Санкт-Петербургский государственный университет
Кафедра технологии программирования

Малых Егор Андреевич

Выпускная квалификационная работа бакалавра

Применение глубоких нейронных сетей к задаче текстонезависимой идентификации диктора по голосу

Направление 010400
Прикладная математика и информатика

Научный руководитель,
старший преподаватель
Малинин К. А.

Санкт-Петербург
2017

Содержание

Введение

Задача идентификации диктора по голосу, носящая в англоязычной литературе название «speaker identification task», позволяет определить по записи голоса его принадлежность определённому диктору. Другими словами, она отвечает на вопрос «Кто это говорит?». Умение отвечать на подобный вопрос открывает дорогу к решению множества прикладных задач из различных областей человеческой деятельности. Среди таких задач можно выделить следующие.

1. **Поиск определённого диктора в потоке голосовых данных.**

   Эта задача может возникнуть, например, в сфере поддержки, когда необходимо среди записей телефонных разговоров call-центра найти записи всех диалогов с недавно звонившим клиентом с целью анализа и улучшения качества работы центра. С применением средств идентификации диктора по голосу такое возможно даже если клиент звонил с телефонов с разными номерами. Аналогичная задача возникает и в сфере безопасности, где поиск записей телефонных разговоров потенциально опасных личностей может производиться среди всевозможных записей определённой телефонной станции.

2. **Биометрическая аутентификация по голосу**

   Путём сравнения текущего диктора со списком заранее заданных автоматическая система может принимать решение о разрешении или запрете авторизации. Такой способ аутентификации может быть использован как замена или дополнение к паролю при разблокировке смартфона или при попытке доступа к банковскому приложению. Благодаря средствам верификации диктора по голосу никогда не слышавший прежде своего собеседника пользователь сможет в автоматическом режиме удостовериться, что собеседник не выдаёт себя за другую личность, а автоматическая система, распознающая и исполняющая голосовые команды, сможет удостовериться, что выполняет команды авторизованного пользователя.

В современном мире потоки информации достигли объёмов, не подвластных ручному анализу. Именно поэтому внимание исследователей скон-

центрировано в основном на автоматических подходах к решению задач. Исключением не является и данная работа.

Подходы, основанные на классических методах машинного обучения и статистики, долгое время оставались и остаются главенствующими при решении задачи автоматической идентификации диктора по голосу. В то же время, активно развивающиеся в последнее десятилетие подходы, основанные на глубоких нейронных сетях, достигли непревзойдённых успехов во многих задачах классификации, распознавания образов, идентификации по лицу. Преимущества подобных подходов очевидны: они просты в разработке и использовании, требуют минимального количества вносимой извне априорной информации и зачастую превосходят традиционные методы по качеству.

В данной работе рассматривается возможность применения глубоких нейронных сетей к задаче автоматической идентификации диктора по голосу в текстозависимых и текстонезависимых условиях, исследуются преимущества и недостатки подобного подхода и проводится сравнение с классическим методом, показывающим лучшие результаты на рассматриваемых базах.

Постановка задачи

Задача автоматической идентификации диктора по голосу сводится к задаче построения идентификатора

$$F : \mathcal{P} \to \mathbb{R}^n,$$

который произвольному произнесению p из пространства всевозможных произнесений $\mathcal{P}$ сопоставит вектор высокоуровневых признаков некоторой заранее определённой размерности n, и функции

$$h : \mathbb{R}^n \times \mathbb{R}^n \to [0; 1],$$

для любых двух векторов признаков возвращающей вероятность принадлежности оригинальных произнесений одному и тому же диктору. Таким образом, **задача** данной работы заключается исследовании возможности применения в качестве функции F глубокой свёрточной нейронной сети при фиксированной h в текстозависимых и текстонезависимых условиях. **Целями** работы являются:

1. Обучение в качестве F классической системы, основанной на i-векторах.
2. Обучение в качестве F системы, основанной на глубоких свёрточных нейронных сетях.
3. Сравнение результатов и вывод о возможности применения подходов глубокого обучения к рассматриваемой задаче.

