Санкт-Петербургский Государственный Университет
Математико-механический факультет

Математическое обеспечение и администрирование информационных систем
Кафедра информационно-аналитических систем

Савченков Павел Александрович

Автоматическое распознавание речи на основе визуальной информации

Бакалаврская работа

Научный руководитель:
доцент, к.ф.-м.н. Михайлова Е. Г.

Рецензент:
ст. преп. Немешев М. Х.

Санкт-Петербург
2016

SAINT-PETERSBURG STATE UNIVERSITY

Department of Analytical Information Systems

Pavel Savchenkov

Automatic speech recognition based on visual information

Bachelor's Thesis

Scientific supervisor:
associate professor, PhD Elena Mikhailova

Reviewer:
senior assistant professor Marat Nemeshev

Saint-Petersburg
2016

Оглавление

Введение

В большинстве случаев под распознаванием речи подразумевают преобразование аудио-последовательности записи голоса человека в текстовые данные. Однако, в некоторых случаях использование не только звуковой, но и видео-информации позволяет улучшить качество распознавания или даже заменить аудио-модели.

Системы основанные на визуальных признаках могут использоваться для аутентификации [9], реализации интерфейсов ввода информации или управления. Последнее особенно актуально в связи с широким распространением мобильных устройств, использование которых часто происходит в зашумленных условиях, сильно понижающих качество распознавания аудио-сигнала. Также данный подход может использоваться в случаях, когда человек по каким-то причинам не имеет возможности говорить вслух.

Однако распознавание речи, основанное на визуальной информации в общем случае сложнее анализа аудио-сигнала. Человеческая речь содержит порядка 50 фонем (минимальная различимая единица аудио-потока) в то время как по губам возможно различить порядка 10-15 визем (групп визуально неразличимых фонем). Таким образом, последовательность визем часто может не соответствовать конкретному слову и точность чтения по губам сильно зависит от контекста. Кроме того, даже среди людей говорящих на одном диалекте соответствие между движениями губ и произнесенными виземами может очень сильно различаться, что делает почти невозможным построение общей видео-модели распознавания без априорной информации о "стиле" движения губ человека.

В данной работе рассматривается проблема распознавания слитной речи на основе визуальной информации (фактически - чтение по губам) с маленьким словарем и небольшим количеством произнесенных слов на рассматриваемом отрезке видеоряда.

Постановка задачи

В рамках данной работы были поставлены следующие задачи:

- Провести обзор существующих подходов к распознаванию речи
- Выделить и описать основные этапы работы алгоритмов распознавания речи
- Предложить подход к преобразованию речи в текст, основанный только на визуальной информации
- Поставить ряд экспериментов и определить применимость предложенных методов

1. Обзор существующих методов

Системы распознавания речи можно условино разделить на несколько классов по тому, какие типы последовательностей слов они способны анализировать:

- Отдельные слова

  Такие системы опираются на то, что произношение каждого слова будет окружено тишиной с обоих сторон, то есть вынуждают говорящего делать паузы между соседними вхождениями. Выделение пауз в речи представляется отдельной задачей. В такой формулировке задачу распознавания можно (но не обязательно) рассматривать как задачу классификации

- Связная речь

  Некоторым вхождениям слов разрешается идти друг за другом с минимальной паузой

- Слитная речь

  Практически естественная речь человека. Одна из сложнейших задач распознавания

В данной работе рассматривается класс, лежащих между первым и вторым пунктом - распознавание слов небольшого словаря на записях, содержащих несколько произнесенных слов подряд.

Большинство подходов к распознаванию речи можно разбить на следующие последовательные шаги:

- Препроцессинг

  Включает выделение отрезков речи/не речи

- Извлечение признаков

- Декодирование

  Собственно, расшифровка сказанной информации. Происходит с использованием:

– Акустической модели

  Описывает зависимость между аудио-сигналом и единицами речи (почти всегда фонемами)

– Словаря

  Множество произносимых слов вместе с их транскрипциями

– Языковой модели

  Распределение вероятностей над множествами всех предложений или отдельных слов

- Постпроцессинг

  На выходе предыдущего пункта может получиться набор возможных последовательностей слов вместе с их оценками (например, вероятностями). При выборе окончательного ответа можно использовать какие-либо высокоуровневые требования, которые не было возможности принять во внимание на описанных ранее этапах.

Схематически процесс можно представить следующим образом:

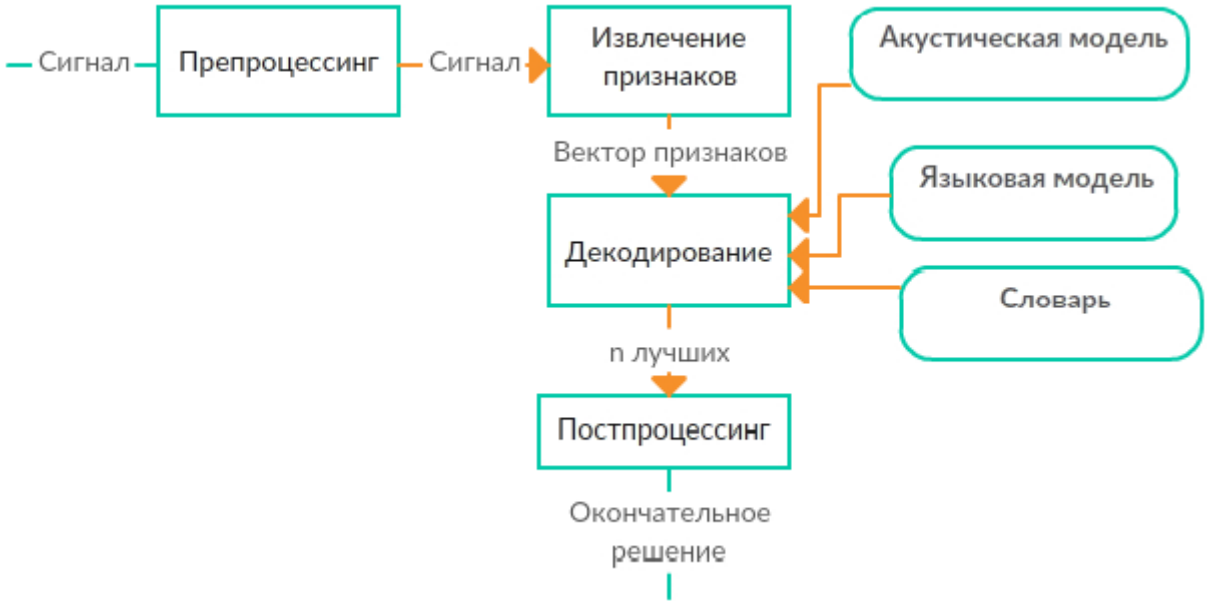

Рис. 1: Общий вид процесса распознавания речи

Также обычно фиксируются эталонные выборки данных для обучения, тестирования и сравнению с другими работами. В рамках данной работы ос-

новное внимание уделяется извлечению визуальных признаков, акустической модели и последующему декодированию в последовательность слов.

1.1. Подходы к построению акустической и языковой моделей

Во многих статьях относящихся к распознаванию речи долгое время базовым подходом считались скрытые марковские модели (например, [30], [15]), однако в последнее время активно развиваются методы, основанные на применении нейронных сетей, в особенности рекуррентных (например, [8]). Далее приводится более подробный обзор указанных методов.

1.1.1. Скрытые марковские модели

Скрытая марковская модель (СММ) представляет из себя систему, имитирующую случайный процесс, эволюция которого зависит только от текущего состояния модели и не зависит от предыдущей истории. В таком случае ставится задача нахождения значений скрытых (неизвестных) параметров модели при заданной последовательности наблюдаемых значений.

