, грусть, спокойствие, удивление, злость и презрение. Если необходимо продолжить демонстрацию, снова нажимается кнопка старт для повторного показа изображений.



Рис. 3.11 Интерфейс программы Detector.

По завершению демонстрации результаты работы программы записываются в Exel файл, который называется именем испытуемого. В файле в несколько колонок записываются основные показатели, полученные в процессе теста:

- 1. Номер картинки
- 2. Эмоция
- 3. Вероятность, с которой она получена

4. Текущее время (значения в ней нужны скорее для внутреннего взаимодействия)

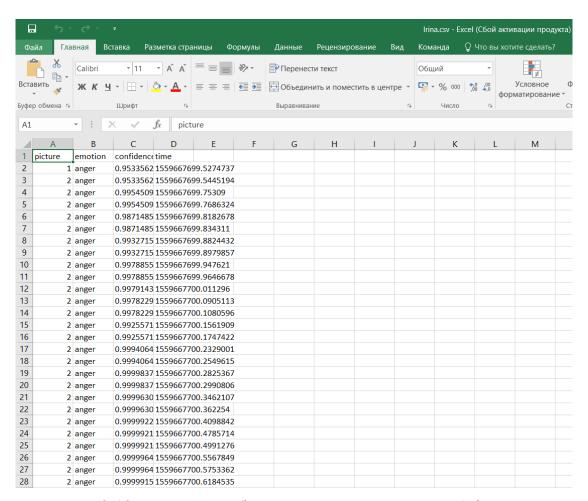


Рис. 3.12 Результат работы программы в виде Exel-файла.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы были проанализированы существующие способы рассмотрены И решения поставленной проблемы, реализован алгоритм распознавания эмоций и разработан программный продукт, который позволит ученым психологам в последующем проводить тестирование ДЛЯ оценки уровня психоэмоционального состояния детей и поможет выбирать стратегии коррекции отклонений в развитии.

Таким образом, в процессе работы были достигнуты поставленные задачи и цели:

- выявлены основные критерии оценивания на основе проведенного анализа проблемы
 - выбрана методика тестирования
- выбран механизм реализации и программные и аппаратные средства
 - реализован и протестирован программный комплекс

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

- 1. Нейроинтерфейс [Электронный ресурс]. Режим доступа: http://www.eye-tracking.ru/neurointerface/, свободный. Загл. С экрана.
- 2. Система контроля психоэмоционального состояния человека VibraImage, ТЕХНИЧЕСКОЕ ОПИСАНИЕ Версия 10.1, Многопрофильное Предприятие «ЭЛСИС», Санкт – Петербург, 2017.
- 3. Emotion Recognition using Wireless Signals. Mingmin Zhao, Fadel Adib, Dina Katabi / Massachusetts Institute of Technology // 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM. ISBN 978-1-4503-4226-1/16/10
- 4. EQ-Radio беспроводным способом реагирует на эмоции (+видео) [Электронный ресурс]. Режим доступа: http://www.robogeek.ru/nauchnye-razrabotki-programmnoe obespechenie/eq-radio-besprovodnym-sposobom-reagiruet-na-emotsii. свободный. Загл. С экрана.
- 5. Распознавание эмоций по выражению лица [Электронный ресурс]. Режим доступа: http://neuro-brand.ru/emodetect/, свободный. Загл. С экрана.
- 6. FaceReaderTM Tool for automatic analysis of facial expressions Reference Manual Version 7/Leanne Loijens, Olga Krips, Fabrizio Grieco, Hans van Kuilenburg, Marten den Uyl, Paul Ivan//2016 Noldus Information Technology b.v.
- 7. Cognitive Services Распознавание лиц и эмоций в Xamarin.Forms с помощью Microsoft Cognitive Services [Электронный ресурс]. Режим доступа: https://msdn.microsoft.com/ru-ru/magazine/mt742868.aspx, свободный. Загл. С экрана.
- 8. FaceReaderTM Tool for automatic analysis of facial expressions Reference Manual Version 7/Leanne Loijens, Olga Krips, Fabrizio Grieco, Hans van Kuilenburg, Marten den Uyl, Paul Ivan//2016 Noldus Information Technology b.v.

