Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к магистерской диссертации

«Автоматическое распознавание слов из ограниченного словаря на основе последовательности изображений с видео»

Автор: Ткаченко Григорий Станиславович _____
Направление подготовки (специальность): 01.04.02 Прикладная математика и информатика
Квалификация: Магистр
Руководитель: Фильченков А.А., кандидат ф.-м. наук _____

К защите допустить
Зав. кафедрой Васильев В.Н., докт. техн. наук, проф. _____

«___» _____ 20___ г.

Санкт-Петербург, 2017 г.

Студент Ткаченко Г.С. **Группа** М4239 **Кафедра** компьютерных технологий **Факультет** информационных технологий и программирования

Направленность (профиль), специализация Технологии проектирования и разработки программного обеспечения

Квалификационная работа выполнена с оценкой _____

Дата защиты «___» _____ 20___ г.

Секретарь ГЭК *Павлова О.Н.* Принято: «___» _____ 20___ г.

Листов хранения _____

Демонстрационных материалов/Чертежей хранения _____

Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

УТВЕРЖДАЮ
Зав. каф. компьютерных технологий
докт. техн. наук, проф.
_____ Васильев В.Н.
«___» _____ 20___ г.

ЗАДАНИЕ
НА МАГИСТЕРСКУЮ ДИССЕРТАЦИЮ

Студент Ткаченко Г.С. **Группа** М4239 **Кафедра** компьютерных технологий **Факультет** информационных технологий и программирования **Руководитель** Фильченков Андрей Александрович, кандидат ф.-м. наук, доцент кафедры КТ Университета ИТМО

1 Наименование темы: Автоматическое распознавание слов из ограниченного словаря на основе последовательности изображений с видео

Направление подготовки (специальность): 01.04.02 Прикладная математика и информатика

Направленность (профиль): Технологии проектирования и разработки программного обеспечения

Квалификация: Магистр

2 Срок сдачи студентом законченной работы: « » мая 2017 г.

3 Техническое задание и исходные данные к работе.

В рамках данной работы требовалось разработать новый подход для распознавания речи из слов ограниченного словаря исключительно на основе визуальных признаков. Важно подчеркнуть, что полученная модель должна работать на произвольном спикере без предварительной подстройки под пользователя.

4 Содержание магистерской диссертации (перечень подлежащих разработке вопросов)

а) Реализация инструмента для препроцессина данных для обучения (обработка видеоряда, выделение ключевых точек лица человека и получение единиц распознавания для каждого видео);

б) Проектирование и реализация алгоритма выделения меток слов из исходного словаря по видео-ряду;

в) Тестирование алгоритма и сравнение с существующими подходами.

5 Перечень графического материала (с указанием обязательного материала)

Не предусмотрено

6 Исходные материалы и пособия

а) Benedikt, Lanthao. «Facial Motion: a novel biometric?.» (2010);
б) Открытая библиотека алгоритмов компьютерного зрения http://dlib.net;
в) John Garofolo Lori Lamel William Fisher Jonathan Fiscus David Pallett Nancy Dahlgren Victor Zue. TIMIT Acoustic-Phonetic Continuous Speech Corpus.

7 Календарный план

№№ пп.	Наименование этапов магистерской диссертации	Срок выполнения этапов работы	Отметка о выполнении, подпись руков.
1	Изучение существующих подходов и моделей распознавания речи	10.2015	
2	Анализ существующих датасетов и поиск новых данных для обучения	12.2015	
3	Изучение алгоритмов выделения ключевых точек на лице человека	02.2016	
4	Разработка инфраструктуры для обработки данных	05.2016	
5	Разработка модели получения распределения вероятности визем на каждом фрейме	09.2016	
6	Разработка модели получения меток слов	11.2016	
7	Тестирование модели на большом числе спикеров	12.2016	
8	Написание пояснительной записки	03.2017	
9	Доработка работы и написание пояснительной записки	05.2017	

8 Дата выдачи задания: «01» сентября 2015 г.

Руководитель _____

Задание принял к исполнению _____ «01» сентября 2015 г.

Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

АННОТАЦИЯ
МАГИСТЕРСКОЙ ДИССЕРТАЦИИ

Студент: Ткаченко Григорий Станиславович

Наименование темы работы: Автоматическое распознавание слов из ограниченного словаря на основе последовательности изображений с видео

Наименование организации, где выполнена работа: Университет ИТМО

ХАРАКТЕРИСТИКА МАГИСТЕРСКОЙ ДИССЕРТАЦИИ

1 Цель исследования: Разработка алгоритма распознавания речи из слов ограниченного словаря исключительно на основе визуальных признаков. Важно подчеркнуть, что полученная модель должна работать на произвольном спикере без предварительной подстройки под пользователя.

2 Задачи, решаемые в работе:

а) Реализация инструмента для препроцессина данных для обучения (обработка видеоряда, выделение ключевых точек лица человека и получение единиц распознавания для каждого видео);

б) Проектирование и реализация алгоритма выделения меток слов из исходного словаря по видео-ряду;

в) Тестирование алгоритма на большом числе спикеров и сравнение с существующими подходами.

3 Число источников, использованных при составлении обзора: 20

4 Полное число источников, использованных в работе: 33

5 В том числе источников по годам

Отечественных			Иностранных		
Последние 5 лет	От 5 до 10 лет	Более 10 лет	Последние 5 лет	От 5 до 10 лет	Более 10 лет
1	1	2	12	6	5

6 Использование информационных ресурсов Internet: 6

7 Использование современных пакетов компьютерных программ и технологий: Обработка видео-данных происходила на языке C++. Вычисление вектора признаков производилось с помощью библиотек dlib и OpenCV. Для обработки аудио-данных и извлечения из них признаков применялся Python. Обучение нейронных сетей происходило на основе библиотеки Torch. Данные хранились преимущественно в hdf5. Кроме того, многие процессы обучения происходили на видеокарте с использованием CUDA. Также для некоторых вспомогательных

скриптов для обработки результатов тестирования человеком, для загрузки видео с YouTube и их обработки были использованы Python и Bash.

8 Краткая характеристика полученных результатов: Полученная модель показала лучшие результаты по сравнению с классическими подходами в этой области на основе проведения ряда экспериментов как на синтетических, так и на реальных данных.

9 Гранты, полученные при выполнении работы: В рамках данной работы была подача на грант КНВШ в 2016 году.

10 Наличие публикаций и выступлений на конференциях по теме работы:

1 *Ткаченко Г.*, *Фильченков А.* Автоматическое распознавание слов из ограниченного словаря на основе визуальных признаков // Всероссийская научная конференция по проблемам информатики СПИСОК. — 2017.

Выпусник: Ткаченко Г.С. _____

Руководитель: Фильченков А.А. _____

«__» _____ 20__ г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

В данное время большое внимание уделяется задаче распознавания речи. Уже трудно представить многие популярные продукты, такие как мобильные ассистенты, навигаторы или развлекательные сервисы без голосового ввода. Но важно отметить, что часто под распознаванием речи понимается преобразование записи голоса человека в текстовые данные. Однако, в некоторых случаях (например, в зашумленных условиях, таких как в метро) одной звуковой информации бывает недостаточно. А в некоторых ситуациях наоборот она сильно зашумляет сигнал и мешает распознаванию. И в этот момент на помощь приходит визуальная информация (например, с фронтальной камеры телефона). В моей работе я обратился именно к этой задаче – распознавание речи на основе исключительно визуальной информации.

Такие технологии распознавания речи на основе исключительно визуальной информации могут применены в множестве областей. Среди основных можно выделить следующие:

- Задача идентификации спикера. Задача состоит в определении говорящего по некоторому набору произнесенных им слов;
- Задача верификации спикера. Задача состоит в определении того, действительно ли в систему пытается зайти тот человек, кому принадлежит данный аккаунт;
- Использование системы для распознавания речи как интерфейс вода команд в систему. К примеру, важной частью работы многих мобильных ассистентов, таких как Siri [1] или Cortana [2], является выполнение некоторых команд (к примеру, для запуска мобильных приложений). А так как зачастую такие команды имеют достаточно ограниченный словарь, то имеет место применение таких моделей, как модели распознавания речи только на основе визуальной информации;
- Использование модели распознавания речи по визуальным признакам для улучшения модели распознавание речи по аудио и видео сигналу. А именно такие модели могут строиться на подмешивании данных с видео в данные с аудио [3], что потенциально может увеличить точность распознавания всей системы.

Поскольку задача в общем виде довольно сложная, то было принято решение начать с более простой задачи – распознавание речи из слов ограни-

ченного словаря. При этом полученная модель может быть использована на любых спикерах, включая даже тех, на которых не происходило непосредственно обучение модели (speaker-independent model), что в некоторых случаях позволит избежать подстройку под конкретного пользователя и упрощает последующее использование полученной модели.

ГЛАВА 1. ОБЗОР

В данное время существует множество подходов к распознаванию речи на основе визуальной информации. Какие-то из них работают на отдельно стоящих словах, в то время как более продвинутые работают на связной речи человека.

В рамках краткого обзора будут приведены основные подходы и алгоритмы по распознавании речи, использующие визуальные признаки. А именно будут разобраны исторически первые подходы, связанные с человеческим чтением по губам и автоматическими подходами.

1.1. Чтение по губам человеком

Чтение по губам не является современным изобретением. Первые опыты были осуществлены уже в 1500 году н.э. и, вероятно, еще до этого времени. Первым успешным учителем по чтению губ был испанский бенедиктинский монах, Пьетро Понсе, который умер в 1588 году. Обучение чтению губ впоследствии распространилось на другие страны.

В литературе описано несколько различных методов для чтения губ человека, таких как методы Мюллера-Валле, Кинзи и Йены. Метод Мюллера-Валле фокусируется на движении губ в контексте создания звуков, а метод Кинзи делит преподавание чтения губ на 3 уровня обучения, в зависимости от сложности (новички, промежуточные и продвинутые).

Хоть и зачастую можно увидеть только 50% речи или меньше, чтец по губам должен уметь догадываться до тех слов, которые он пропустил. Однако, вне зависимости от большого разнообразия известных подходов, все методы так или иначе зависят от движения губ, которые чтецу по губам необходимо увидеть.