Таким образом марковский процесс можно изобразить так:

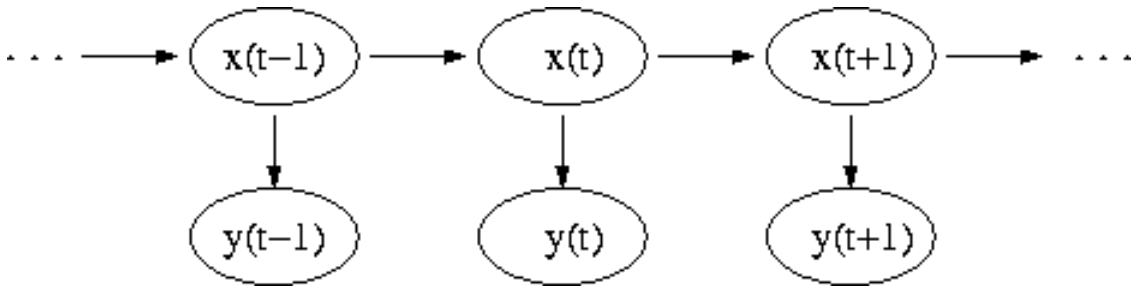

Рис. 2: Пример марковского процесса

Здесь $x_t \in \{1, \ldots, N\}$ - скрытое состояние модели в момент времени t, $y_t \in R^d$ - наблюдаемая переменная в момент времени t. Значение y_t зависит только от x_t, а x_t только от состояния в момент времени $t-1$, то есть от x_{t-1}. Для дальнейшего задания модели требуется определить матрицу вероятностей переходов $a \in R^{N \times N} \mid a_{i,j} = p(x_t = j | x_{t-1} = i)$, распределения наблюдаемых переменных для каждого состояния $p(y_t|x_t)$ и распределение

вероятностей начальных состояний $p(x_1)$. Обычно распределние $p(y_t|x_t) = b_t$ делают нормальным $b_t(y) = \mathcal{N}(y; \mu_t, \Sigma_t)$.

В общем случае вероятностные модели применяются при распознавании речи следующим образом [19]. После извлечения векторов признаков из сигнала $Y = y_1, \ldots, y_T$ декодер пытается найти последовательность слов $w = w_1, \ldots, w_L$, которыми с наибольшей вероятностью сгенерирована Y, другими словами находит $w_{best} = argmax_w\{P(w \mid Y)\}$. Часто неудобно работать с вероятностью $P(w \mid Y)$, поэтому, используя формулу Байеса, ее заменяют на

$$argmax_w\{\frac{P(Y|w)\cdot P(w)}{P(Y)}\} = argmax_w\{P(Y \mid w) \cdot P(w)\}$$

Здесь $P(Y \mid w)$ определяется акустической моделью, а $P(w)$ - языковой.

Каждому слову соответствует упорядоченная последовательность фонем (возможно не одна, поскольку у слова может быть несколько правильных произношений). Акустическая модель для конкретного слова получается конкатенацией моделей для отдельных фонем, которые, в свою очередь, обучаются на размеченной тренировочной выборке, например, с помощью алгоритма прямого-обратного хода [25]. Можно уточнить формулу вычисления $P(Y \mid w)$ как суммирование по всем возможным произношениям последовательности w

$$P(Y \mid w) = \sum_Q P(Y \mid Q) \cdot P(Q \mid w)$$

Первый множитель $P(Y \mid Q)$ при заданной марковской модели для $Q = q^{(w_1)}, \ldots, q^{(w_L)}$, полученной конкатенацией моделей для своих составляющих, считается как сумма вероятностей по всем последовательностям состояний $\Theta = \theta_0, \ldots, \theta_{T+1}$, а именно

$$P(Y \mid Q) = \sum_\Theta P(\Theta, Y \mid Q) = \sum_\Theta a_{\theta_0,\theta_1} \prod_{t=1}^{T} b_{\theta_t}(y_t) a_{\theta_t,\theta_{t+1}}$$

Состояния θ_0 и θ_{T+1} введены для удобства последовательной конкатенации моделей друг с другом и большой роли не играют.

Второй же множитель считается как просто произведение вероятностей по каждому слову, что оно было произнесено соответствующим образом:

$$P(Q \mid w) = \prod_{l=1}^{L} P(q^{(w_l)} \mid w_l)$$

.

Однако, обычно описанные модели используются не для нахождения последовательности сказанных слов, а для распознавания фонем, то есть для получения распределения вероятностей произнесенных фонем в каждый момент времени. Дальшейший анализ производится с помощью языковой модели и алгоритмов, специализированных для конкретной задачи. В качестве примеров можно привести Weighted Finite State Transducer [20] и нейронные сети, про которые речь пойдет далее [7].

1.1.2. Нейронные сети

Нейронная сеть - это модель, используемая для оценки неизвестной функции $f(x) : R^{d_1} \to R^{d_2}$, зависящей от большого числа параметров. Она представляет из себя набор взаимодействующих между собой т.н. нейронов. Соединения между ними имеют численные веса, которые могут быть пересчитаны (обучены) для лучшего соответствия заданным входным данным. Такая модель впервые появилась в 40-х годах прошлого века, вдохновленная базовыми принципами работы человеческого мозга.

В общем случае нейронная сеть может быть визуализирована таким образом:

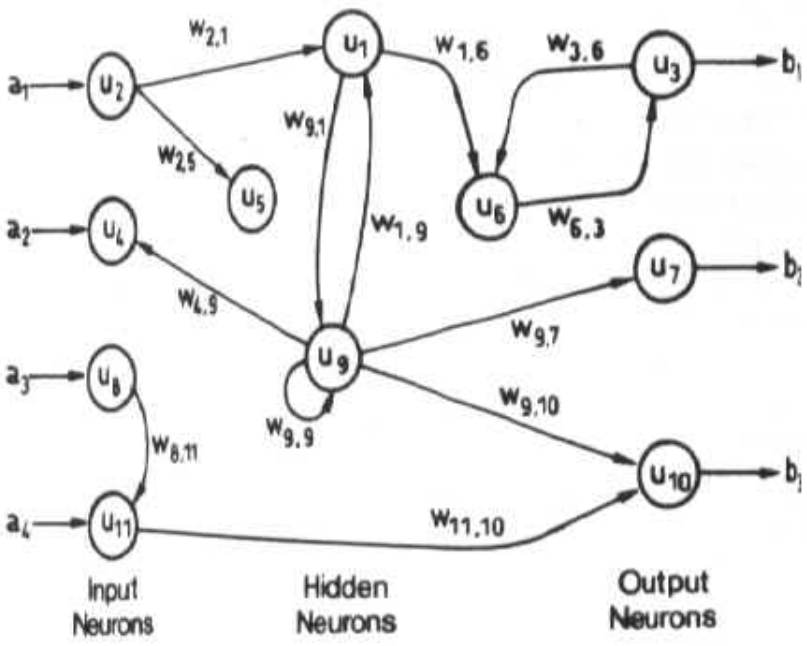

Рис. 3: Общий вид нейронной сети

Здесь сеть принимает на вход вектор (a_1, a_2, a_3, a_4) и выдает результат (b_1, b_2, b_3).

Обычно структура нейронной сети представляет из себя последовательность соединенных между собой слоев, причем для каждой пары соседних слоев заданы веса $w_{i,j}$ между i-м нейроном 1 слоя и j-м нейроном 2-го (при этом не обязательно соединять все пары нейронов). Таким образом, если $u_i^{(k)}$ - выход i-го нейрона k-го слоя, то на вход к j-му нейрону $k+1$-го слоя подается $\sum_i w_{i,j}^{(k)} \cdot u_i^{(k)}$. Все слои кроме первого (входного) и последнего (выходного) называются скрытыми. Выход нейрона определяется его активационной функцией, применяемой ко входу. Популярными функциями являются сигмоидная $\sigma(z) = \frac{1}{1+e^{-z}}$, пороговая $\phi(z) = max(0, z)$, гиперболический тангенс $\phi(z) = \frac{2}{1+e^{-2z}} - 1$, rectified linear unit $f(x) = max(x, 0)$ или тождественная функция $\phi(z) = z$.