- Facial Expression Recognition Using Principal Component Analysis / Ajit P.
 Gosavi, S. R. Khot// International Journal of Soft Computing and Engineering
 (IJSCE) ISSN: 2231-2307, Volume-3, Issue-4, September 2013.
- 10. Анализ существующих подходов к распознаванию лиц [Электронный ресурс]. Режим доступа: https://habr.com/ru/company/synesis/blog/238129/, свободный.- Загл. С экрана.
- 11. Face Recognition Using Principal Component Analysis/ Ningthoujam Sunita Devi1, K. Hemachandran2// national Journal of Computer Science and Information Technologies, Vol. 5 (5), 2014, 6491-6496.
- 12. Face Recognition using Principle Component Analysis /Kyungnam Kim//
 Department of Computer Science University of Maryland, College Park MD
 20742, USA.
- 13. Face Recognition Techniques: A Review Rajeshwar Dass, Ritu Rani, Dharmender Kumar/ Volume 4, ctp.76// Issue 7 (November 2012).
- 14. Анализ существующих подходов к распознаванию лиц [Электронный ресурс]. Режим доступа: https://habr.com/ru/company/synesis/blog/238129/, свободный.- Загл. С экрана.
- 15. Face Recognition Techniques: A Review Rajeshwar Dass, Ritu Rani, Dharmender Kumar/ Volume 4, Issue 7 (November 2012).
- 16. Advances in Multimedia Information Processing/ PCM 2001: Second IEEE/ ctp.256 //Pacific Rim Conference on Multimedia Beijing, China, October 24–26, 2001 Proceedings.
- 17. Statistical Models of Appearance for Computer Vision, T.F. Cootes and C.J.Taylor/ Imaging Science and Biomedical Engineering, University of Manchester, Manchester M13 9PT, U.K// March 8, 2004.
- 18.Выравнивание лица, Вычисление и сравнение признаков [Электронный ресурс]. Режим доступа: https://studbooks.net/1783679/informatika/vyravnivanie_litsa, свободный.-Загл. С экрана.

- 19. Robust Face Recognition using Key-point Descriptors /Soeren Klemm, Yasmina Andreu, Pedro Henriquez and Bogdan J. Matuszewski//University of Central Lancashire, Preston, U.K., 2015.
- 20.Обнаружение устойчивых признаков изображения: метод SURF [Электронный ресурс]. Режим доступа: http://twinpeppers.blogspot.com/2012/03/surf-httphabrahabrrupost103107.html, свободный.- Загл. С экрана.
- 21. Wavelet Transform Based Face Recognition Using SURF Descriptors/Musa M. Ameen, Alaa Eleyan and Gülden Eleyan//Konya, Turkey, g Vol. 5, No. 1, February 2017.
- 22. Histogram of Oriented Gradients for Detection of Multiple Scene Properties / MaesenChurchill //Adela Fedor.
- 23. Обзор методов распознавания изображений [Электронный ресурс]. Режим доступа: https://moluch.ru/conf/tech/archive/300/14199/, свободный.- Загл. С экрана.
- 24.Peter Burkert, Felix Trier, Muhammad Zeshan Afzal. DeXpression: Deep Convolutional Neural Network for Expression Recognition [Электронный ресурс]. Режим доступа: https://pdfs.semanticscholar.org/8cea/404e8a5c4c11064923e5a6c023a0ae594a5 a.pdf, свободный.- Загл. С экрана.
- 25. Буй Тхи Тху Чанг, Фан Нгок Хоанг, В.Г. Спицын. Распознавание лиц на основе применения метода виолы—джонса, Вейвлет-преобразования и метода главных компонент признаков [Электронный ресурс]. Режим доступа: http://earchive.tpu.ru/bitstream/11683/4244/1/bulletin_tpu-2012-320-5-10.pdf, свободный.- Загл. С экрана.
- 26.Ole Helvig Jensen. Implementing the Viola-Jones Face Detection Algorithm Recognition [Электронный ресурс]. Режим доступа: https://pdfs.semanticscholar.org/40b1/0e330a5511a6a45f42c8b86da222504c71 7f.pdf, свободный. Загл. С экрана.