Обзор литературы

Системы на основе i-векторов долгое время были и являются по сей день лучшими системами для задачи автоматической текстонезависимой идентификации диктора по голосу [20–23]. Однако недавно эта задача стала рассматриваться исследователями с позиции методов глубокого обучения. К примеру, глубокая нейронная сеть, построенная и обученная для задачи автоматического распознавания речи [22, 24], позволяет разделить акустическое пространство на классы синонов для того чтобы затем дикторы могли быть разделены в этом пространстве классической TV моделью (total variability) [20]. Два основных подхода могут быть выделены в подобных глубоких фонетико-дискриминативных моделях. Первый из них заключается в использовании постериорных вероятностей, извлечённых с помощью глубокой нейронной сети, для подсчёта статистик Баума-Велша. Второй подход, считающийся более устойчивым к изменяющимся акустическим условиям [25], заключается в использовании промежуточного представления диктора, извлечённого из некоторого скрытого слоя глубокой сети, вместе с дикторо-специфичными признаками, такими как MFCC, для обучения полной TV-UBM модели. Успех подобного подхода в текстонезависимой задаче приводит к попыткам его использования и в текстозависимых условиях [26–30]

Параллельно с этим, существуют исследования подходов глубинного обучения к задаче автоматической идентификации диктора в текстозависимой задаче, ставящие своей целью создание цельной глубокой системы, сопоставляющей высокоуровневые дикторские признаки непосредственно низкоуровневому представлению сигнала [31–33]. В этих работах рассматриваются небольшие произнесения длительностью до пяти секунд, а в качестве признаков используются низкоразмерные MFCC или банки фильтров.

В данной работе исследуется возможность применения глубоких моделей к текстозависимой и текстонезависимой задачам автоматической идентификации диктора. В отличии от [31–33] рассматриваются более длинные произнесения, а в качестве признаков используются необработанные спектрограммы.

Глава 1. Подготовка данных

1.1 Описание используемых баз

1.1.1 RSR2015

База Robust Speaker Recognition 2015 (RSR2015) [2] является текстозависимой речевой базой, содержащей 197100 произнесений, относящихся к 300 дикторам. Каждому диктору принадлежит 657 прознесений, которые разделены на 9 сессий по 73 произнесения в каждой. Три сессии являются эталонными — записанными при регистрации пользователя, а ещё шесть сессий — верификационными, т. е. записанными при попытке авторизации пользователя. Для получения каждой сессии использовались как минимум три различных из следующих шести портативных устройств:

- 1 Samsung Nexus,
- 2 Samsung Galaxy S,
- 2 Samsung Tab,
- 1 HTC Desire.

В целях сбора базы авторами было разработано приложение для ОС Android, отображающее фразу, которую необходимо произнести и записывающее звук с микрофона устройства, пока диктором нажата соответствующая кнопка. Записи предоставлены в сыром виде в PCM-формате с частотой дискретизации 16 kHz и 16-битной глубиной.

База RSR2015 собрана в Сингапуре и распределение по этнической принадлежности и полу, представленное на рисунке 1, отражает структуру этой республики. Как видно из статистики, база сбалансирована по количеству мужчин и женщин. Возраст дикторов варьируется от 17 до 42 лет.

Так как база RSR2015 является текстозавсисимой, её лексическая вариативность ограничена. Для каждой сессии диктор произносит тридцать коротких фраз, выбранных из базы TIMIT [3] так, чтобы эти фразы покрывали все фонемы английского языка. Число слов в различных фразах варьируется от 4 до 8. Как ранее было описано, для записи пользователю необходимо нажать и удерживать кнопку, поэтому все произнесения содержат отрезки тишины случайного размера перед началом и после окончания