Таким образом в базовом случае сеть выглядит так:

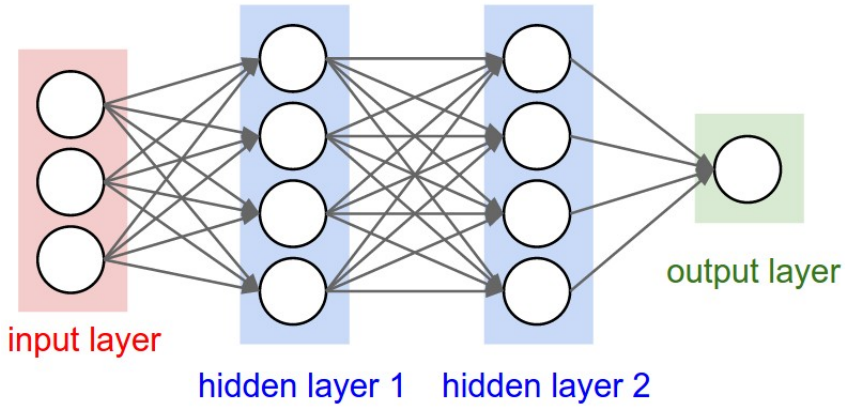

Рис. 4: Пример нейронной сети

Подобные слои называются линейными, поскольку представляют из себя умножение входного вектора на матрицу весов $w_{i,j}^{(k)}$ на k-ом слое. На практике к такому преобразованию обычно добавляют смешение b_k, то есть выходной вектор k-го слоя $u^{(k)}$ вычисляется через предыдущий слой как

$$u^{(k)} = \phi^{(k)}(A^{(k)} \cdot u^{(k-1)} + b_k),\ A^{(k)} \in R^{d_1 \times d_2},\ b_k \in R^{d_1}$$

.

В случае дифференцируемых функций активации преобразование $f(x)$, задаваемое сетью также дифференцируемо, что позволяет вычислять градиенты в точке по параметрам $w_{i,j}^{(k)}$.

При условии задания обучающей выборки $\{(x_i, y_i)\}$ и функционала потерь $L(y_i, f(x_i))$ (также дифференцируемого) становится возможным оптимизировать параметры сети с помощью метода градиентного спуска. Такой способ обучения получил название метода обратного распространения ошибки поскольку вычисление градиента идет в противоположном порядке по сравнению с вычислением значения функции $f(x)$.

Также существуют специальные нелинейные слои, заточенные под решение различных классов задач. Например, сверточные слои [14], принимающие на вход изображения (обычно двумерные матрицы) и на выходе получающие какие-либо комбинации их сверток с различными ядрами. Также существуют преобразования, осуществляющие свертки с функцией max или min – max и

min-pooling соответственно. Подобные подходы в данный момент считаются state-of-the-art в анализе изображений. Кроме того, есть слои помогающие с проблемой переобучения - dropout (убирает долю связей между слоями во избежание слишком хорошей подгонки под тренировочные данные) [6] или batch normalization (нормализация данных не только перед загрузкой из в сеть, но и в процессе прохождения их между слоями) [12]. Все упомянутые подходы также позволяют считать градиенты по параметрам сети и соответственно обучать модель методом градиентного спуска.

Однако в рассматриваемой задаче сопоставления видеозаписи последовательности векторов признаков $Y = y_1, \ldots, y_T$ она может быть достаточно длинной, T не всегда фиксировано. Также требуется на выходе получать последовательность слов, а не единственную метку класса. В связи с подобными проблемами появились т. н. рекуррентные нейронные сети, которые обладают "обратной связью", другими словами позволяют передавать накопленную по ходу информацию дальше во времени.

Визуально такой процесс можно изобразить следующим образом:

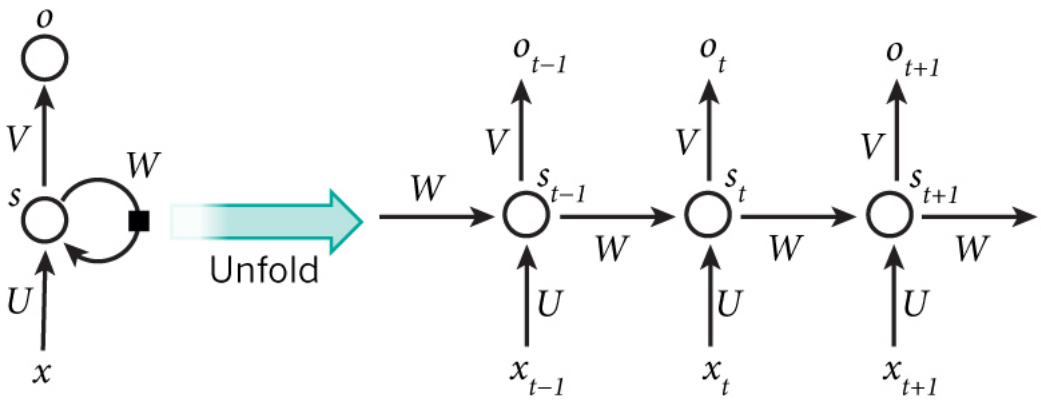

Рис. 5: Процесс работы рекурретной нейронной сети

Здесь сеть в момент времени t принимает входной вектор x_t, скрытое состояние на предыдущем шаге s_{t-1} и вычисляет выходной вектор o_t. После этого происходит новое состояние s_t передается в следующую итерацию процесса. Такие сети позволяют обрабатывать последовательности неизвестной длины, учитывая связи между настоящим и прошлым.

Основным методом обучения таких сетей является метод обратного распространения ошибки во времени. Он работает следующим образом: проис-

ходит ”развертка” сети во времени, то есть если входная последовательность имела длину T, то сеть скопируется T раз. После этого сеть рассматривается как нерекуррентная, поэтому можно запустить обычный алгоритм обратного распространения ошибки и посчитать градиенты по весам (здесь существуют варианты запускать пересчет весов сразу для всей последовательности длины T или для каждого префикса длины $1 \leq T' \leq T$). Затем сеть обратно сворачивается в исходное состояние и происходит обновление весов средним значением градиента из всех копий сети.

Простейшие архитектуры подобных сетей хоть и способны обрабатывать последовательности произвольной длины, но подвержены таким проблемам при обучении как ”проблема изчезающего градиента” и ”проблема взрывающегося градиента”. Они связаны с тем, что при последовательном умножении маленьких/больших чисел результат растет/падает экспоненциально и градиент при обратном распространении ошибки может стать очень большим либо очень близким к нулю. В обычных нейронных сетях такая проблема также присутствует, но в рекуррентных моделях она выражена особенно остро. По тем же причинам, такие сети плохо учитывают завимости между сравнительно далеко отстоящими во времени моментами.

Для борьбы упомянутыми проблемами была придумана специальная архитектура под названием Long Short Term Memory [24]. Визуально слой такой сети можно изобразить следующим образом:

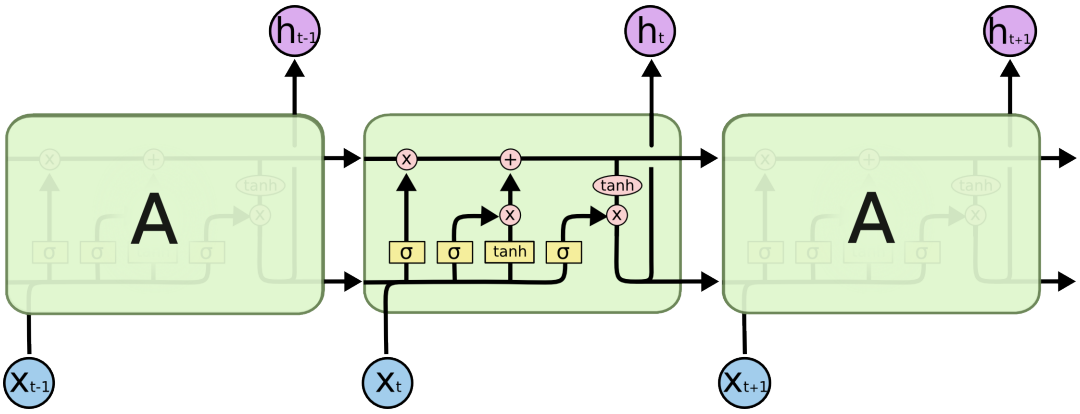

Рис. 6: Структура слоя LSTM

Главная идея такой сети - это введение т. н. cell state, к которому в момент времени t последовательно применяются forget gate и update gate. Таким

образом состояние сети проходит сквозь такие преобразования удалив часть информации, добавив новую и какую-то долю старого состояния оставив почти без изменения. Таким образом долгосрочные зависимости могут долго сохраняться при работает такой сети. Кроме того по причине нормализации данных с помощью сигмоидной и тангенсной активационных функций проблемы исчезающего и взрывающегося градиента гораздо меньше проявляют себя. Подобная архитектура очень часто используется при работе с рекуррентными нейронными сетями и подавляющем большинстве случаев получаются модели лучшего качества, чем при использовании стандартных рекуррентных моделей.