- 27. К. И. Ананьева, И. А. Басюл, В. Б. Дорохов, А. Н. Гусев, О. А. Королькова, А. А. Обознов, Е. В. Соловьева, Л. И. Сурат, Е. Г. Хозе, Ю. Е. Шелепин. Айтрекинг в психологической науке и практике. Москва 2016- 410с.
- 28. Эргономика рабочего места [Электронный ресурс]. Режим доступа: http://www.up-pro.ru/encyclopedia/ergonomika-rabochego-mesta.html, свободный.- Загл. С экрана.
- 29.Radboud Faces Database [Электронный ресурс]. Режим доступа: http://www.socsci.ru.nl:8180/RaFD2/RaFD, свободный.- Загл. С экрана.
- 30.L. V. Tulchak, A. O. Marchuk. History of Python [Электронный ресурс]. Режим доступа: https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/10471/461.pdf?sequence

=3, свободный.- Загл. С экрана.

31.С.В. Томилов. Распознавание лиц на базе библиотеки OpenCV [Электронный ресурс]. – Режим доступа: https://dspace.susu.ru/xmlui/bitstream/handle/0001.74/1504/45.pdf?sequence=1 ?sequence=1, свободный.- Загл. С экрана.

ПРИЛОЖЕНИЕ 1

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import os
import sys
import cv2
import time
import logging
import json
import tensorflow as tf
import numpy as np
import glob
import tqdm
import design
import traceback
from PyQt5 import QtWidgets, QtGui
from PyQt5.QtCore import QThread
from PyQt5.QtWidgets import QFileDialog
from PyQt5.QtGui import QPixmap
sys.path.append(os.path.join(os.path.dirname(__file__), os.pardir))
from src.trainer import model
from src.__init__ import *
TIME_TO_CHANGE_PICTURE = 2
config = tf.ConfigProto()
config.gpu_options.allow_growth = True
class GUIApp(QtWidgets.QMainWindow, design.Ui_MainWindow):
    def __init__(self):
        super().__init__()
        self.setupUi(self)
        self.commandLinkButton.clicked.connect(start)
def start():
    thread.start()
class AThread(QThread):
```

```
def run(self):
        def softmax(x):
            e_x = np.exp(x - np.max(x))
            return e_x / e_x.sum()
        def inference(sess, gray img input):
            img = gray_img_input.reshape(1, 48, 48, 1).astype(float) / 255
            y_c = sess.run(y_conv, feed_dict={X: img, keep_prob: 1.0})
            y_c = softmax(y_c)
            p = np.argmax(y_c, axis=1)
            score = np.max(y_c)
            # logger.debug('''
                  softmax-out: {},
                  predicted-index: {},
                  predicted-emotion: {},
                  confidence: {}'''.format(y_c, p[0], index_emo[p[0]], score))
            return index emo[p[0]], score
        def from_cam(sess):
            start_time = int(time.time())
            counter = 1
            face_cascade =
cv2.CascadeClassifier(config_parser['OPEN_CV']['cascade_classifier_path'])
            cap = cv2.VideoCapture(0)
            while True:
                pixmap = QPixmap(f'pictures/{counter}.png')
                window.label_2.setPixmap(pixmap)
                window.label_2.resize(pixmap.width(), pixmap.height())
                ret, frame = cap.read()
                gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
                faces = face_cascade.detectMultiScale(gray, 1.3, 5)
                for (x, y, w, h) in faces:
                    cv2.rectangle(gray, (x, y), (x + w, y + h), (255, 0, 0), 2)
                    face_img_gray = gray[y:y + h, x:x + w]
                    face_img_gray = cv2.resize(face_img_gray, (48, 48))
                    s = time.time()
                    emotion, confidence = inference(sess, face_img_gray)
                    logger.critical('model inference time: {}'.format(time.time() - s))
                    with open(f'{name}.csv', 'a') as n:
                        n.write(f"{counter};{emotion};{confidence};{time.time()}\n")
                    if int(time.time()) - start_time > TIME_TO_CHANGE_PICTURE:
                        start_time = int(time.time())
```