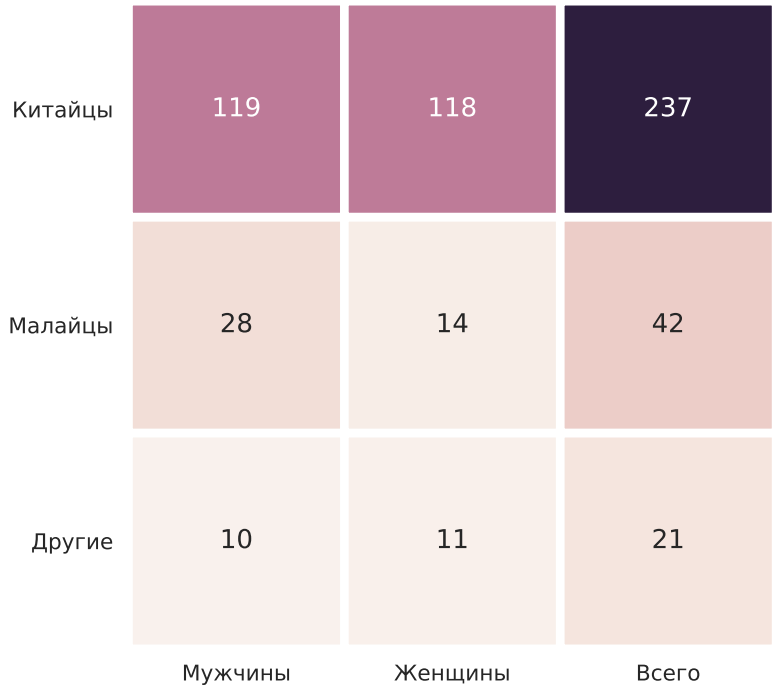

Рис. 1: Распределение дикторов в базе RSR2015

произнесения фразы. Таким образом, средняя длина произнесения составляет 3.2 секунды, максимальная длина — 10 секунд.

Три непересекающихся множества образуют базу RSR2015:

1. **Background множество** (50 мужчин, 47 женщин). Предназначено для обучения модели.

2. **Development множество** (50 мужчин, 47 женщин). Предназначено для настройки модели.

3. **Evaluation множество** (57 мужчин,49 женщин). Предназначено для оценки качества модели.

Для некоторых экспериментов, проведённых в рамкой данной работы, было подготовлено **расширенное background множество**, представляющее собой объединение **background** и **development** множеств и насчитывающее 100 мужчин и 94 женщины в качестве дикторов.

Благодаря ограниченным дикторской и лингвистической вариативностям четыре типа сравнений возможны для базы RSR2015:

1. **Одинаковый диктор + одинаковая фраза**.
2. **Одинаковый диктор + разные фразы**.
3. **Разные дикторы + одинаковая фраза**.
4. **Разные дикторы + разные фразы**.

Сравнения типа 1 являются *target*-сравнениями (должны распознаваться системой как успешная попытка авторизации), а сравнения 2, 3 и 4 — *imposter*-сравнениями (должны отклоняться системой). В рамках данной работы все четыре типа сравнений используются для проверки моделей, поэтому можно говорить о таком сценарии использования, когда пользователь для авторизации произносит произвольую фразу из множества заранее определённых.

1.1.2 NIST

Вторая используемая в данной работе база является текстонезависимой. Она представляет собой объединение семи баз, предоставляемых национальным институтом стандартов и технологий (National Institute of Standards and Technology, NIST) [4] для проведения международных соревнований по обучению систем текстонезависимой идентификации диктора за 1998, 2000, 2002, 2004, 2006, 2008 и 2010 года.

Фонограммы для базы NIST записаны в трёх различных условиях:

- Телефон.
- Гарнитура (близкий к диктору микрофон).
- Микрофон (отдалённый от диктора).

База содержит записи с речью на десяти разных языках и, как и RSR2015, является хорошо сбалансированной относительно пола диктора. Однако, в целях упрощения задачи и уменьшения количества обучающих примеров, было использовано подмножество описываемой базы, содержащее только фонограммы на английском языке с дикторами мужчинами. Полученное подмножество состоит из 22042 произнесений, принадлежащих 2001 диктору.

Для тестирования моделей, обученных на данной базе, использовалось тестовое множество C2 базы NIST за 2012 год, а точнее только то его подмножество, что удовлетворяет описанным выше требованиям для обучающего множества.

Лингвистическая вариативность рассматриваемой базы крайне велика: не существует двух фонограмм, содержащих одинаковый текст. Каждый диктор произносит произвольный текст на протяжении произвольного количества времени.

Для приведения базы NIST в соответствии с базой RSR2015, а так же для ускорения экспериментов, каждая фонограмма подверглась обрезке до 10 секунд после применения VAD (см. 1.2). При этом не играет большой роли тот факт, что лингвистическая информация могла потеряться или исказиться (например, обрезкой записи на середине произнесения слова), так как такая информация не существенна для текстонезависимой идентификации диктора.