Рекуррентные нейронные сети успешно используются при построении акустических [8] и языковых [7] моделей.

В данной работе при выборе между скрытыми марковскими моделями и нейронными сетями выбор сделан в пользу нейронных сетей, так как они делают меньше предположений о природе входных данных и в тоже время являются практически state-of-the-art решениями в распознаваниии речи.

1.2. Обзор методов выделения признаков

Среди подходов к выделению признаков из объектов на изображениях можно выделить алгоритмы, основанные на выделении контуров объекта (в рассматриваемом случае - губ) и методы, работающие напрямую со значениями пикселей в некоторой области на изображении. Также существуют алгоритмы, использующие методы трекинга лица для улучшения точности признаков в случае, если входом является не фиксированная картинка, а видеоряд (например, [23]).

1.2.1. Дескрипторы, основанные на выделении контуров

К классическим методам первой группы можно отнести активные модели внешнего вида (active appearance models) [27] и активные модели формы (active shape models) [18]. Они представляют из себя методы подгона некоторой статистической модели изображений, учитывающей форму или текстуру

объекта, под конкретную картинку [5]. На похожих идеях основано множество подходов к нахождению ключевых точек на изображениях (например, [28] [17]). Выходом данных алгоритмов является множество ключевых точек на изображении, обозначающих форму лица, губ, глаз и т.п.

Существуют готовые реализации некоторых подходов разметки лица, описанных в статьях. Например, на изображении приведен пример работы приложения IntraFace, основанного на статье [33]

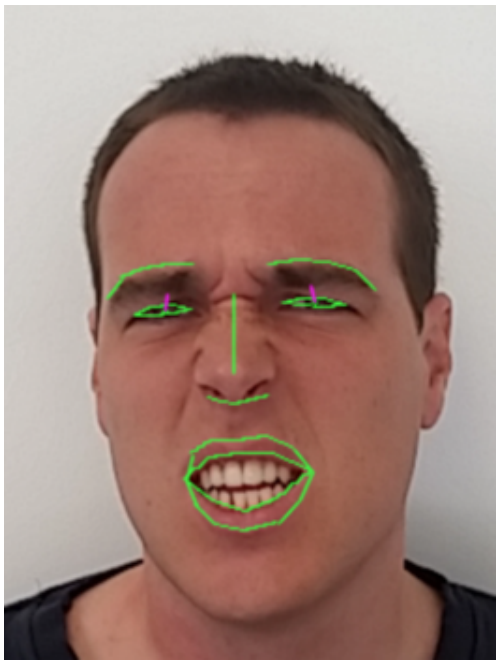

Рис. 7: Пример работы алгоритма разметки лица из IntraFace

Также широко известен метод, предложенный в статье [17]. Он основан на подгоне статистической модели формы лица к заданному изображению с помощью градиентного бустинга на регрессионных деревьях.

Ниже приведены примеры его работы, полученные с использованием библиотеки dlib, где данный подход реализован (показаны только контуры губ):

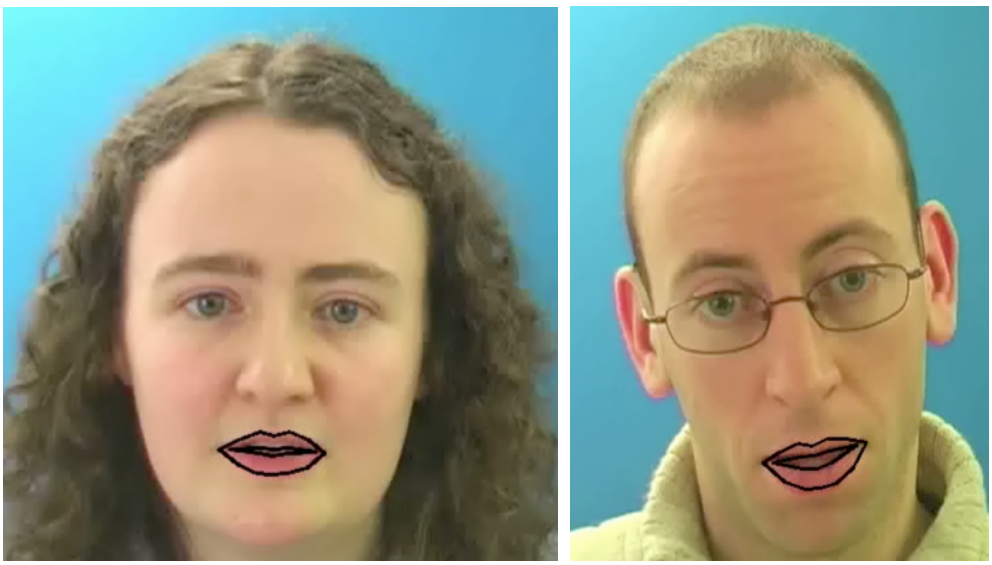

Рис. 8: Пример работы алгоритма разметки лица из dlib

В данной работе в качестве базового алгоритма выделения признаков был использован последний упомянутый подход, поскольку он по качеству сопоставим с лучшими решениями и при этом имеет реализацию с открытыми исходными кодами, что может позволить в будущем подгонять его под нужные задачи.

1.2.2. Дескрипторы, основанные на анализе значений пикселей

Ко второй группе можно отнести такие подходы как обучение извлечения вектора признаков из набора пикселей изображения с помощью нейронных сетей. Такие сети могут работать следующим образом [22]: на вход подаются значения пикселей изображения или региона интереса и на выходе ожидается вектор, в точности равный входному, при это скрытые слои содержат гораздо меньше нейронов, чем входной. В процессе обучения происходит настройка весов скрытых слоев, чтобы они выдавали компактное представление исходных данных, из которого можно с какой-то степень точности их восстановить. Преимуществом такого метода является отсутствие необходимости вручную решать какие признаки лучше всего опишут входные данные для конкретной задачи, придумывать различные ручные дескрипторы. Также такой подход

меньше зависит от способа подачи данных, например, можно вычислять признаки сразу для нескольких подряд идущих изображений из видео-потока.

Также стоит упомянуть про алгоритмы основанные на методы главных компонент. Он может использоваться для сжатия вектора признаков большой размерности (например, списка значений пикселей региона интереса [3]). В его основе лежит применение ортогонального преобразования к исходному пространству таким образом, чтобы выборочная дисперсия вдоль первой координаты была максимальна и далее выборочная дисперсия вдоль каждой последующей координаты максимальна, при условии ее ортогональности всем предыдущим координатам. Для применения подобных подходов на практике требуется предварительно вычислить матрицу ортогонвального преобразования и вектор средних по обучающей выборке. После этого работа с заданным тестовым изображением представляет из себя вычитание среднего вектора с последующим преобразованием в вектор маленькой размерности, который можно рассматривать как вектор признаков и использовать для дальнейшей классификации как компактное представление исходного изображения.