```
counter += 1
                if counter > 6:
                    break
            cap.release()
        try:
            name = window.lineEdit.text()
            with open(f'{name}.csv', 'a') as f:
                f.write(f"picture;emotion;confidence;time\n")
            logger = logging.getLogger('emojifier.predictor')
            CHECKPOINT_SAVE_PATH = os.path.join(os.path.dirname(__file__), os.pardir,
'model_checkpoints')
            EMOJI_FILE_PATH = os.path.join(os.path.dirname(__file__), os.pardir, 'emoji')
            tf.reset_default_graph()
            # used to map the output from the prediction to the emotion class
            index_emo = {v: k for k, v in EMOTION_MAP.items()}
            # dictionary of emoji name and the corresponding read image
            emoji_to_pic = {k: None for k in EMOTION_MAP.keys()}
            emoji_png_files_path = os.path.join(EMOJI_FILE_PATH, '*.png')
            files = glob.glob(emoji_png_files_path)
            logger.info('loading the emoji png files in memory ...')
            import platform
            if platform.system() == 'Windows':
                split_string = '\\'
            else:
                split_string = '/'
            for file in tqdm.tqdm(files):
                logger.debug('file path: {}'.format(file))
                emoji_to_pic[file.split(split_string)[-1].split('.')[0]] =
cv2.imread(file, -1)
            X = tf.placeholder(
                tf.float32, shape=[None, 48, 48, 1]
            keep_prob = tf.placeholder(tf.float32)
            y_conv = model(X, keep_prob)
```

```
saver = tf.train.Saver()
    with tf.Session(config=config) as sess:
        saver.restore(sess, os.path.join(CHECKPOINT_SAVE_PATH, 'model.ckpt'))
        from_cam(sess)
        window.label_2.setText('END')
    except Exception as e:
        pass

if __name__ == "__main__":
    app = QtWidgets.QApplication(sys.argv)
    thread = AThread()
    window = GUIApp()
    window.show()
    app.exec_()
```

ПРИЛОЖЕНИЕ 2

```
import os
import sys
import cv2
import time
import logging
import json
import tensorflow as tf
import numpy as np
import glob
import tqdm
sys.path.append(os.path.join(os.path.dirname(__file__), os.pardir))
import src
from src.data_manager import EmojifierDataManager
from src.__init__ import *
config = tf.ConfigProto()
config.gpu_options.allow_growth = True
logger = logging.getLogger('emojifier.model')
def weight_variable(shape):
    initial = tf.truncated_normal(
        shape=shape, stddev=0.1
    )
    return tf.Variable(initial)
def bias_variable(shape):
    initial = tf.constant(0.1, shape=shape)
    return tf.Variable(initial)
def conv2d(x, W):
    return tf.nn.conv2d(
        x, W, strides=[1,1,1,1], padding='SAME'
    )
```

```
def max_pool_2x2(x):
    return tf.nn.max_pool(
        x, ksize=[1,2,2,1], strides=[1,2,2,1], padding='SAME'
    )
def conv layer(input, shape):
    W = weight_variable(shape)
    b = bias_variable([shape[3]])
    return tf.nn.relu(tf.layers.batch_normalization(conv2d(input, W) + b))
def full_layer(input, size):
    insize = int(input.get_shape()[1])
    W = weight_variable([insize, size])
    b = bias_variable([size])
    return tf.matmul(input, W) + b
def model(x, keep prob):
    C1, C2, C3 = 30, 50, 80
    F1 = 512
    conv1_1 = conv_layer(x, shape=[3, 3, 1, C1])
    conv1_1_pool = max_pool_2x2(conv1_1)
    conv1_2 = conv_layer(conv1_1_pool, shape=[3, 3, C1, C2])
    conv1_2_pool = max_pool_2x2(conv1_2)
    conv1_drop = tf.nn.dropout(conv1_2_pool, keep_prob=keep_prob)
    conv2_1 = conv_layer(conv1_drop, shape=[3, 3, C2, C3])
    conv2_1_pool = max_pool_2x2(conv2_1)
    conv2_flat = tf.reshape(conv2_1_pool, [-1, 6*6*C3])
    conv2_drop = tf.nn.dropout(conv2_flat, keep_prob=keep_prob)
    full1 = tf.nn.relu(full_layer(conv2_drop, F1))
    full1_drop = tf.nn.dropout(full1, keep_prob=keep_prob)
   y_conv = full_layer(full1_drop, len(EMOTION_MAP))
```