Согласно недавнему исследованию, проведенному в Манчестерском университете, люди с проблемами слуха могли понять около 21% речи, но если они использовали слуховой аппарат или чтение по губам, они могли понять 64%. Если они используют как слуховые аппараты, так и чтение по губам, их понимание речи повышается до 90%.

В 2014 году был проведен эксперимент, чтобы приблизительно измерить способность человека к чтению по губам и понять объем информации, который можно увидеть из речи. В этом эксперименте использовались четыре видеопоследовательности, 2 мужчины и 2 женщины. На каждом видео было

записано 10 цифр, где цифры и их последовательности отличаются от одного видеоролика к другому, а аудиосигналы были удалены из всех четырех видео. Пятьдесят пять участников были приглашены, чтобы попытаться расшифровать то, что говорилось на каждом видео. Каждый участник имел возможность воспроизводить каждое видео до 3 раз, поэтому у участников было достаточно времени для определения цифры. На всех видеороликах произносили только цифры, $1,2,3,\ldots,9$, участники были проинформированы об этом, поэтому людям было гораздо проще читать по губам, чем это происходило бы в общем случае. Результаты эксперимента приведены в таблице ниже. В каждой строке таблицы представлена точность чтения по конкретным людям.

Таблица 1 – результаты чтения по губам человеком

Человек	Результат
Женщина 1	61%
Мужчина 1	50%
Мужчина 2	37%
Женщина 2	63%
Среднее	53%

Как видно из таблицы выше, некоторые видеоролики были более легкими для чтения, чем другие (61% и 63% для видеороликов 1 и 4 соответственно), в то время как у некоторых других видеороликов было понятно меньше информации для читателей губ, или спикеры по своей природе либо говорили быстрее, чем это обычно бывает, или не давали достаточной информации для считывателей губ. Мы можем заметить, что женщины дают больше информации для читателей. Этот факт, конечно, трудно обосновать, так как это лишь небольшой эксперимент с использованием небольшого количества видеороликов, поэтому слишком рано делать такие выводы с такими доказательствами. Однако самое важное, что этот эксперимент может выявить - это общая способность к чтению по губам человека, которая составляет 53%. Еще одна интересная вещь - разные люди также имеют разные способности воспринимать речь с использованием визуальных сигналов. В этом эксперименте лучший результат считывания губ составил 73%, а наихудший - 23%. Эти эксперименты иллюстрируют вариацию индивидуальных навыков чтения губ и вариацию индивидуальной способности создавать четкий читаемый визуальный сигнал,

что добавляет множество проблем создания работоспособной автоматической системы для чтения по губам.

1.2. Автоматическое распознавание речи

В противовес профессионалам, которые занимаются чтением по губам, существуют автоматические подходы, позволяющие делать то же самое.

При этом важно отметить, что многие такие алгоритмы могут в качестве входной информации использовать как один источник, так и несколько ([4]). В качестве таких источников могут быть жесты, движения губ, человеческая речь и так далее. В данной работе, как уже отмечено выше, упор сделан исключительно на распознавание речи на основе движения губ человека.

Подавляющее большинство таких алгоритмов можно разбить на два больших шага:

— Выделение признаков из фреймов видео, которые бы «описывали» текущий фрейм видео;
— Преобразование сигнала с изображения (фактически, вектора признаков изображения) в метки слов.

При этом первый пункт может уже достаточно широко рассмотрен в научной сфере и существует большое разнообразие подходов, которые получают получить требуемый вектор. Такие подходы могут брать во внимание как простые и понятные человеку геометрические признаки губ, так и более сложные модели с применением нейронных сетей.

Второй же пункт часто может быть рассмотрен под несколькими углами, такими как построение так называемых «Акустических моделей» (преобразования входного сигнала в распределение по единицам распознавания речи), «Языковых моделей» (выявление распределения вероятностей над всем множеством слов) и других.

Рассмотрим типичные подходы для каждого из этапов построения системы распознавания.

1.3. Методы выделения визуальных признаков

Первым большим шагом в построении системы распознавания речи является выделение признаков с изображения. В качестве таких признаков могут выступать как сырые данные (значения пикселей из региона интереса), так и всячески преобразованные координаты ключевых точек или геометрические

признаки, но неудивительно, что все такие признаки так или иначе связаны с поведением и положением губ в каждый момент времени. Кроме того, существуют подходы, которые используют методы трекинга лица для увеличения точности признаков в том случае, когда на вход подается не просто картинка, а видео-ряд (к примеру, [5]). Таким образом, такие подходы можно разделить на три большие группы, где алгоритм выделения признаков в своей основе полагается на

— Геометрические признаки. Такие признаки могут быть основаны на различных геометрических параметрах губ, таких как высота, ширина или различные отношения этих или других смежных величин;
— Признаки на основе значения пикселей в регионе интереса. Такие признаки представляют из себя преобразованные разными способами значения пикселей в области губ;
— Признаки на основе выделения контура губ. Такие же признаки сфокусированы на выделении некоторых контуров и ключевых точек и последующей обработки таких объектов для построения искомого вектора.

Рассмотрим подробно каждый класс алгоритмов и наиболее известные подходы в них по отдельности.

1.3.1. Алгоритме на основе геометрических признаков

Наиболее естественным и понятным человеку подходом для выделения признаков является попытка составить вектор легко интерпретируемых значений, основанных на геометрии губ.

Так в работе 2014 года [6] автор использует, помимо признаков на основе значений пикселей, в своем алгоритме такие признаки как высоту и ширину области рта. Также популярным методом является использование в качестве признаков отношение этих величин, что является естественной метрикой для человеческого глаза, поскольку мы хорошо замечаем положение губ, в котором они находятся при произнесении таких букв как «О» (в этом положении это отношение равно примерно единице, то есть высота и ширина примерно совпадают).

В некоторых работах, таких как [7], автор предлагает в качестве вектора признака использовать как геометрические признаки (отношение высоты эллипса области губ к его ширине), так и координаты точек контура в одном векторе, который впоследствии уменьшается методом главных компонент.

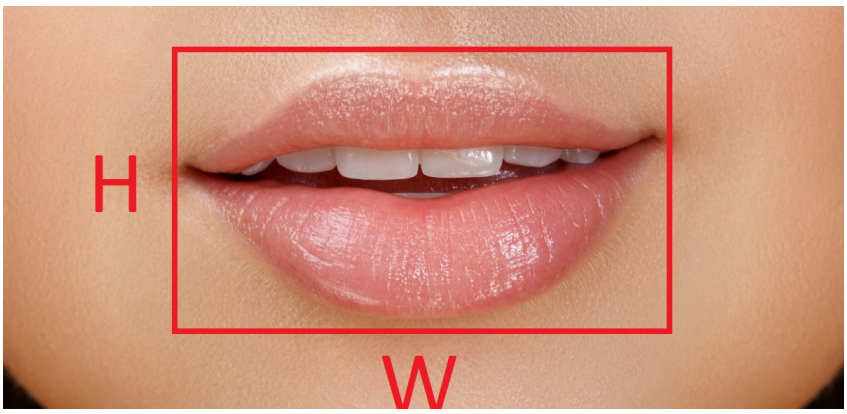

Рисунок 1 – Пример геометрических признаков (H - высота, W - ширина)

В целом использование таких подходов зачастую не является популярным решением прямо сейчас, потому что другие алгоритмы неявно извлекают ту же информацию, которую может придумать человек. А помимо этого они позволяют строить более сложные модели, которые практически невозможно придумать человеку. Достоинством же этого подхода является его высокая интерпретируемость.

1.3.2. Алгоритмы на основе анализа значений пикселей

Вторым естественным подходом является анализ непосредственно значений пикселей в регионе интереса.

Самым простым способом является попытка анализа непосредственно цвета пикселей и цветовая сегментация области губ. Так в работе [8] сперва находится потенциальная область губ с помощью линии глаз, а после методами оконтуривания и цветовой сегментации строятся контуры губ.

Так в статье [9] был предложен подход чуть более продвинутый алгоритм, который заключается в выделении нужного вектора с помощью метода главных компонент. В работе были использованы первые 10 собственных векторов для представления губ.

Другой известный подход называется гистограмма направленных градиентов (HOG, [10]). Данная техника основана на подсчете количества направлений градиента в локальных областях изображения. Такой подход позволяет достаточно точно «описывать» губы на текущем фрейме и используется во многих современных алгоритмах, таких как [11].

К другим продвинутым техникам этой группы можно отнести подходы, которые основаны на нейронных сетях специального вида - автокодировщик

(к примеру, [12]). В дальнейшем будем называть автокодировщик английским аналогом autoencoder. Основная идея такой сети состоит в использовании нейронной сети специального вида (схематически представлена на рисунке ниже). Такая структура позволяет получить некоторое компактное представление входных данных за счет попытки на выходе предсказать ровно те данные, которые пришли сети на вход. Соответсвенно размеры входного и выходного слоя сети должны совпадать, в то время как скрытый слой, компактно описывающий требуемый объект имеет меньшую размерность. В нашем случае с помощью такого подхода можно получить «описание» губ на этом скрытом слое нейронной сети. Важно отметить, что такое векторное описание губ не будет возможно легко интерпретировать человеком, но это вовсе не будет означать в его непригодности для алгоритма (а, скорее, наоборот).

Рисунок 2 – Схема autoencoder

1.3.3. Алгоритмы на основе выделения контура

К основным алгоритмам последней группы обычно относят те алгоритмы, идея которых состоит в попытке подгона некоторой статистической модели губ человека под изображение, которое поступает на вход алгоритма [13]. Особенно широко распространены две следующие модели:

— Active Appearence Models (AAM) [14];
— Active Shape Models (ASM) [15].

Помимо них на подобных идеях работают алгоритмы, которые находят ключевые точки на лице человека (к примеру, [16] и [17]). В нашем случае нас интересуют ключевые точки губ человека.

Кроме того, широко известен подход, который предложен в статье [17]. Такой метод опять же основан на подгоне статистической модели формы лица человека к входной картинке с использованием градиентного бустинга на регрессионных деревьях. Такой подход за счет специфики бустинга на регрессионных деревьев работает довольно быстро и показывает достойные результаты.