1.3. Эталонные выборки

Подавляющее количество свободно доступных датасетов с размеченными аудио- и видеозаписями доступно на английском языке. Среди них можно выделить несколько самых популярных:

- GRID [4]

  Содержит 33 говорящих, для каждого записано 1000 3-х секундных видео с произношением 6 слов из классов: команда (4 слова), цвет (4 слова), предлог (4 слова), буква (25 штук, без W), цифра (10 штук), наречие (4 штуки).

- CUAVE [2]

  36 говорящих, словарь из 10 цифр

- LILiR TwoTalk corpus [11]

  4 диалога по 12 минут между двумя людьми

- AVLetters1 [10]

  10 говорящих, по 3 повторения каждой буквы английского алфавита (всего 780 произношений)

В данной работе был использован датасет GRID.

2. Предложенный метод

2.1. Общее описание метода

Как будет подробно рассказано ниже, последовательность шагов по распознаванию множества сказанных слов на видеозаписи выглядит следующим образом:

1. Для каждого кадра считается вектор признаков
2. С помощью акустической модели для каждого кадра считается распределение вероятностей произнесенных визем
3. Сгенерированная последовательность вероятностей проходит через специальный вид рекуррентной нейронной сети (т.н. decoder LSTM) и кодируется в вектор фиксированной размерности
4. Полученный на предыдущем шаге вектор служит входным параметром для другой рекуррентной нейронной сети (т.н. encoder LSTM), которая последовательно генерирует метки выходных слов

2.2. Описание выбранной эталонной выборки

Как было сказано в разделе 1.3 для экспериментов был выбран датасет GRID, записанный работниками университет Шеффилда. Он содержит в себе записи речи 34 спикеров, по 1000 видео на человека с частотой 25 кадров в секунду. Каждая запись представляет из себя последовательность слов из искусственной грамматики вида

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

После двоеточия указано количество различных слов соответствующего типа. Пример предложения из указанной грамматики: ”bin blue by m three again”. Более подробно содержание записей описано в следующей таблице

Таблица 1: грамматика в GRID corpus

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

Также для каждой записи известен порядок слов, произнесенных на ней, вместе с начальным и конечным кадром каждого произношения. Поскольку некоторые слова (и буквы) длились всего несколько кадров, было решено принимать в рассмотрение только фразы, которые длятся не слишком мало. А именно, для каждого слова было подсчитано его средняя длина в кадрах и были оставлены для дальнейшей работы только следующие слова, длящиеся в среднем не меньше 4 кадров

Таблица 2: список выбранных слов

place	lay	now	red	white	green	please	again	blue
soon	by	with	set	bin	five	three	nine	eight
two	six	zero	four	one	seven			

2.3. Выделение признаков

В качестве базового метода извлечения признаков из видеопотока использовался метод, предложенный в статье [17]. Преимуществами данного подхода являются качество, сопоставимое с коммерческими аналогами и малое время работы, что позволяет получать вектора признаков из видео "на лету". К тому же этот алгоритм реализован в библиотеке с открытым кодом dlib, что добавляет возможность его модификации под возникающие задачи.

Таким образом, на выходе с каждой картинки получается последовательность из 68 точек (x, y), из которых мы оставляем 20 штук $(x_1, y_1, ..., x_{19}, y_{19})$, соответствующих контуру губ, как указано на рисунке.

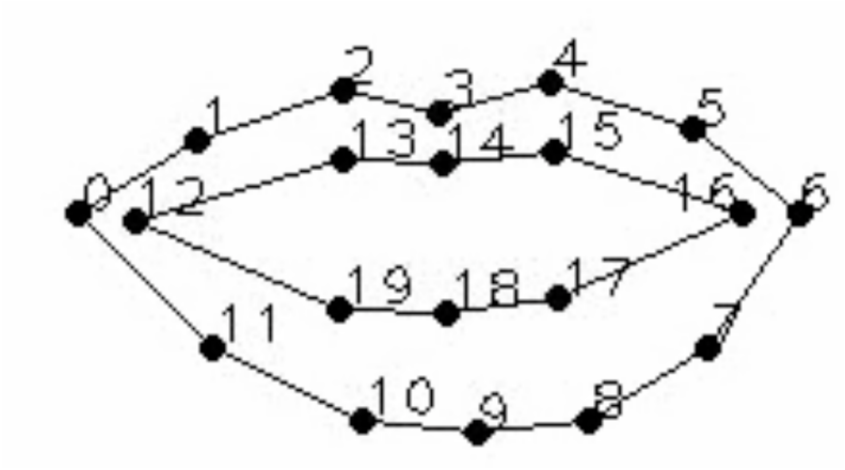

Рис. 9: Ключевые точки на губах

Затем происходит нормализация полученного вектора. Этот шаг необходим, поскольку при сдвиге лица на кадре или его повороте вектор признаков должен меняться как можно меньше. Но при этом задача восстановления начального положения обьекта (в рассматриваемом случае - губ) в общем случае представляется довольно сложной задачей, поэтому был использован аналог простого метода нормализации, предложенного в статье [32]. Он заключается в переносе координат уголков губ в точки с координатами $(-1, 0)$ и $(1, 0)$ соответственно. Более подробно, вычисляется средняя точка между уголками рта по формуле

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

Можно получить угол поворота и радиус

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

После этого нормализованные координаты контура можно получить следующим образом

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

Таким образом, для последовательности из 75 кадров для каждой записи из рассматриваемого датасета, можно посчитать последовательность векторов признаков размерности 40. Кроме того, данные были линейно экстраполированы с 25 fps до 100 fps (т.е. теперь в одной записи после этого содержится 300 кадров). Такое преобразование является широко известным и применяется во многих работах (например [34]). Кроме того, такой шаг кажется полезным, потому что 25 кадров в секунду - это немного, а модели в распознавании речи (в том числе по звуковым данным) обычно работают с данными большей частоты. Например, очень широко используемые в анализе речи мел-кепстральные коэффициенты (mel-frequency cepstral coefficients) и PLP cepstral coefficients обычно считаются на окне шириной 25 миллисекунд, с шагом 10 миллисекунд по звуковому сигналу. Другими словами, на выходе получается последовательность частотой 100 fps, что совпадает с количеством векторов признаков после проведенной экстраполяции.

Как будет показано далее, в предложенном подходе можно добиться незначительного улучшения путем применения экспоненциального сглаживания второго порядка к последовательности векторов в пределах одной записи. Более подробно, алгоритм экспоненциального сглаживания первого порядка временного ряда $\{u_t\}_{t=1}^{t=T}$ представляет из себя

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

Можно заметить, что алгоритм представляет из себя следующее: новый элемент сглаженной последовательности $u_t^{(smooth)}$ получается прибавлением к предыдущему элементу $u_{t-1}^{(smooth)}$ его разности с текущим элементом последовательности $(u_t - u_{t-1}^{(smooth)})$, домноженной на α. Таким образом, в каком-то смысле сглаживается первая "производная" последовательности. Такое преобразование, довольно долго реагирует на резкие изменения в поведении $\{u_t\}$, поэтому в данной работе было использовано более чуствительное к таким колебаниям сглаживание второго порядка

$$s_1 = u_1$$
$$b_1 = 0$$
$$s_t = \alpha u_t + (1-\alpha)(s_{t-1} + b_{t-1})$$

$$b_t = \beta(s_t - s_{t-1}) + (1 - \beta)b_{t-1}$$
$$u_t^{(smooth)} = s_t + b_t$$

В экспериментах использовались константы $\alpha = 0.95, \beta = 0.1$. Несмотря на его простоту такой метод позволил немного улучшить результаты.