```
return y_conv
```

```
def test(emoji_data, sess):
    logger.info('CALCULATING TESTSET ACCURACY ...')
    L = len(emoji_data.test.labels)
    x = emoji_data.test.images
    y = emoji_data.test.labels
    accs = []
    for i in tqdm.tqdm(range(0, L, 30)):
        if i+30 <= L:
            x_i = x[i:i+30].reshape(30, 48, 48, 1)
            y_i = y[i:i+30].reshape(30, len(EMOTION_MAP))
        else:
            x_i = x[i:].reshape(L-i, 48, 48, 1)
            y_i = y[i:].reshape(L-i, len(EMOTION_MAP))
        accs.append(sess.run(accuracy, feed_dict={X:x_i, Y:y_i, keep_prob:1.0}))
    acc = np.mean(accs)
    logger.critical('test-accuracy: {:.4}%'.format(acc*100))
if __name__ == '__main__':
    CHECKPOINT_SAVE_PATH = os.path.join(os.path.dirname(__file__), os.pardir,
'model_checkpoints')
    if not os.path.exists(CHECKPOINT_SAVE_PATH):
        os.makedirs(CHECKPOINT_SAVE_PATH)
    BATCH_SIZE = config_parser.getint('MODEL_HYPER_PARAMETERS', 'batch_size')
    STEPS = config_parser.getint('MODEL_HYPER_PARAMETERS', 'train_steps')
    LEARNING_RATE = config_parser.getfloat('MODEL_HYPER_PARAMETERS', 'learning_rate')
    KEEP_PROB = config_parser.getfloat('MODEL_HYPER_PARAMETERS', 'dropout_keep_prob')
    X = tf.placeholder(
        tf.float32, shape=[None, 48, 48, 1]
```

```
)
Y = tf.placeholder(tf.float32, shape=[None, len(EMOTION MAP)])
keep_prob = tf.placeholder(tf.float32)
emoset = EmojifierDataManager()
logger.info("Number of train images: {}".format(
    len(emoset.train.images)
))
logger.info("Number of train labels: {}".format(
    len(emoset.train.labels)
))
logger.info("Number of test images: {}".format(
    len(emoset.test.images)
))
logger.info("Number of test labels: {}".format(
    len(emoset.test.labels)
))
y_conv = model(X, keep_prob)
cross_entropy = tf.reduce_mean(
    tf.nn.softmax_cross_entropy_with_logits(
        labels=Y,
        logits=y_conv
    )
)
train = tf.train.AdamOptimizer(LEARNING_RATE).minimize(cross_entropy)
correct_predictions = tf.equal(
    tf.argmax(y_conv, 1), tf.argmax(Y, 1)
)
accuracy = tf.reduce_mean(
    tf.cast(correct_predictions, tf.float32)
)
saver = tf.train.Saver()
with tf.Session(config=config) as sess:
```

```
sess.run(tf.global_variables_initializer())

for i in tqdm.tqdm(range(STEPS)):
    x_data, y_data = emoset.train.next_batch(BATCH_SIZE)

acc, loss, _ = sess.run(
        [accuracy, cross_entropy, train],
        feed_dict={X:x_data, Y:y_data, keep_prob: KEEP_PROB}
)

if i % 20 == 0:
    logger.info('accuracy: {:.4}%, loss: {:.4}'.format(
        acc*100, loss
    ))

test(emoset, sess)

save_path = saver.save(sess, os.path.join(CHECKPOINT_SAVE_PATH, 'model.ckpt'))

logger.info("Model saved in path: {}".format(save_path))
```