Ниже приведен пример работы этого алгоритма. На картинке лицу человека сопоставляется набор ключевых точек разных частей лица. Изображение ниже было получено с использованием открытой библиотеки машинного обучение dlib, где был эффективно реализован этот подход.

Рисунок 3 – Пример работы алгоритма из библиотеки dlib

Как будет продемонстрирован позднее данный подход и был выбран в качестве основного алгоритма нахождения ключевых, поскольку он имеет высокую точность, сопоставимую с платными решениями, высокую скорость работы и возможность его модификации за счет открытого исходного кода библиотеки dlib.

1.4. Акустические модели

В построении акустических моделей долгое время классическим подходом являлись скрытые марковские модели (например, [18, 19]). Основной идеей таких алгоритмов является эмуляция работы процесса, похожего на марковский процесс с неизвестными параметрами, и задачей ставится разгадывание неизвестных параметров (в нашем случае искомые единицы реи) на основе наблюдаемых. При этом из каждого состояния известны вероятности перехода в другие. Пример такой модели представлен на рисунке ниже.

В этой схеме каждая вершина обозначает отдельную часть речи (adj - прилагательное, noun - существительное, а det - артикль), в которой записываются пары (слово; вероятность, что слово относится именно к этой части речи). При этом переходы показывают возможную вероятность следования одной части речи за другой. Так, к примеру, вероятность того, что подряд будут идти 2 прилагательных, при условии, что встретится артикль, будет равна 0,218.

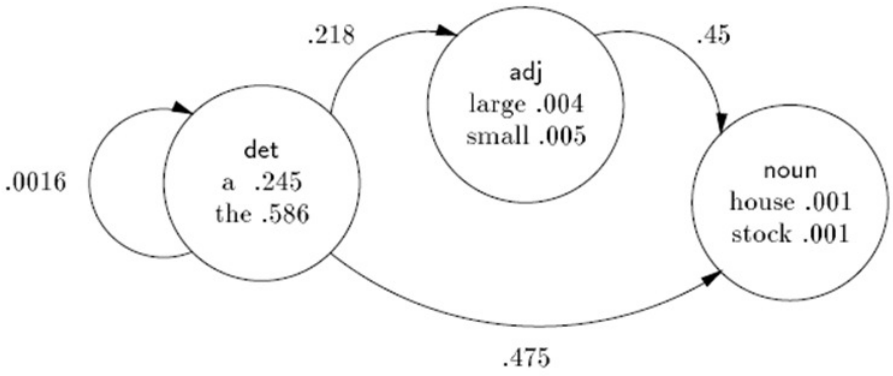

Рисунок 4 – Пример марковской модели

Однако, в настоящее время такие модели отходят на второй план, поскольку нейронные сети в задачах распознавания речи позволяют уловить значительно более сложные связи и показывают заметно более высокую точность. Кроме того, они являются, по сути, state-of-the-art подходом в автоматическом распознавании речи.

Нейронные сети позволяют предсказывать функции, зависящие от большого числа параметров. При этом они способны автоматически находить сложные зависимости без ручного вмешательства. Такая модель берет свое на-

чало от принципах работы человеческого мозга и является одним из наиболее популярных и эффективных моделей в машинном обучении в наши дни.

Классическая нейронная сеть представляет из себя последовательно соединенный между собой слои с весами на переходах между слоями. Однако, такие сети имеют фиксированного размера вход и выход, что не всегда является удобным в алгоритмах распознавании речи, потому что алгоритм должен генерировать метки слов, количество которых заранее неизвестно. Для решения этой задачи можно применять специальный вид нейронной сети - рекуррентная нейронная сеть.

Рекуррентные нейронные сети обладают «обратной связью», то есть такая сеть запоминает накопленную по ходу информацию и использует ее для принятия решения в текущий момент времени.

Однако и такие сети подвержены проблемам, таким как «проблема взрывающегося градиента» и «проблема изчезающего градиента». Это связано с тем, что последовательные умножения в ходе обучения могут давать как очень большое число, так и число, близкое к нулю. Для решения этих проблем была предложена рекуррентная нейронная сеть специального вида - LSTM.

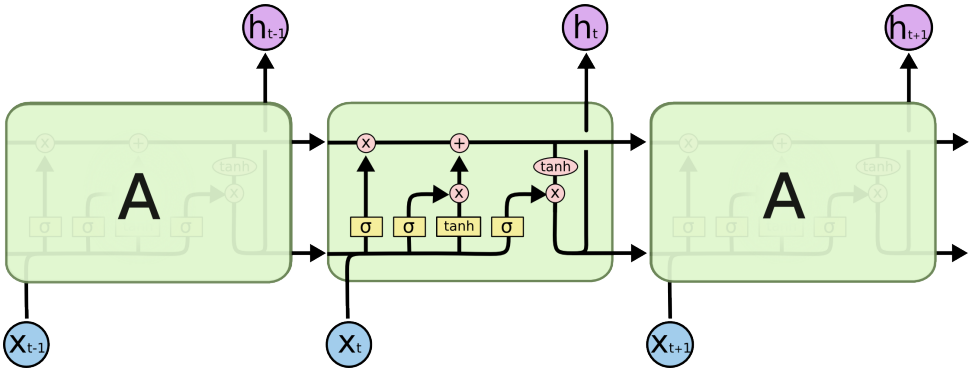

Рисунок 5 – Общая схема LSTM

Основной идеей в таких сетях является введение специального состояния, которому в определенный момент времени применяется сначала forget gate, а затем update gate. В таком подходе во-первых отлично сохраняются долгосрочные зависимости, а во-вторых проблемы, описанные выше, выражены гораздо менее остро за счет использования сигмоида и тангенса в качестве активационных функций.

В данной работе выбор был сделан в пользу нейронных сетей, потому что в последнее время они применяются в абсолютном большинстве совре-

менных систем для распознавания речи и показывают значительно лучшие результаты по сравнению с классическими подходами (скрытыми марковскими моделями).

1.5. Языковые модели

Языковые модели используются уже на более высоком уровне и обычно напрямую не работают с «сырыми» данными, такими как сигнал с аудио или видео дорожки. Однако такие модели обладают «знанием» про специфику используемого алфавита и в некоторых случаях способны «догадаться», опираясь на это знание, какое именно слово было произнесено.

В данной работе такие модели практически не рассматривались, однако важно отметить, что обычно для построения таких моделей современные системы применяют нейронные сети, а именно рекуррентные нейронные сети. Примеры применения такого подхода можно встретить во многих статьях, таких как [20] и [21].

1.6. Алгоритмы визуального распознавания речи

Как уже было сказано ранее, большинство алгоритмов распознавания строится на некотором извлечении признаков либо с аудио, либо с видео дорожки и дальнейшем преобразовании таких признаком в метки слов. Подобных алгоритмов довольно много, поэтому в этой части я рассмотрю только два типовых классических подхода, которые работают по похожей схеме.

1.6.1. Алгоритм на основе HOG и Eigenlips

Основная идея алгоритма на основе дескрипторов HOG и Eigenlips (описание которых подробно представлено выше) состоит в построении некоторого вектора, который описывает входную последовательность фреймов и использование метода Support Vector Machine (SVM) для определения метки слова.

Важно отметить, что поскольку SVM на вход требует один вектор фиксированной размерности, то для все полученные вектора с каждого фрейма видео конкатенируются в один вектор, а поскольку размерность его должна быть фиксирована, то такой вектор нормализуется до требуемой размерности. В дальнейшем классификатор возвращает метку слова.

1.6.2. Алгоритм на основе геометрических признаков

Алгоритм на основе геометрических признаков основан на использований «ручных» признаков (таких как высота, ширина и различные отношения).

В дальнейшем полученные вектора, как и в предыдущем алгоритме, конкатенируются, преборазуются до фиксированной размерности и после, в отличие от предыдущего алгоритма, подаются на вход классификатору k-nearest neighbours (KNN). Такой классификатор пытается найти максимально близкий вектор и в результате такого поиска возвращает метку слова, который соответствует найденному вектору.

Рисунок 6 – Пайплайн распознавания на основе геометричких признаков

Схематически работа такого алгоритма может быть проиллюстрирована схемой выше. Важным преимуществом такого подхода может служить то, что классификатор KNN является достаточно простым и не требует больших объемов данных для своего обучения.

1.6.3. Алгоритм на основе нейронной сети

Нельзя не отметить класс наиболее современных алгоритмов в области распознавания речи на основе визуальных признаков - алгоритмы с применений нейронный сетей.

В контексте рассматриваемой задачи разные структуры нейронных сетей могут быть использованы буквально на каждом этапе работы алгоритмы - начиная от выделения признаков с изображения и заканчивая моделью распознавания меток ключевых слов, поскольку такие алгоритмы делают меньше предположений о природе входных данных и позволяют решать требуемую задачу с максимальной точностью. Так в работе [22] нейронная сеть используется для классификации меток выходных слов. При этом стоит отметить, что более сложные структуры нейронных сетей (рекуррентные нейронные сети)

способны распознавать не просто одну метку слова, а предсказывать целый ряд меток за счет входа «динамического» размера.

ГЛАВА 2. ДАННЫЕ ДЛЯ ОБУЧЕНИЯ

Важным этапом в работе, безусловно, является поиск и выбор данных для обучения. В ходе исследований были реализованы алгоритмы поиска как новых данных, так и обработки существующих датасетов. А также был записан свой датасет на основе команд ассистента Siri ([23]).

2.1. Требования к данным

В качестве основных критериев для данных для обучения были выделены следующие особенности входных видеозаписей:

- В видео должна быть четкая видео- и аудио- дорожки;
- Разрешение видео должно быть не менее, чем 640 x 480;
- Спикер должен в течение всего видео сидеть в анфас;
- Произнесенное множество слов должно быть из небольшого словаря.

Для поиска подходящих данных было выбрано три направления

- Поиск данных на видео хостингах;
- Поиск готовых датасетов;
- Запись нового датасета.

2.2. Поиск видео на видео хостингах

В качестве формата видео, которое потенциально могло бы подойти по нашим требованиям, можно выделить две основные категории:

- Интервью;
- Новости.