2.4. Доразметка обучающей выборки

Поскольку распознавание предполагалось проводить в 2 этапа - распознавание последовательности визем (акустическая модель), затем непосредственно распознавание слов, то требуется разметка обучающей выборки по виземам. Для этого требуется ее распределение вероятностей фонем на каждом кадре, имея которую можно получить требуемую разметку, воспользовавшись любой таблицей, указывающей на возможное соответствие между фонемами и виземами. Таким образом, появилась задача по имеющимся аудиозаписям для рассматриваемого датасета GRID corpus получить распределение фонем с помощью какой-либо аудио-модели.

Для обучения данной модели был выбран стандартный датасет TIMIT [16]. В качестве признаков использовались логарифмы от т.н. filter-bank features. Они вычислялись, как уже было упомянуто в разделе 2.3, на окне длиной 25 миллисекунд подряд идущих измерений сигнала и были вычислены с шагом 10 миллисекунд. Кроме того, окончательный вектор признаков для фрейма составлялся его конкатенацией векторов признаков размерности 40 с 5 предыдущих фреймов и 10 следующих. Таким образом вектор признаков имел размерность $40 \cdot (5 + 1 + 10) = 640$ чисел. Затем обучалась нейронная сеть со следующей архитектурой

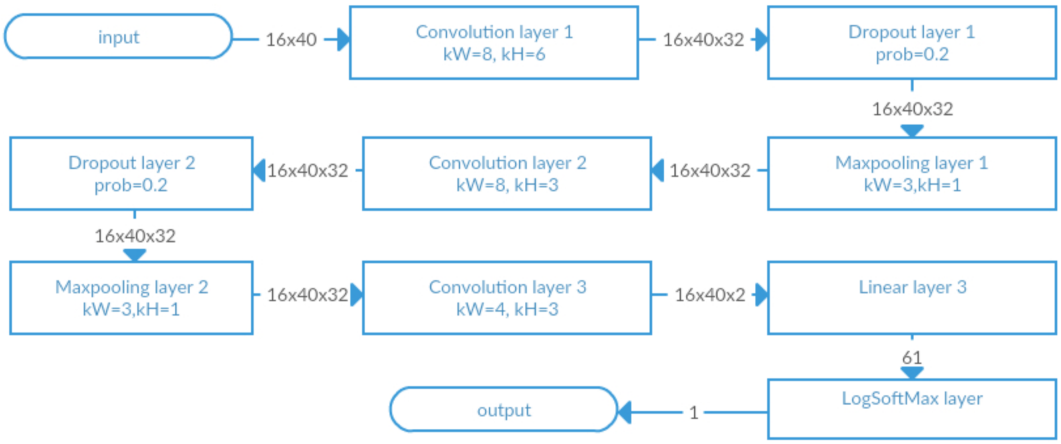

Рис. 10: Архитектура сети для обучения аудио-модели

Здесь на стрелках подписана размерность данных, полученных после применения соответствующего слоя. Описание принципа работы обозначенных слоев можно найти в разделе 1.1.2. Заметим, что сверточные и max-pooling слои используются не только при анализе изображений, но и в задачах распознавания речи (например, [29], [21]). Выходом сети служило распределение вероятностей 61 фонемы, список которых приведен ниже

Таблица 3: список фонем

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

Точность такой работы такой модели на выборке из TIMIT (80% - тренировочное множество, 20% - тестовое), если из распределения выбирать фонему с максимальной вероятностью, получилась 71%.

Следующим шагов является преобразование вектора вероятностей фонем в каждый момент времени в соответствующий ему вектор вероятностей визем. Это было сделано с помощью расширенной на 61 фонемы таблицы приведенной в [1]

Таблица 4: соответствие визем и фонем

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

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

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

Вероятность одной из 14 визем считается как сумма вероятностей соответствующих ей фонем.

2.5. Построение акустической модели

Модель, предсказывающая распределение вероятностей визем на каждом фрейме представляла из себя нейронную сеть с несколькими линейными сло-

ями с активационными функциями, возможно, со dropout-слоями между ними. Таким образом в нашем случае модель задавалась выходной размерностью каждого линейного слоя, активационной функцией после каждого слоя и вероятностью в каждом dropout слое. Также, по аналогии с методом, использованным в предыдущем разделе, вектор признаков для кадра считался конкатенацией векторов признаков на каждом фрейме в пределах определенного окна. Однако в нашем случае можно более аккуратно использовать вектор признаков, ведь формы губ на соседних кадрах очень сильно коррелируют между собой и логичным шагом кажется сжать сконкатенированный вектор в несколько раз, одновременно избавившись от ненужного шума, лишней информации и увеличив скорость обучения за счет меньшей размерности входных данных. Сжатие вектора осуществлялось с помощью метода главных компонент. В этом подходе требуется выбрать сколько компонент оставить в качестве признаков и, соответственно, какие компоненты не принимать во внимание. В качестве ориентира для этого параметра использовалось правило, заключающееся в том, что остаются только компоненты с соответствующим собственным числом большим среднего среди всех собственных чисел (т.н. Kaiser rule, более подробно в [13]). Также, были проведены эксперименты с применением экспоненциального сглаживания первого, второго порядка.

Пример сети, использованной в экспериментах

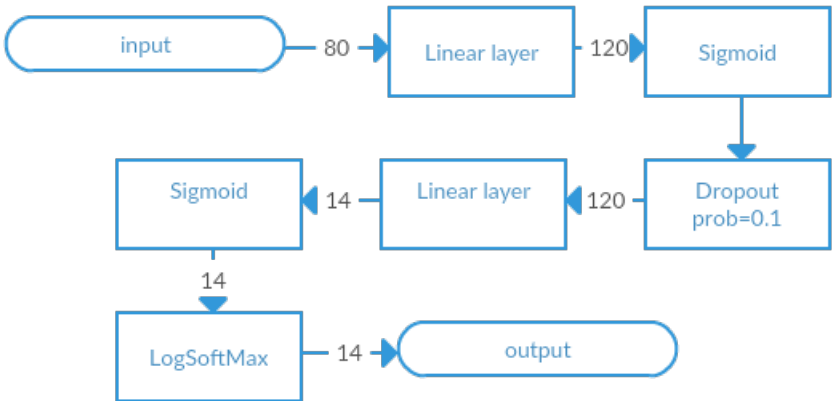

Рис. 11: Одна из возможных сетей в акустической модели

В качестве тренировочной выборки использовалось множество из 15 спикеров, а тестовой выборке содержалось 3 спикера (не из числа первых пятнадцати). В таблице приведены результаты некоторых экспериментов. Описание модели состоит из размера окна, которое использовалось для извлечения признаков, количества компонент после сжатия размерности (применения pca - principal component analysis), описания структуры сети, какие еще преобразования были применены к векторам признаков. Например, первая запись означает, что в качестве вектор признаков рассматривались сконкатенированные вектора из 10 предыдущих и 5 следующих фреймов и модель представляла из себя 3 линейных слоя с сигмоидными активационными функциями и выходными размерностями 128, 64, 14 соответственно. Также указано, является ли выбранное количество компонент оптимальным по указанному выше правилу. Во всех указанных экспериментах был использован размер батча 128, метод обучения adam с начальным learning rate 0.001 (данные параметры показывали лучшие результаты на большинстве архитектур сетей)

Таблица 5: некоторые результаты обучения акустической модели

Описание модели	Точность распознавания визем
Window 10-5, pca 72(opt) 128(sigmoid)+64(sigmoid)+14(sigmoid)	54%
Window 10-5, pca 72(opt) exp smoothing first order (α=0.97) 128(sigmoid)+64(sigmoid)+14(sigmoid)	52%
Window 20-10, pca 80(opt) exp smoothing second order (α=0.95,β=0.1) 140(sigmoid)+80(sigmoid)+14(sigmoid)	55%
Window 15-10, pca 75(opt) exp smoothing second order (α=0.95,β=0.1) 110(relu)+80(relu)+14(relu)	56%
Window 15-10, pca 75(opt) exp smoothing second order (α=0.95,β=0.1) 110(relu,dropout=0.1)+ 80(relu,dropout=0.15)+14(relu)	58%

Таким образом, лучшая модель имела точность 58%, она и использовалась в качестве первого шага в распознавании. Заметим, что такая точность представляет из себя качество распознавания разметки, которая была получена аудио-моделью с собственной точностью 71%, поэтому такой результат может оказаться лучше, чем кажется на первый взгляд.

2.6. Модель, распознающая произнесенное слово на каждом кадре

В качестве следующего шага было решено попробовать рекурентную нейронную сеть, а именно LSTM, про который было рассказано в разделе 1.1.2. Более подробно структуру слоя LSTM, изображенного на рисунке 6, можно описать как

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t \cdot \tanh(c_t)$$

Слои называются соответственно с их главной задачей: f_t - forget gate, i_t - input gate, c_t - cell state, o_t - output gate, h_t - hidden state. В данном случае сеть состояла из одного слоя LSTM, куда последовательно подавались вектора распределения вероятностей 14 визем на каждом фрейме. В качестве выхода ожидалась метка класса слова, звучащего в данный момент времени.

Результаты некоторых экспериментов приведены в таблице. В ней в качестве оценки качество приведена точность распознавания принадлежности фрейма правильному слову из множества выбранных слов. Также указан размер скрытого слоя в LSTM и вероятность dropout слоя

Таблица 6: некоторые результаты с использованием LSTM

Описание модели	Точность распознавания слов
LSTM hiddenSize=100	49%
LSTM hiddenSize=120	48%
LSTM hiddenSize=60	46%
LSTM hiddenSize=120, dropout=0.2	49%
LSTM hiddenSize=120, dropout=0.2 mark only 2nd half of word	48%

В таком подходе есть такой недостаток, что при разметке каждого фрейма, принадлежащего слову соответствующим классом в том числе размечаются и самые первые фреймы. Но при использовании рекуррентной нейронной сети происходит последовательная подача входных данных с каждого кадра, поэтому ожидать, что модель научится правильно определять слово уже по первым фреймам, содержащим его, не стоит. Более того, такая разметка начала слова в каком-то смысле приводит сеть в заблуждение, поскольку мы все-таки от нее ожидаем, что она как раз сможет угадывать слово уже по первым фреймам. Поэтому была предпринята попытка размечать соответствующей меткой класса только вторую половину каждого слова, но такой метод не дал улучшения. Возможно, это связано с тем, что мы все еще вводим сеть неправильно

сообщаем сети, что от нее нужно - ведь первая часть слова все-таки в нему относится, а размечена как что-то ни к чему не относящее, точно также как отрезки, где вообще ничего не произносится.

2.7. Модель, распознающая короткие последовательности слов

Проблемы, описанные в предыдущем разделе могут быть частично решены, если разрешить нейронной сети сначала пропустить через себя всю запись (последовательность векторов признаков, соответствующих конкретной видеозаписи) и уже потом предсказывать множество произнесенных слов. Такая задача очень похожа на задача sequence-to-sequence learning, т.е. преобразование из последовательности в последовательность. Сложность заключается в том, что нейронные сети принимают на вход и выдают на выходе вектор фиксированного размера. Классические рекуррентные нейронные сети принимают на вход последоватльеность произвольной длины, но на выходе все еще выдают вектор фиксированного размера. Одним из возможных решений является архиктура т.н. encoder-decoder LSTM [26].Главной идеей этого метода является сначала получение какого-то вектора фиксированного размера, описывающего входную последовательность, а затем его разворачивание уже в выходную последовательность. Более подробно, обычно процесс происходит следующим образом: вход пропускается через рекуррентную нейронную сеть (в нашем случае со слоем LSTM), и cell state (вместо в hidden state) после прохождения всей последовательности считается вектором, описывающим вход. Этот этап кодирует данные в вектор, поэтому соответствующая нейронная сеть называется encoder. На втором этапе (decoding) стоит задача по вектору понять выходную последовательность. Для этого полученный cell state считается начальным состоянием c_t (тоже с h_t) в следующей сети с использованием LSTM, которая последовательно генерирует символы из выходного алфавита пока не сгенерирует специальный терминальный символ. Часто такая архитектура используется в задачах машинного перевода, поэтому здесь было введено понятие выходного алфавита, также в классическом варианте символ, сгенерированный в момент времени t подается на вход сети в следую-

щий момент времени $t+1$. Но в рассматриваемой в работе задаче, в этом нет необходимости, выходной алфавит (закодированное множество слов) имеет небольшой обьем, и символы полученного алфавита (который в свою очередь является множеством слов уже над английским алфавитом) не образуют между собой никаких связей. Поэтому использовалась описанная архитектура без передачи сгенерированного символа (т.е. слова) дальше в процессе работы decoder-LSTM.

Описанную структуру можно увидеть на рисунке

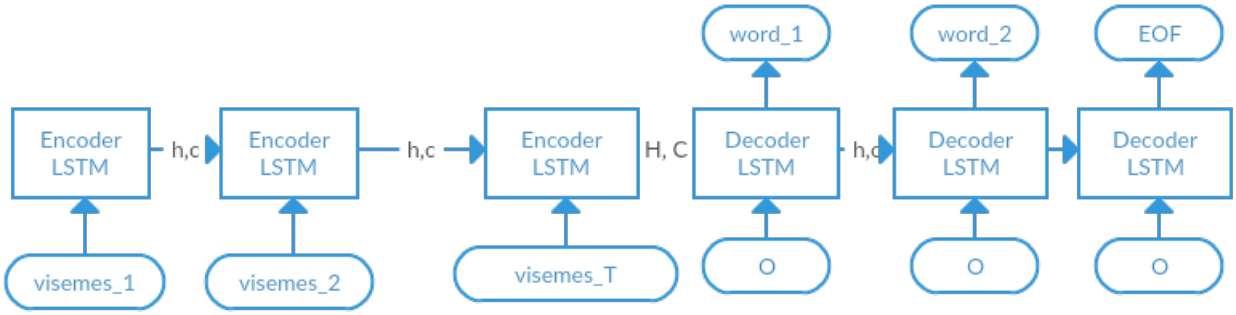

Рис. 12: Модель encoder-decoder LSTM

В качестве выхода ожидалась последовательность меток слов, произнесенных на данной видеозаписи. Слишком короткие слова помечались как пустое слово, но не выкидывались полностью. Фактически, это означает, что все короткие слова оказались обьединенными в один большой класс.

В качестве тренировочной выборки было использовано множестве из 30 спикеров, в то время как в тестовой находилось 5 спикеров. При этом записи одного и того же спикера не могли находиться одновременно в обучающей и тестовой выборках.

В таблице приведены некоторые показательные результаты экспериментов. Записи про dropout слой с заданной вероятностью и про размер скрытого слоя означают их параметры на соответствующем этапе (encoding или decoding)

Таблица 7: некоторые результаты с архитектурой encoder-decoder LSTM

Описание модели	Точность распознавания слов
encHiddenSize=128 decHiddenSize=50	73%
encHiddenSize=100 decHiddenSize=40	72%
encHiddenSize=110, dropout=0.2 decHiddenSize=60, dropout=0.1	75%
encHiddenSize=128, dropout=0.2 decHiddenSize=40	76%

Таким образом, лучшей точностью, которой удалось добиться является 76%. Также стоит заметить, что на подавляющей большинстве записей (97%) правильно определилось количество слов, на остальных же видео все распознанные вхождения слов считались ошибочными, даже если какие-то из них совпали с правильными разметкой.

Кроме того, если в качестве рассматриваемого множества слов использовать все доступные слова, без выкидывания самых коротких по продолжительности, то точность последней модели на длинных словах понизится до 55%, а качество работы на коротких фразах будет составлять 23%.

3. Используемые технологии

Работа непосредственно с видеоданными осуществлялась на языке C++, на нем же осуществлялось вычисление признаков с использованием библиотеки dlib и opencv3. При работе с аудио-данными и извлечении признаков из аудио использовался язык python. Обучение нейронных сетей происходило с помощью библиотеки torch, написанной на C и использующей при работе язык lua. Почти всегда данные хранились в формате hdf5, поскольку с ним можно работать как из python так и из lua, то это позволяло без проблем извлекать признаки на одном языке, а использовать их на другом. Также почти все процессы обучения происходили на видеокарте с использованием CUDA, что позволяло ускорить работу с данными и обучение на порядок.