В роли сервиса для получения таких данных был выбран популярный видео-хостинг YouTube. Алгоритм поиска видео представлял из себя программный вызов поискового API сервиса YouTube на ключевых словах, связанных с интервью и новостными видеороликами. Кроме того, был выбран список популярных профилей этого сервиса по вышеобозначенным тематикам и получены видеоролики с этих профилей.

Рисунок 7 – Пример подходящего видео с сервиса Youtube

По результатам отсмотра видео многие видеоролики подходили по большинству критериев, однако в подавляющем большинстве роликов используемый набор слов оказался слишком велик для того, чтобы использовать эти данные для обучения модели. Поэтому единственным возможным вариантом оставалось либо записывать свои данные, либо же использовать готовый датасет.

2.3. Поиск готовых датасетов

Среди уже готовых датасетов подавляющее большинство записано на английском языке. Среди этого множества наиболее яркими представителями являются следующие выборки:

- GRID [24],
  34 спикера, 1000 трехсекундных видео для каждого с произнесением слов из ограниченного словаря;
- AVLetters1 [25],
  10 спикеров, 3 повторения каждой буквы английского алфавита;
- LILiR TwoTalk corpus [26],
  4 диалога (2 спикера), 12 минут каждый;
- CUAVE [27],
  36 спикера, 10 цифр.

Среди описанных датасетов два из них содержат внушительное количество спикеров - это GRID и CUAVE. Однако важно заметить, что в последнем словарь состоит только из цифр, когда как в первом словарь гораздо насыщеннее и интереснее. Так как по всем остальным критериям оба датасета нам подходят, то в контексте выбора существующего датасета выбор был сделан в пользу датасета GRID.

2.4. Описание выбранного существующего датасета

Как было представлено в предыдущей секции для экспериментов был выбран датасет GRID в силу большого числа спикеров и подходящего объема словаря. Этот датасет состоит из записей речи 34 спикеров, где каждый из спикеров записал 1000 видео из фраз специального вида. При этом каждая такая фраза в общем видео может быть представлена следующим образом:

<команда:4><цвет:4><предлог:4>
<буква:25><цифра:10><наречие:4>

После двоеточия в примере выше приведено общее число различных слов соответствующего множества. Пример предложения такого вида: «lay green at A zero now». Более подробная структура предложений в записанных видео приведена в таблице ниже:

Таблица 2 – грамматика в GRID corpus

команда	цвет	предлог	буква	цифра	наречие
lay	red	at	A-Z	1-9	again
bin	green	with	без W	zero	soon
set	blue	in			please
place	white	by			now

Все видео в датасете имели как аудио, так и видео дорожку, при этом каждое видео имело достаточно высокое разрешение 720 на 576, что позволяло впоследствии достаточно точно находить ключевые точки губ на изображении. Также частота кадров была 25 кадров в секунду.

В датасете каждый спикер в течение всего видео смотрел строго в камеру. А также в датасете присутствовали как представители мужского пола, так и женского.

Кроме того, важно заметить, что для каждого видео известна его полная разметка, то есть количество, порядок и время произнесения каждого слова. Так как некоторые слова длились слишком маленькое количество кадров, то было решено оставить только те слова, которые длятся не менее, чем 4 кадра.

2.5. Запись нового датасета

Выбранный выше датасет удовлетворяет всем техническим параметрам и прекрасно подходит для исследовательской работы, однако в силу жесткой структуры каждой фразы он является довольно синтетическим и имеет малое отношение к реальному миру. В связи с этим было решено для тестирования модели записать датасет на подмножестве команд популярного голосового ассистента Siri (команды приведены в таблице ниже).

Таблица 3 – выбранные команды Siri

Disable Airplane mode	Disable Cellular
Disable Night mode	Disable Alarms
Disable Bluetooth	Disable Wi-Fi
Download Facebook	Download Telegram
Download WhatsApp	
Open Mail	Open Documents
Open Downloads	Open Movies
Open Music	Open Safari
Open Photos	Open Settings
Activate Airplane mode	Activate Cellular
Activate Night mode	Activate Alarms
Activate Bluetooth	Activate Wi-Fi
Show me Mail	Show me Documents
Show me Downloads	Show me Movies
Show me Music	Show me Safari
Show me Photos	Show me Settings

Запись датасета происходила в комнатных условиях в дневное время. При этом для записи каждого приглашенного спикера на расстоянии примерно полтора метра от камеры был поставлен стул, на котором в течение всей записи сидел человек. Камера находилась примерно на уровне головы спикера. В течение всей записи за спиной спикера находилась белая стена.

Запись каждого видео происходила с помощью камеры GoPro HERO 5 Black на штативе в разрешении 1080p (1920 на 1080) с частотой кадров 60 кадров в секунду.

Схематически процесс можно описать схемой ниже. Черная полоса слева изображения представляет из себя стену.

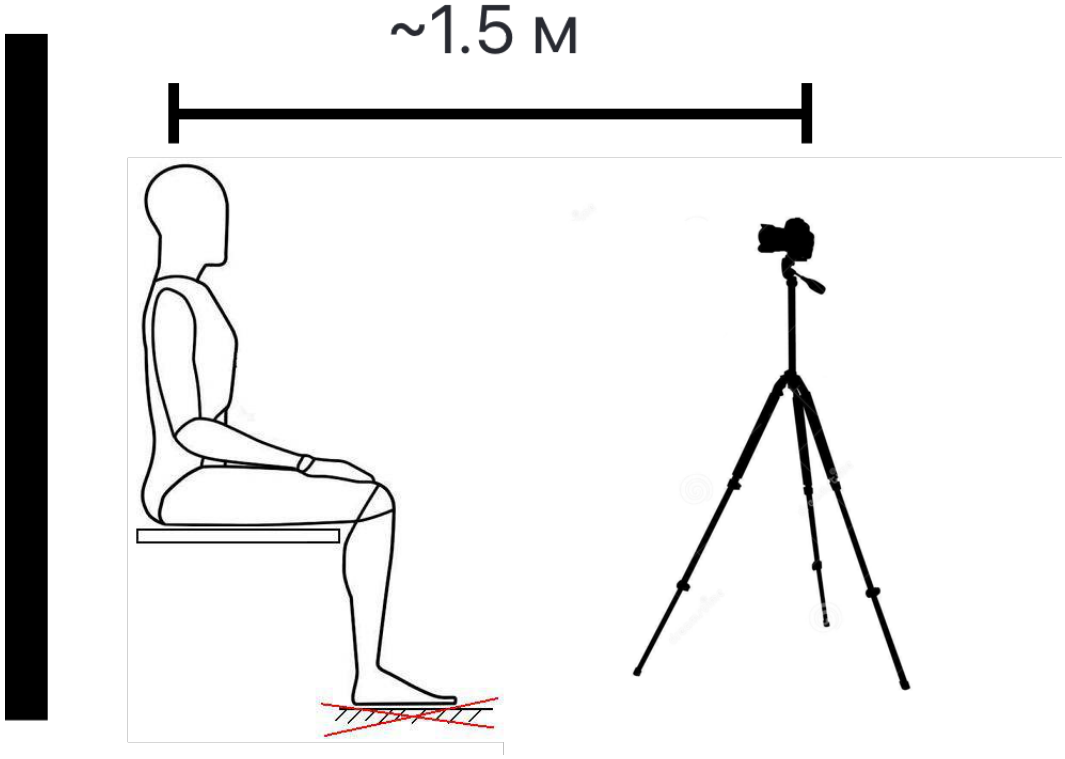

Рисунок 8 – Схема расположения камеры и человека в момент записи датасета

Опишем подробнее непосредственно полученные записи. Для датасета были выбраны 10 спикеров, каждый из которых записал по 15 раз каждую из фраз (то есть, к примеру, первый спикер произнес в камеру фразу «Open Mail» 15 раз за всю сессию записи). Таким образом, каждая фраза из таблицы выше была записана 150 раз. Для записи принимали участие представители как мужского, так и женского пола (среди них был 5 мужчин и, соответсвенно, 5 представительниц прекрасного пола). Также важно отметить, что составные

слова, такие как «Airplane mode» или «Night mode», считались одним словом в дальнейших экспериментах.

Пример фрагмента из полученного датасета представлен на рисунке ниже.

Рисунок 9 – Кадр из записанного датасета

ГЛАВА 3. ПРЕДЛОЖЕННЫЙ МЕТОД

3.1. Общий подход к построению системы распознавания речи

Большинство алгоритмов распознавания речи разработаны по похожей схеме. Предложенный в этой работе алгоритм тоже ее придерживается, поэтому ниже представлена типичная последовательность шагов в системе распознавания речи.

— Предобработка

  Включает в себя работу по обработке данных. А именно состоит из следующих шагов:

  – Чтение видео-данных и выделение фреймов из видео-ряда;
  – Выделение признаков по изображению (например, это могут быть обработанные вектора ключевых точек на лице человека);
  – Получение разметки по фонемам и виземам для каждого фрейма.

— Обучение модели предсказания распределения единиц визуального распознавания речи (визем) по входному сигналу;

— Получение меток искомых слов по распределениям единиц речи на каждом фрейме.

Схематически общий агоритм построения такой системы может представлен следующим образом:

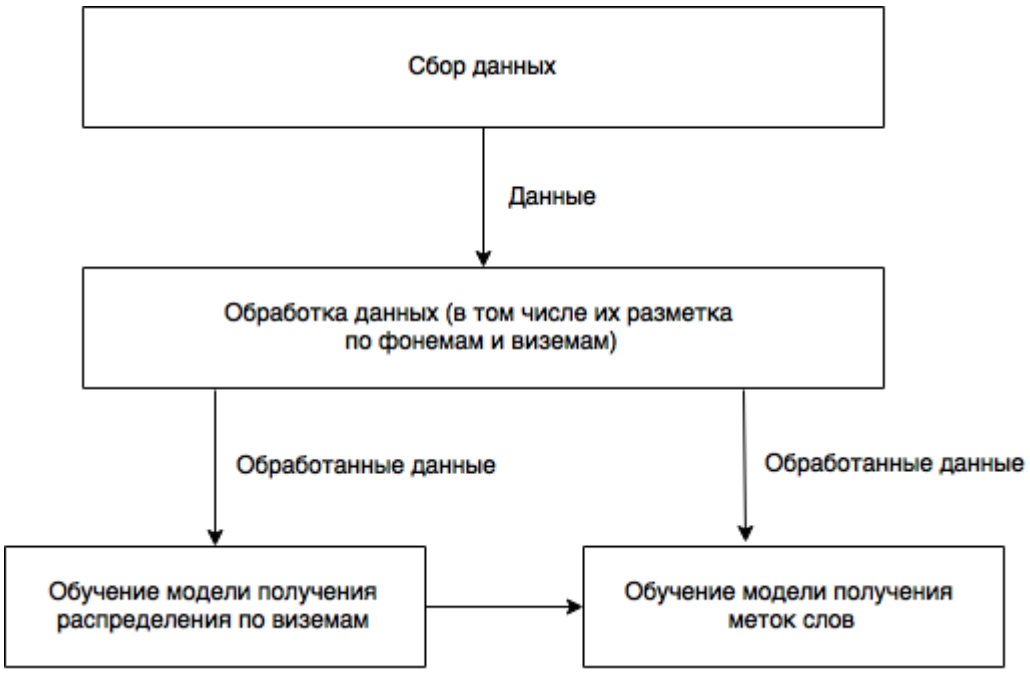

Рисунок 10 – Схема построения системы распознавания

Как будет детально показано ниже, предложенный в данный работе подход состоит как из ряда готовых компонент, каждый из которых была выбран исходя из соответствующих метрик, так и из использования новых для этой задачи подходов. Рассмотрим каждый из шагов построения системы распознавания по отдельности.

3.2. Сбор данных

Данный шаг подробно описан в предыдущей главе. Стоит лишь напомнить, что в качестве данных для обучения и тестирования были применены как существующих датасет GRID, так и был записан новый датасет Siri на основе подмножества команд популярного голосового ассистента под операционную систему iOS.

3.3. Обработка данных

3.3.1. Чтение видео-данных

Данный шаг не представляет научный интерес, однако он очень важен для успешной работы алгоритма и требует аккуратной технической реализаций. А именно требуется по входному видео отделить аудио и видео дорожки, а также разбить обе дорожки на минимальные единицы. Это может быть сделано одним из многих качественных библиотек для обработки видео. Для обработки видео-ряда в данной работе успешно применялась библиотека компьютерного зрения с открытым исходным кодом OpenCV.

3.3.2. Выделение признаков

Следующим шагом в процессе распознавания речи по визуальной информации является выделение вектора признаков, который будет описывать некоторым образом определенный фрейм видео. Как было показано в обзоре, это могут быть как некоторые геометрические признаки (зачастую, выбранные ручным образом), так и сырые данные, такие как значения пикселей в регионе интереса. Для алгоритма в данной работе было принято решение использовать подход с выделением ключевых точек с лица человека с последующей постобработкой этого вектора. Для этого необходимо на каждом фрейме найти ключевые точки губ лица человека.

Как было описано в секции 1.3 существует несколько наиболее популярных подходов для решения этой задачи. В качестве основного метода для

нашей работы выбран метод из статьи [17]. Важными преимуществами этого метода перед другими являются его сопоставимое качество по сравнению с проприетарными решениями, высокая скорость работы и возможность использования уже готовой реализации из открытой библиотеки машинного обучение dlib.

На выходе этого алгоритма мы имеем 68 точек, которые представляют из себя координаты ключевых точек на лице человек. Из них мы оставляем только необходимые нам 20 точек - координаты губ человека $(x_1, y_1, ..., x_{20}, y_{20})$. Схематически эти точки представлены на рисунке ниже (искомые точки губ на рисунке имет координаты $(x_{49}, y_{49}, ..., x_{68}, y_{68})$).

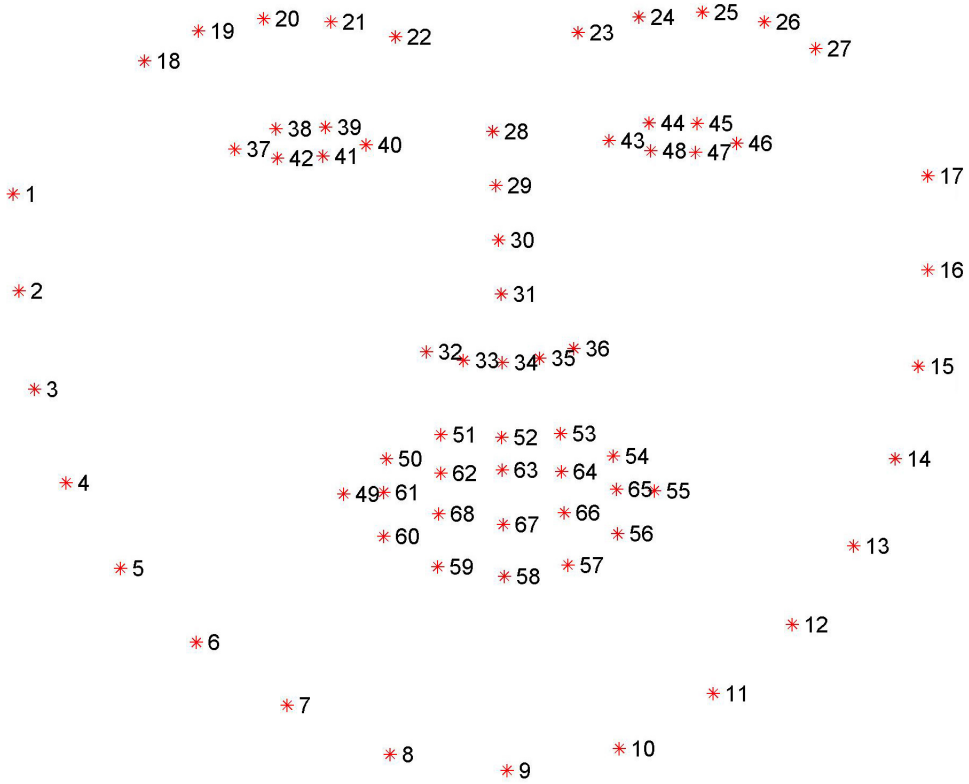

Рисунок 11 – Расположение ключевых точек на лице человека

Однако, использование этих точек в чистом виде, конечно, было бы нерационально, так как у положения губ может меняться как угол, так и их удаленность от камеры, что представляет из себя шум, от которого хочется избавиться. Для этого необходимо провести нормализацию полученных координат. В данной работе был использован несложный метод нормализации, предложенный в статье [28]. Метод состоит в переносе координат левого и правого уголка губ в точки $(-1, 0)$ и $(1, 0)$ с параллельным изменением координат всех

остальных точек. Рассмотрим формулы, по которым эти преобразования происходят.

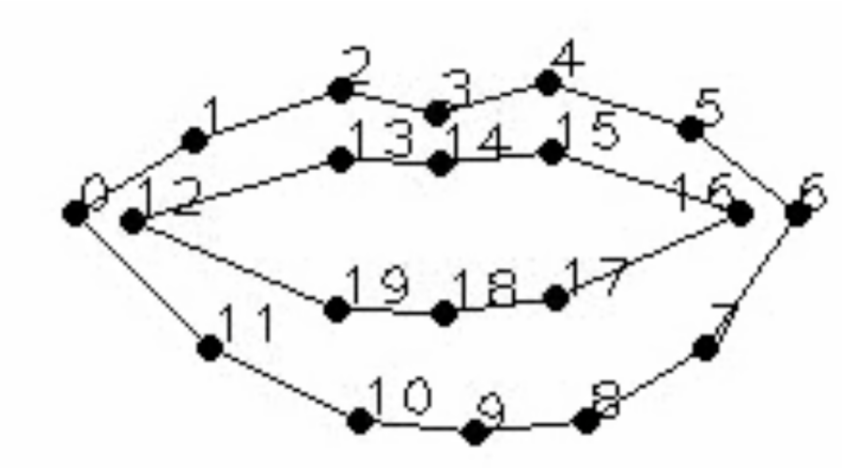

Рисунок 12 – Нумерация точек на губах человека

Пронумеруем все точки губ, как упомянуто выше, $(x_1, y_1, ..., x_{20}, y_{20})$. Таким образом необходимо, чтобы точка (x_1, y_1) оказалась в $(-1, 0)$, а точка (x_6, y_6) оказалась в $(1, 0)$. Для этого вычислим координаты вспомогательной точки, лежащей на середине отрезка $[(x_1, y_1), (x_6, y_6)]$

$$x_c = \frac{x_0+x_6}{2}, y_c = \frac{y_0+y_6}{2}$$

Угол поворота и радиус вычисляются по следующей формуле

$$\alpha = \tan^{-1} \frac{y_6-y_0}{x_6-x_0} \; r = \sqrt{(x_0 - x_c)^2 + (y_0 - y_c)^2}$$

Теперь для получения новых нормализованных координат всех остальных точек воспользуемся вычисленными углом поворота и радиусом следующим образом:

$$x_i^{(norm)} = \frac{(x_i-x_c)\cos\alpha+(y_i-y_c)\sin\alpha}{r}$$
$$y_i^{(norm)} = \frac{(y_i-y_c)\cos\alpha+(x_i-x_c)\sin\alpha}{r}$$

Итого после этих операций для каждого фрейма видео имеем по 40 чисел - координаты 20 нормализованных точек губ (список координат по оси x и по оси y). Казалось бы, на этом можно прекратить обработку, но существует две важные оптимизации

— Сглаживание координат;
— Увеличение FPS с 25 до 100 (апсемплинг).

Рассмотрим каждую из оптимизаций подробнее.