4. Заключение

4.1. Результаты

В рамках данной дипломной работы были поставлены и решены следующие задачи:

- Проведен обзор существующих методов в распознавании речи
- Выделены и проанализированы основные этапы работы алгоритмов распознавания речи
- Предложен метод преобразования речи в последовательность слов, основанный только на визуальной информации
- Проведен ряд экспериментов, сравнивающих варианты предложенного метода между собой и показывающих применимость предложенного подхода

4.2. Сравнение с другими работами

В большинстве работ посвященных распознаванию речи по видеозаписи или чтению по губам рассматривается свой датасет, часто с сильно ограниченным набором слов или даже собственноручно записанный. К тому анализ речи по визуальным признакам гораздо более чуствителен к смене спикера, различным наборам слов, акцентам. Также очень часто авторы приводят качество своих моделей, обученных на аудио и визуальных признаках вместе, чтобы сравнить с моделями, анализирующими только аудио. Поэтому сравнивать работы между собой сложнее, чем, например, при работе со звуковой информацией. Все же существуют работы, которые работали с тем же датасетом GRID corpus и анализировали возможность чтения по губам. Например, в работе [31] авторам удалось добиться точности 79%, используя только видеопризнаки. Однако там все эксперименты были speaker-dependent, то есть тестовая и тренировочная выборки всегда брались с одного и того же спикера, в отличие от данной работы.

Список литературы

[1] Benedikt Lanthao. Facial Motion: a novel biometric? — 2010.

[2] CUAVE: A new audio-visual database for multimodal human-computer interface research / E. K. Patterson, S. Gurbuz, Z. Tufekci, J. N. Gowdy // In Proc. ICASSP. — 2002. — P. 2017–2020.

[3] Christoph Bregler Yochai Konig. "EIGENLIPS" FOR ROBUST SPEECH RECOGNITION. — 1994.

[4] Cookea Martin, Jon Barker Stuart Cunningham Xu Shao. 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.

[5] Cootes T. F., Edwards G., Taylor C.J. Comparing Active Shape Models with Active Appearance Models. — 1999. — P. 173–182.

[6] Dropout: A Simple Way to Prevent Neural Networks from Overfitting / Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky et al. // Journal of Machine Learning Research. — 2014. — Vol. 15. — P. 1929–1958. — URL: http://jmlr.org/papers/v15/srivastava14a.html.

[7] Exploring the Limits of Language Modeling / Rafal Józefowicz, Oriol Vinyals, Mike Schuster et al. // CoRR. — 2016. — Vol. abs/1602.02410. — URL: http://arxiv.org/abs/1602.02410.

[8] Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition / Hasim Sak, Andrew W. Senior, Kanishka Rao, Françoise Beaufays // CoRR. — 2015. — Vol. abs/1507.06947. — URL: http://arxiv.org/abs/1507.06947.

[9] Hassanat Ahmad Basheer. Visual Passwords Using Automatic Lip Reading. — 2014. — Vol. abs/1409.0924. — URL: http://arxiv.org/abs/1409.0924.

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

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

[12] Ioffe Sergey, Szegedy Christian. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift // CoRR. — 2015. — Vol. abs/1502.03167. — URL: http://arxiv.org/abs/1502.03167.

[13] Jackson Donald A. Stopping rules in principal component analysis: a comparison of heuristical and statistical approaches. — 1993.

[14] Jacobs David. Correlation and Convolution // Class Notes for CMSC 426. — 2016. — URL: http://www.cs.umd.edu/~djacobs/CMSC426/Convolution.pdf.

[15] Jendoubi Siwar, Yaghlane Boutheina Ben, Martin Arnaud. Belief Hidden Markov Model for speech recognition // CoRR. — 2015. — Vol. abs/1501.05530. — URL: http://arxiv.org/abs/1501.05530.

[16] John Garofolo Lori Lamel William Fisher Jonathan Fiscus David Pallett Nancy Dahlgren Victor Zue. TIMIT Acoustic-Phonetic Continuous Speech Corpus. — URL: https://catalog.ldc.upenn.edu/LDC93S1.

[17] Kazemi Vahid, Sullivan Josephine. One Millisecond Face Alignment with an Ensemble of Regression Trees // CVPR. — 2014.

[18] Le Thai Hoang, Vo Truong Nhat. Face Alignment Using Active Shape Model And Support Vector Machine // CoRR. — 2012. — Vol. abs/1209.6151. — URL: http://arxiv.org/abs/1209.6151.

[19] Mark Gales Steve Young. The Application of Hidden Markov Models in Speech Recognition. — 2008. — URL: http://mi.eng.cam.ac.uk/~mjfg/mjfg_NOW.pdf.

[20] Mehryar Mohri Fernando Pereira Michael Riley. Weighted Finite-State Transducers in Speech Recognition. — 2001. — URL: http://www.cs.nyu.edu/~mohri/pub/csl01.pdf.

[21] Ossama Abdel-Hamid Abdel-rahman Mohamed Hui Jiang Gerald Penn. Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition.

[22] Paleček Karel. Extraction of Features for Lip-reading Using Autoencoders. — 2005.

[23] Sak Hasim, Senior Andrew W., Beaufays Françoise. 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.

[24] Sak Hasim, Senior Andrew W., Beaufays Françoise. Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition // CoRR. — 2014. — Vol. abs/1402.1128. — URL: http://arxiv.org/abs/1402.1128.

[25] Sridharan Ramesh. HMMs and the forward-backward algorithm. — 2010. — URL: http://people.csail.mit.edu/rameshvs/content/hmms.pdf.

[26] Sutskever Ilya, Vinyals Oriol, Le Quoc V. Sequence to Sequence Learning with Neural Networks // CoRR. — 2014. — Vol. abs/1409.3215. — URL: http://arxiv.org/abs/1409.3215.

[27] Timothy F. Cootes Gareth J. Edwards Christopher J. Taylor. Active Appearance Models. — 2008. — URL: http://www.comp.hkbu.edu.hk/~ymc/papers/conference/cis08_publication_version.pdf.

[28] Timothy F. Cootes Gareth J. Edwards Christopher J. Taylor. Improving Visual Features for Lip-reading. — 20111. — URL: https://pdfs.semanticscholar.org/6778/68449c6b05a3df45d25a18f9782550b69661.pdf.

[29] Toth Laszlo. Convolutional Deep Maxout Networks for Phone Recognition. — URL: https://pdfs.semanticscholar.org/0a24/5098455a6663f922a83d318f7b61d357ab1f.pdf.

[30] Virginia Estellers Jean-Philippe Thiran. Multi-pose lipreading and Audio-Visual Speech Recognition. — 2012. — URL: http://vision.ucla.edu/~virginia/publications/Estelle2012EURASIP.pdf.

[31] Wand Michael, Koutník Jan, Schmidhuber Jürgen. Lipreading with Long Short-Term Memory // CoRR. — 2016. — Vol. abs/1601.08188. — URL: http://arxiv.org/abs/1601.08188.

[32] Wang S. L., Lau W. H., Leung S. H. Automatic Lip Contour Extraction from Color Images // Pattern Recogn. — 2004. — . — Vol. 37, no. 12. — P. 2375–2387. — URL: http://dx.doi.org/10.1016/j.patcog.2004.04.016.

[33] Xiong Xuehan, la Torre Fernando De. Supervised Descent Method and its Applications to Face Alignment. — 2012.

[34] Yuxuan Lan Richard Harvey, Theobald Barry-John. Insights into machine lip reading. — 2012. — URL: https://pdfs.semanticscholar.org/c573/c71213b46a2b966546c7b7848b5bbe0536ec.pdf.