При попытке подсчета координат губ на каждом фрейме в некоторых случаях возникают так называемые выбросы (сильно неверные определения координат точек губ), а также возникает случайный шум (координаты отклоняются на соседних фреймах на небольшое значение). В качестве одной из гипотез, которая потенциально может частично побороть эти проблемы может выступить сглаживание координат точек губ. Как будет показано далее, такое сглаживание действительно дало некоторое улучшение в точности модели распознавания. А именно в работе было применено экспоненциальное сглаживание второго порядка (по каждой координате по отдельности). Более детально, алгоритм экспоненциального сглаживания первого порядка временного ряда $\{u_t\}_{t=1}^{t=T}$ представляет из себя

$$u_t^{(smooth)} = \alpha u_t + (1-\alpha)u_{t-1}^{(smooth)}, \text{ где } 0 < \alpha < 1 \text{ - параметр}$$

Однако стоит заметить, что в таком алгоритме, фактически, сглаживается первая производная последовательности (потому что для получения нового элемента, предыдущий элемент последовательности складывается с разностью последовательных элементов, умноженной на α.). Недостатком такого сглаживания является то, что оно довольно долго реагирует на скачки, что в нашей работе скорее навредит конечной модели, чем ей поможет. Поэтому было принято решение использовать экспененциальное сглаживание второго порядка, которое реагирует на изменение гораздо быстрее и может быть вычислено по следующим формулам

$$\begin{gathered} s_1 = u_1 \\ b_1 = 0 \\ s_t = \alpha u_t + (1-\alpha)(s_{t-1} + b_{t-1}) \\ b_t = \beta(s_t - s_{t-1}) + (1-\beta)b_{t-1} \\ u_t^{(smooth)} = s_t + b_t \end{gathered}$$

В работе применялись значения переменных $\alpha = 0.95, \beta = 0.1$, что, как будет показано дальше, дало общее улучшение качества модели.

Вторая же упомянутая выше оптимизация заключалась в увеличении числа фреймов в секунду с 25 до 100 линейной экстраполяцией. Для этого после вычисления координат точек между каждой парой соседних точек было добавлено еще 3 точки (другими словами, мы искусственно увеличил частоту кадров в 4 раза).

Рассмотрим в качестве примера две точки $(x_i, y_i), (x_{i+1}, y_{i+1})$. Координаты новых точек (x_{ij}, y_{ij}) вычисляются по следующим формулам:

$$\delta_x = (x_{i+1} - x_i)/4$$
$$\delta_y = (y_{i+1} - y_i)/4$$
$$x_{ij} = x_i + \delta_x * j$$
$$y_{ij} = y_i + \delta_y * j$$

Улучшение качества модели таким подходом может быть объяснено тем, что данные в видео 25 кадров в секунду - это достаточно мало для моделей распознавания речи, которые зачастую работают с данными значительно большей частоты. Этот подход широко применятся в ряде других работ (таких как [29]). В распознавании речи на основе аудио информации часто выбираются окна с шириной 25 мс и шаг 10 мс, что фактически совпадает с полученной после апсемплинга нами частотой (также 100 единиц в секунду).

3.3.3. Разметка данных по единицам речи

Дальнейшим шагом для алгоритма является обучение модели, которая бы по данным для обучения умела предсказывать распределения визем на них. Но для обучения такой модели нужна разметка по виземам для обучающих данных. Соответсвенно встает задача построения модели разметки данных по виземам. Но поскольку вручную сделать это практически невозможно в силу высоких трудозатрат на этот процесс, то необходимо использовать автоматический подход. Выбранный метод состоял в предварительной разметке видео по фонемам (использование аудио-дорожки видео для получения распределения вероятностей произнесения соответствующей фонемы на каждом фрейме) и последующем преобразовании этих фонем в виземы.

Для получения фонемы на каждом фрейме была выбрана уже готовая модель, основанная на датасете [30]. Такая модель основана на нейронной сети, которая на вход принимает признаки с аудио, а на выходе возвращает

распределение вероятностей по фонемам для соответствующего фрейма. Список возможных 60 фонем, на которых строилась данная модель, представлен в таблице ниже.

Таблица 4 – список использованных фонем

ux	f	el	v	aw	p
ah	ey	en	ch	uh	pau
jh	nx	dx	ih	hg	gcl
g	w	epi	q	ao	eng
l	axr	ow	n	m	sh
iy	hv	ae	dcl	d	y
er	aa	r	kcl	k	s
uw	tcl	t	ix	eh	oy
ay	dh	hh	z	pcl	ax
th	bcl	b	ax-h	zh	em

Полученная модель выбирала фонему с максимальной веротяностью из полученного распределения, что позволило добиться точности 71% на датасете TIMIT.

В дальнейшем, стояла задача получения распределения по виземам для соответсвующего фрейма. В данной работе использовался подход, в котором соответсвующее распределение получается из таблицы соответсвия между фонемами и виземами из работы [31].

В таблице 5 представлено упомянутое выше соответствие визем и фонем. Заметим, что одной виземе соответствует целая группа фонем. Это связано с тем, что многие фонемы на губах человека «выглядят» очень похоже друг на друга. К примеру, фонемы «ao» и «ow» выглядят довольно близко, поэтому они объединились в одну визему «O». Также стоит отметить, что при этом некоторым виземам все же соответсвует ровно одна фонема.

При подсчете финального распределения вероятностей визем на текущем фрейме, вероятности соответствующих по таблице фонем складывались.

Таблица 5 – Соответствие фонем и визем

Визема	Фонема
uh	uh uw ux
k	el en eng epi g hh hv k l n ng nx y
O	ao ix ow oy

Визема	Фонема
ey	iy ih
f	f v
ey	ey ae aw eh
t	d dh dx s d t th z
ah	ah ax ax-h ay
er	er axr

Визема	Фонема
aa	aa
ch	ch bcl dcl jh kcl pcl sh zh
w	w r
sil	sil pau gcl q tcl
p	p b em m

3.4. Построение акустической модели

Как только данные для обучения размечены по виземам, следующим логичным шагом алгоритма является разработка модели предсказания вероятностей визем на каждом фрейме. Для этого будут использоваться данные, полученные в предыдущем разделе, а именно стоит задача построить соответствие между соответствующим вектором признаком с визуальной информации и вектором распределения вероятности произнесенных визем на каждом фрейме. В качестве модели для распознавания использовалась нейронная сеть, которая задается выходной размерностью каждого слоя, вероятностью dropout в каждом слое и активационной функцией.

Однако важно заметить, что данных с одного фрейма недостаточно, чтобы предсказать визему, поскольку одно положение губ может соответствовать многим промежуточным состояниям разных визем. Решением этой проблемы служит конкатенация векторов с соседних фреймов и передача конкатенированного вектора сети.

Но в таком подходе возникает другая проблема - поскольку данные на соседних фреймах очень сильно коррелируют между собой, то вектор содержит много ненужной информации и, соотвественно, его размерность может быть уменьшена. Кроме того, вместе с этим мы избавимся от ненужного шума.

Существует ряд подходов по уменьшению размерности. В данной работе было рассмотрено и использовано два из них

— Метод главных компонент;
— Нейронные сети (autoencoder).

В первом подходе (метод главных компонент) сжатие вектора осуществлялось широко известному Kaiser rule [32], который позволяет выбрать оптимальное количество компонент в методе главных компонент. Метод утверждают, что должны остаться только те компоненты, собственное число которых больше среднего среди всех собственных чисел. В данной работе для PCA был выбрана размерность искомого вектора, равная 72.

Во втором подходе для эксперимента был взят autoencoder (нейронная сеть специального вида, которая предсказывает в точности тот же вектор, который подается на вход сети) с 1 скрытым слоем размером 72.

Кроме того, в обоих случаях были проведены эксперименты с использованием экспоненциального сглаживания (как первого, так и второго порядка), что, как будет показано в главе с экспериментами, дало небольшой прирост к точности модели.

3.5. Модель, распознающая метки слов

Как только мы научились получать распределение по виземам на каждом фрейме, то следующим логичным шагом идет построение модели, которая бы предсказывала метку слова по набору визем.

Важно отметить, что эту задачу можно ставить двумя принципиально разными способами

— Онлайн. Модели постепенно на вход подаются распределения на фреймах и она должна в ходе этого процесса предсказывать метки слов;

— Оффлайн. Модель имеет возможность сначала прочитать весь вход, пропустить его через себя, а уже только после этого сказать свое «предсказание» (последовательность меток слов).

В данной работе акцент был сделан именно на оффлайн задачу, потому что мы рассматриваем достаточно небольшой словарь и довольно небольшую длину фраз, поэтому с точки зрения использования продукта, в котором будет реализован предложенный алгоритм, пользователь не будет испытывать большой дискомфорт, что метки слов не выдаются по ходу движения его губ. Кроме того, модель в онлайн задаче имеет явный недостаток - по ходу распознавания метки слов проставляются и для первых фреймов каждого видео. А поскольку в момент распознавания данных еще не достаточно, чтобы верно определить искомое слово, то модель часто ошибается. Эта проблема не имеет место в оффлайн задаче.

Для алгоритма распознавания меток было принято решение использовать нейронную сеть, поскольку такая модель является state-of-the-art в распознавании речи. Однако классическая нейронная сеть не подходит для решения этой задачи, поскольку они на вход принимают вектор фиксированной длины (в нашем случае слова имеют разную длину и, следовательно, имеют представление в виде визем разной длины), поэтому была рассмотрена рекурентная нейронная сеть, а именно LSTM. Однако и реккурентная нейронная сеть в чистом виде не способна решить эту задачу, потому что она имеет выход фиксированной размерности. Для решение этой задачи (так называемая sequence-to-sequence learning) используется механизм encoder-decoder LSTM [33]. Основной идеей этого подхода является кодирование исходной последовательности в виде некоторого вектора фиксированной размерности (который будет «описывать» входные данные) и последующее декодирование этого вектора в искомые метки слов.

Более подробно процесс работы такого алгоритма можно описать следующим образом: на первом этапе алгоритм кодирует входную последовательность в некоторый вектор фиксированного размера, который «описывает эту последовательность». Это происходит за счет последовательной подачи распределений по виземам с каждого фрейма модели (на рисунке A, B, C). Впоследствии decoder пытается понять искомые метки слов до тех пор, пока не сгенерирует метку конца потока (на рисунке ниже eos). Заметим, что в процес-

се декодирования на каждый следующий декодер подается метка распознанного в предыдущей итерации слова, что позволяет модели запомнить некоторую структуру языка (своего рода, понять языковую модель).

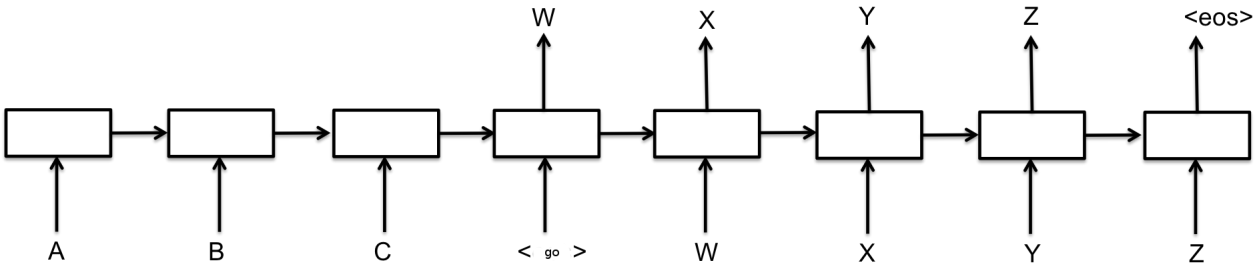

Рисунок 13 – Схема работы encoder-decoder LSTM

Стоит еще раз отметить, что выходом этой модели являются непосредственно метки слов. Слишком короткие слова, на которых распознавание является практически невозможным, помечаются некоторой отметкой специального класса (то есть все такие слова объединяются в один класс).

3.6. Схема полученного алгоритма

Как только каждая из полученных моделей была построена и обучена, то опишем непосредственно как работает алгоритм на новых данных.

Схематически алгоритм представлен на рисунке 14. На вход алгоритму поступает видеопоток данных, при этом в этих данных отсутствует звук, а есть лишь картинка человеческого лица.

На первом шаге алгоритма идет считывание соответсвующего фрейма, после чего полученное изображение отдается на вход модели, которая находит координаты ключевых точек на лице человека. При этом данные подготовлены таким образом, что в каждый момент времени человек смотрит четко в камеру, поэтому фреймов, на которых лицо человека не было найдено оказалось ничтожно мало (около 1 процента). На таких фреймах брались координаты с последнего успешного фрейма. Как только координаты были получены, происходит процедура нормализации, подробно описанная в секциях выше. Важно отметить, что как выбранный алгоритм получения ключевых точек на лице человека, так и алгоритм нормализации работают практически в режиме реального времени для видео, записанного с частотой кадров 25 кадров в секунду. Это происходит за счет эффективной реализации алгоритма в библиотеке dlib и знаниями о координатах точек с предыдущих фреймов в первом случае, и

простыми математическими преобразованиями в случаями с нормализацией, что позволяет рассчитывать на такую высокую скорость работы.

Рисунок 14 – Алгоритм распознавания меток слов

Следующим шагом алгоритма является получение вектора признаков для фрейма с учетом соседних фреймов. Для этого шага потребовалось объединить нормализованные вектора с соседних фреймов (в качестве параметра для этого

шага выступала ширина окна, которая будет фигурировать в главе с тестированием модели), а после уменьшить размерность полученного вектора методом главных компонент. Стоит отметить, что в моей реализации этот процесс происходил уже после того, как все данные были прочитаны, так как входные данные были относительно небольшого размера. Однако, этот шаг можно запускать асинхронно вместе с первым шагом.

Полученные сжатые вектора в дальнейшем идут на вход модели, которая, фактически, для текущего фрейма строит распределение вероятностей визем на нем. Поскольку данные приходят не только с текущего кадра, но и из нескольких соседних, то у модели, в большинстве случаев, достаточно информации, чтобы довольно точно понять какая визема произнесена на текущем фрейме.

И в качестве последнего шага алгоритма на вход модели encoder-decoder LSTM идут распределения визем с каждого фрейма. При этом на выходе этой модели генерируются метки слов, которые и являются результатом работы всего алгоритма.

ГЛАВА 4. ТЕСТИРОВАНИЕ ПОЛУЧЕННОЙ МОДЕЛИ

В ходе тестирования предстояло оценить качество полученной модели рапознавания визем и непосредственно модели определения меток слов. В ходе сравнения были выбраны классические подходы решения этой задачи, а также в качестве бейслайна был выбран подход, состоящий в ручной разметке слов.

4.1. Используемые технологии

В ходе экперимента обработка видео-данных происходила на языке C++. Вычисление вектора признаков производилось с помощью библиотек dlib и OpenCV. Для обработки аудио-данных и извлечения из них признаков применялся Python. Обучение нейронных сетей происходило на основе библиотеки Torch. Данные хранились преимущественно в hdf5. Кроме того, многие процессы обучения происходили на видеокарте с использованием CUDA. Также для некоторых вспомогательных скриптов для обработки результатов тестирования человеком, для загрузки видео с YouTube и их обработки были использованы Python и Bash.

4.2. Описание эксперимента

В качестве данных для тестирования модели определения качества распознавания визем был взят датасет GRID, поскольку в этой модели не требуется приближенность данных к реальному миру и в датасете GRID содержится большое количество размеченных данных, что позволит достаточно точно измерить качество работы модели. В свою очередь для тестирования второй модели (распознавание меток слов) был взят самостоятельно записанный датасет подмножества команд ассистента Siri, поскольку он является менее синтетическим и больше приближен к реальному миру.

Обе модели обучались на 80% данных, а тестировались на оставшихся 20% (таким образом, тестирование происходило speaker-independent, то есть множества спикеров для обучения и тестирования не пересекались).

4.3. Тестирование модели определения визем

Как уже было отмечено выше, для этого тестирования был использован датасет GRID. Из него для эксперимента было выбрано подмножество из 30 спикеров. При этом на 24 их них происходило обучение, а тестировалась модель на 6 оставшихся.

В ходе эксперимента была использована наивная реализация (где для определения виземы рассматривался только текущий фрейм), метод с использованием autoencoder и метода главных компонент. Кроме того, были проведены эксперименты с экспоненциальным сглаживанием второго и первого порядков.

Таким образом, каждая модель задается размером окна, которое использовалось для извлечения признаков, типом использованного алгоритма сжатия, количеством компонент после сжатия размерности и параметрами оптимизаций, описанных выше.

Результаты экспериментов приведены в таблице 6. Как видно из таблицы, наивный алгоритм сильно уступает по качеству алгоритмам, которые на свой вход принимают информацию с нескольких последовательных фреймов. Между алгоритмами, которые используют соответсвенно autoencoder и метод главных компонент, разница незначительная, поэтому во всех дальнейших экспериментах будем применять метод главных компонент. Также важно отметить, что использование экспоненциального сглаживания первого порядка несколько ухудшило результаты, потому что такое сглаживание медленно реагируют на изменения. В то же время экспоненциальное сглаживание второго порядка с параметрами $\alpha = 0.95, \beta = 0.1$ дало небольшой прирост в точности конечной модели предсказания визем.

Таким образом, лучшая полученная модель имеет точность 58%. Эта модель построена на нейронной сети с использованием relu активационной функции, дропаутом и применением экспоненциального сглаживания второго порядка. Именно эта модель и применялась в дальнейших шагах алгоритма. Необходимо подчеркнуть, что точность модели могла быть заметно выше при использовании более продвинутой модели для распознавания фонем (напомню, выбранная выше модель имеет точность 71%).

Таблица 6 – результаты тестирования модели предсказания визем

Описание модели	Точность распознавания визем
Окно 1 (только текущий фрейм) 128(sigmoid)+64(sigmoid)+14(sigmoid)	12%
Окно 10-5, autoencoder 72 128(sigmoid)+64(sigmoid)+14(sigmoid)	53%
Окно 10-5, pca 72(opt) 128(sigmoid)+64(sigmoid)+14(sigmoid)	54%
Окно 10-5, pca 72(opt) эксп. сглаживание 1 пор. ($\alpha = 0.97$) 128(sigmoid)+64(sigmoid)+14(sigmoid)	52%
Окно 20-10, pca 80(opt) эксп. сглаживание 2 пор. ($\alpha = 0.95, \beta = 0.1$) 140(sigmoid)+80(sigmoid)+14(sigmoid)	55%
Окно 15-10, pca 75(opt) эксп. сглаживание 2 пор. ($\alpha = 0.95, \beta = 0.1$) 110(relu)+80(relu)+14(relu)	56%
Окно 15-10, pca 75(opt) эксп. сглаживание 2 пор. ($\alpha = 0.95, \beta = 0.1$) 110(relu,dropout=0.1)+ 80(relu,dropout=0.15)+14(relu)	58%

4.4. Тестирование модели определения меток слов

Для тестирования модели распознавания были взяты подходы, описанные в обзоре, ручной подход (определение распознанного слова человеком) и предложенный в данной работе метод. Рассмотрим выбранные конфигурации каждого метода по отдельности.

4.4.1. Алгоритм на основе HOG и Eigenlips

Подробно принципы работы этих методов описаны в обзоре, но стоит заострить внимание на основной идее этого подхода - на первом шаге алгоритма для каждого фрейма считается некоторым образом вектор, описывающий этот фрейм. После для нескольких последовательных фреймов такие вектора объединяются, и этот вектор отправляется на вход Support Vector Machine. Важно сказать, что такой алгоритм принимает на вход вектор фиксированного размера, поэтому полученный вектор усредняется и приводится к заданной длине.

Также важно сказать, что поскольку алгоритм спроектирован для классификации одного слова, то границы слов в фразе были искуственно «сказаны» алгоритму.

4.4.2. Алгоритм на основе геометрических признаков

В качестве выбранных геометрических признаков были взяты признаки из [6]. Общая же схема алгоритма похожа на ту, что описана в предыдущем пункте, за исключением того, что для классификации был использован алгоритм KNN (K-nearest neighbors).

4.4.3. Алгоритм на основе нейронной сети

В качестве альтернативы предыдущего метода существует подход, который заключался в использовании нейронной сети в качестве классификатора для меток слов. Обратим внимание, что поскольку число входных фонем изменялось, то была реализована рекуррентная нейронная сеть, которая и использовалась для классификации. Такой алгоритм, однако, работает только с классификацией одного слова, поэтому фраза была искусственно разбита на части и каждая из частей была скормлена сети по отдельности.

4.4.4. Алгоритм распознавания меток человеком

В качестве простейшего бейслайна было предложено человеческому глазу распознать фразы из датасета Siri. Для этого в эксперименте принимало участие 5 человек, каждому из которых показывались видео с фразами без звука и предлагалось выбрать один из 15 вариантов. Результаты по каждому из испытуемых представлены в таблице ниже.

Таблица 7 – ручной алгоритм

Человек	Точность
Человек №1	36%
Человек №2	39%
Человек №3	56%
Человек №4	60%
Человек №5	39%

Таким образом средняя точность распознавания в этом подходе равна 46%.

4.4.5. Предложенный в работе подход

В качестве конфигурации алгоритма из данной работы в эксперименте задавались основные параметры encoder-decoder LSTM, а именно размеры слоев encoder и decoder. Кроме того, модель конфигурировалась вероятностью dropout.

4.4.6. Сравнение подходов

Для тестирования модели был взят записанный датасет Siri. При этом в обучении использовалось 8 спикеров, а для тестирования 2 спикера.

В таблице ниже представлены результаты сравнения моделей. Фраза считалась точно распознанной, если правильно распознаны все слова в этой фразе.

Как видно из таблицы, из классических существующих подходов наилучшие результаты показывает модель с нейронной сетью, при этом, как ожидалось, ручной подход выступает в качестве бейслайна и показывает худшие результаты среди всех подходов. Среди подходов с использованием различного рода более простых, чем нейронная сеть, классификаторов (KNN, SVM) наилучшие результаты показала комбинация алгоритма Support Vector Machine и HOG признаков, в то время как геометрические признаки в паре с KNN дали худшие из результатов среди всех автоматических алгоритмов.

Таблица 8 – некоторые результаты с архитектурой encoder-decoder LSTM (encSize и decSize - размеры слоев сети)

Метод	Конфигурация	Точность
Ручной алгоритм	число тестеров=5	44%
KNN подход	Геометрические признаки	49%
SVM подход	Eigenlips признаки	68%
SVM подход	HOG признаки	70%
NN классификатор	реккуррентная нейронная сеть	75%
Предложенный метод	encSize=128 decSize=50	82%
Предложенный метод	encSize=100 decSize=40	81%
Предложенный метод	encSize=110, dropout=0.2 decSize=60, dropout=0.1	84%
Предложенный метод	encSize=128, dropout=0.2 decSize=40	85%

Что касается предложенного в данной работе метода, то он показывает наилучшие результаты по поставленной задаче. Это может быть в том числе объяснено тем, что предложенная структура (encoder-decoder LSTM) как хорошо работает на отдельно взятых словах, так и имеет способность «подстраиваться» под структуры конкретных фраз, что позволяет ей точнее определять некоторые слова. Также заметим, что среди вариантов конфигурации предложенного алгоритма лучше всех себя показала модель с использованием droupout=0.2.

ЗАКЛЮЧЕНИЕ

В рамках данной магистерской работы был предложен новый метод распознавания речи, основанный исключительно на визуальной информации, который во-первых является speaker-independent, а во вторых позволяет добиться высокой точности на ограниченном словаре, которая превосходит точность существующих классических подходов.

Такой подход может быть использован для реализации алгоритма чтения по губам для некоторого подмножества команд известных голосовых ассистентов (таких как Siri) и для камер наблюдения (детектировать специфические подозрительные слова). Кроме того, минимальные модификации предложенного алгоритма могут быть использованы для разного вида идентификации и авторизации пользователя (как это описано в работе [6]) в тех ситуациях, когда произносить фразу в слух является небезопасным.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Siri mobile assistant. — URL: https://www.apple.com/ios/siri/.

2 Intelligent personal assistant Cortana. — URL: https://www.microsoft.com/en/mobile/experiences/cortana/.

3 *Karpov A.* An automatic multimodal speech recognition system with audio and video information // Automation and Remote Control. — 2014. — Дек. — Т. 75, № 12. — С. 2190–2200. — URL: https://link.springer.com/article/10.1134/S000511791412008X.

4 Speech technologies in multimodal interfaces / A. A. Karpov [и др.] // SPIIRAS Proceedings. — 2004. — Т. 1, № 2. — С. 254–256. — ISSN 2078-9599. — URL: http://www.proceedings.spiiras.nw.ru/ojs/index.php/sp.

5 *Sak H.*, *Senior A. W.*, *Beaufays F.* A Novel Motion Based Lip Feature Extraction for Lip-reading. — 2008. — URL: http://www.comp.hkbu.edu.hk/~ymc/papers/conference/cis08_publication_version.pdf.

6 *Hassanat A. B.* Visual Passwords Using Automatic Lip Reading. Abs/1409.0924. — 2014. — URL: http://arxiv.org/abs/1409.0924.

7 *Stanislav S.* Lip Reading: Preparing Feature Vectors // Proc. Int. Conf. Graphicon'03. — 2003. — С. 254–256. — URL: https://pdfs.semanticscholar.org/4166/53d02dcc4acd67572f028ccad643d28e6eef.pdf.

8 *Крак Ю.*, *Бармак А.*, *Тернов А.* Информационная технология для автоматического чтения по губам украинской речи // Компьютерная математика. — 2009. — Т. 1. — С. 86–95. — URL: http://dspace.nbuv.gov.ua/handle/123456789/6232.

9 *Christoph Bregler Y. K.* "EIGENLIPS" FOR ROBUST SPEECH RECOGNITION //. — 1994.

10 *King D.* Dlib-ml: A machine learning toolkit // The Journal of Machine Learning Research 10. — 2009. — С. 1755–1758.

11 *Pei Y.*, *Kim T.-K.*, *Zha H.* Unsupervised Random Forest Manifold Alignment for Lipreading // 2013 IEEE International Conference on Computer Vision. — 2013. — URL: http://ieeexplore.ieee.org/document/6751125/.

12 *Paleček K.* Extraction of Features for Lip-reading Using Autoencoders //. — 2005.

13 *Cootes T. F.*, *Edwards G.*, *Taylor C.* Comparing Active Shape Models with Active Appearance Models. — 1999.

14 *Timothy F. Cootes Gareth J. Edwards C. J. T.* Active Appearance Models. — 2008. — URL: http://www.comp.hkbu.edu.hk/~ymc/papers/conference/cis08_publication_version.pdf.

15 *Le T. H.*, *Vo T. N.* Face Alignment Using Active Shape Model And Support Vector Machine // CoRR. — 2012. — T. abs/1209.6151. — URL: http://arxiv.org/abs/1209.6151.

16 *Timothy F. Cootes Gareth J. Edwards C. J. T.* Improving Visual Features for Lip-reading. — 20111. — URL: https://pdfs.semanticscholar.org/6778/68449c6b05a3df45d25a18f9782550b69661.pdf.

17 *Kazemi V.*, *Sullivan J.* One Millisecond Face Alignment with an Ensemble of Regression Trees // CVPR. — 2014.

18 *Virginia Estellers J.-P. T.* Multi-pose lipreading and Audio-Visual Speech Recognition. — 2012. — URL: http://vision.ucla.edu/~virginia/publications/Estelle2012EURASIP.pdf.

19 *Jendoubi S.*, *Yaghlane B. B.*, *Martin A.* Belief Hidden Markov Model for speech recognition // CoRR. — 2015. — T. abs/1501.05530. — URL: http://arxiv.org/abs/1501.05530.

20 Exploring the Limits of Language Modeling / R. Jozefowicz [и др.] // CoRR. — 2016. — T. abs/1602.02410. — URL: http://arxiv.org/abs/1602.02410.

21 Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition / H. Sak [и др.] // CoRR. — 2015. — T. abs/1507.06947. — URL: http://arxiv.org/abs/1507.06947.

22 *Bagai A.*, *Gandhi H.*, *Goyal R.* Lip-Reading using Neural Networks // IJCSNS International Journal of Computer Science and Network Security. — 2009. — T. VOL.9 No.4.

23 Complete list of Siri commands. — URL: https://www.cnet.com/how-to/the-complete-list-of-siri-commands/.

24 *Cookea M.*, *Jon Barker Stuart Cunningham X. S.* An audio-visual corpus for speech perception and automatic speech recognition. — 2006. — URL: http://laslab.org/upload/an_audio-visual_corpus_for_speech_perception_and_automatic_speech_recognition.pdf.

25 Index of AVLetters [HTML]. — URL: http://www2.cmp.uea.ac.uk/~bjt/avletters/.

26 Index of LiLiR [HTML]. — URL: http://www.ee.surrey.ac.uk/Projects/LILiR/datasets.html.

27 CUAVE: A new audio-visual database for multimodal human-computer interface research / E. K. Patterson [и др.] // In Proc. ICASSP. — 2002. — С. 2017–2020.

28 *Wang S. L.*, *Lau W. H.*, *Leung S. H.* Automatic Lip Contour Extraction from Color Images // Pattern Recogn. — New York, NY, USA, 2004. — Дек. — Т. 37, № 12. — С. 2375–2387. — ISSN 0031-3203. — DOI: 10.1016/j.patcog.2004.04.016. — URL: http://dx.doi.org/10.1016/j.patcog.2004.04.016.

29 *Yuxuan Lan R. H.*, *Theobald B.-J.* Insights into machine lip reading. — 2012. — URL: https://pdfs.semanticscholar.org/c573/c71213b46a2b966546c7b7848b5bbe0536ec.pdf.

30 TIMIT Acoustic-Phonetic Continuous Speech Corpus / J. Garofolo [и др.]. — URL: https://catalog.ldc.upenn.edu/LDC93S1.

31 *Benedikt L.* Facial Motion: a novel biometric? — 2010.

32 *Jackson D. A.* Stopping rules in principal component analysis: a comparison of heuristical and statistical approaches. — 1993.

33 *Sutskever I.*, *Vinyals O.*, *Le Q. V.* Sequence to Sequence Learning with Neural Networks // CoRR. — 2014. — T. abs/1409.3215. — URL: http://arxiv.org/abs/1409.3215.