1.2 Предварительная обработка сигнала

Исходные фонограммы наряду с речевыми сегментами содержат и участки, на которых речи нет. Этим участкам соответствуют например паузы между словами, паузы в речи, кашель и т. п. Пример изображён на рисунке 2. От подобных фрагментов важно избавиться ещё на этапе построения признаков, так как они не содержат полезной для распознавания диктора информации, но при этом увеличивают размер входных признаков по временной оси. Подсистема, ответственная за выделение речевых сегментов на спектрограмме называется Voice Activity Detector (VAD).

Простейший VAD может опираться на энергию сигнала как на главный критерий для выделения тихих участков, однако такой подход не сработает, если сигнал содежит, например, аддитивный или фоновый шумы. Более сложные модели могут быть построены с использованием тех же признаков, какие используются для распознавания диктора. В данной работе признаками для VAD служили 19 коэффициентов MFCC (см. 1.3.2), построенные с окном размером 25 миллисекунд и шагом 10 миллисекунд,

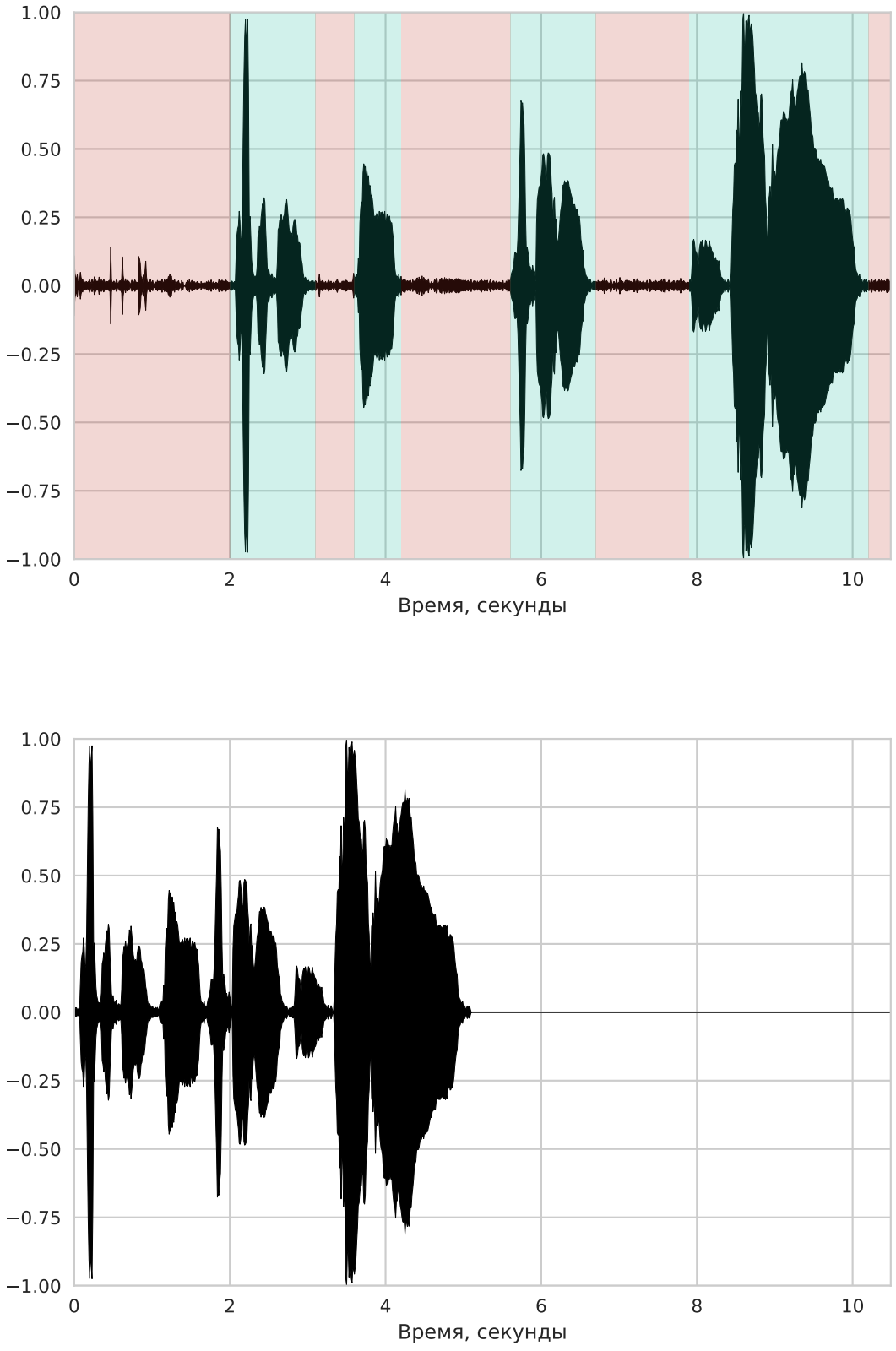

Рис. 2: Исходный сигнал (сверху) с выделенными речевыми (зелёным) и не речевыми (красным) сегментами и результат применения VAD (снизу)

конкатенированные с логарифмом энергии сигнала:

$$S(1 \dots 19, 1 \dots T) = \mathrm{MFCC}_{19}(s),$$
$$S(20, 1 \dots T) = \log \sum_{n=1}^{N} |s(n)|^2,$$

где N — длина дискретного сигнала, T — размер полученных признаков MFCC по временной оси.

Построенные таким образом признаки размера $20 \times T_1$ затем снова разбивались на окна по 11 фреймов и к ним применялось дискретное косинусное преобразование (discrete cosine transform, DCT) вдоль временной оси, после чего первые 5 коэффициентов каждой строки конкатенировались в вектор размерности 100. Выбор подобных признаков обусловлен их успешным применением в задаче автоматического распознавания речи [25,34,35]. Полученные признаки подавались на вход VAD для определения, принадлежат данные 11 фреймов MFCC (110 миллисекуд) к речевому сегменту или нет. В качестве самого VAD использовалась скрытая марковская модель (hidden markov model, HMM) с двумя состояниями:

1. Окно, по которому построены текущие коэффициенты MFCC принадлежит речевому сегменту.

2. Окно, по которому построены текущие коэффициенты MFCC принадлежит не речевому сегменту.

Второе состояние само состоит из двух подсостояний:

2.1. Тишина.

2.2. Шум толпы.

Вероятнсоти перехода в каждое состояние равны между собой и равны 0.5. При этом на модель были наложены следующие ограничения:

- Длина речевого сегмента на может быть меньше 500 миллисекунд.

- Длина не речевого сегмента не может быть меньше 200 миллисекунд.

Для моделирования состояния 1 использовалось смесь из 1024 гауссов, которая была обучена на базе FISHER [5]. Состояние 2.1 описывается смесью

64 гауссов. Для его обучения использовались 30 файлов длительностью 15 секунд каждый, полученных суммированием некоторого подмножества записей из базы RSR2015 для моделирования шума толпы. Состоянию 2.2 соответствует смесь 8 гауссов, обученная на 12 фонограммах с записью фонового шума офиса и файле с нулевым значением сигнала на протяжении всей его длительности.

1.3 Извлечение признаков

Два типа признаков использовались в исследовании:

- Частотно-временная спектрограмма, полученная оконным преобразованием Фурье.
- Мел-частотные кепстральные коэффициенты (Mel-frequency cepstral coefficients, MFCC).

1.3.1 Спектрограмма

Обозначим исходный дискретный звуковой сигнал как $s(n), n \in \{1, \ldots, N\}$. Вычтем из сигнала среднее:

$$s'(n) = s(n) - \frac{1}{N}\sum_{k=1}^{N} s(k),\ n \in \{1, \ldots, N\}.$$

Зададимся шириной окна $L = 512$ и шагом $h = 128$ и разделим исходный сигнал на соответствующие окна [6]. Затем каждое окно $s_i(n)$, $i \in \{1, \ldots, \lfloor N/h \rfloor\}, n \in \{1, \ldots, L\}$ умножим поэлементно на оконную функцию Блэкмана

$$w(n) = 0.42 - 0.5\cos\frac{2\pi n}{L} + 0.08\cos\frac{4\pi n}{L},\ n \in \{1, \ldots, L\}.$$

Выбор оконной функции обусловлен её «хорошим» частотным откликом с боковыми лепестками начинающимися на уровне -58 ДБ и обеспечивающими тем самым более резкую и контрастную спектрограмму, что, судя по опыту решения задач распознавания образов, является важным свойством входных изображений при использовании свёрточных нейронных сетей. На рисунке 3 приведено сравнение частотных откликов и результиру-

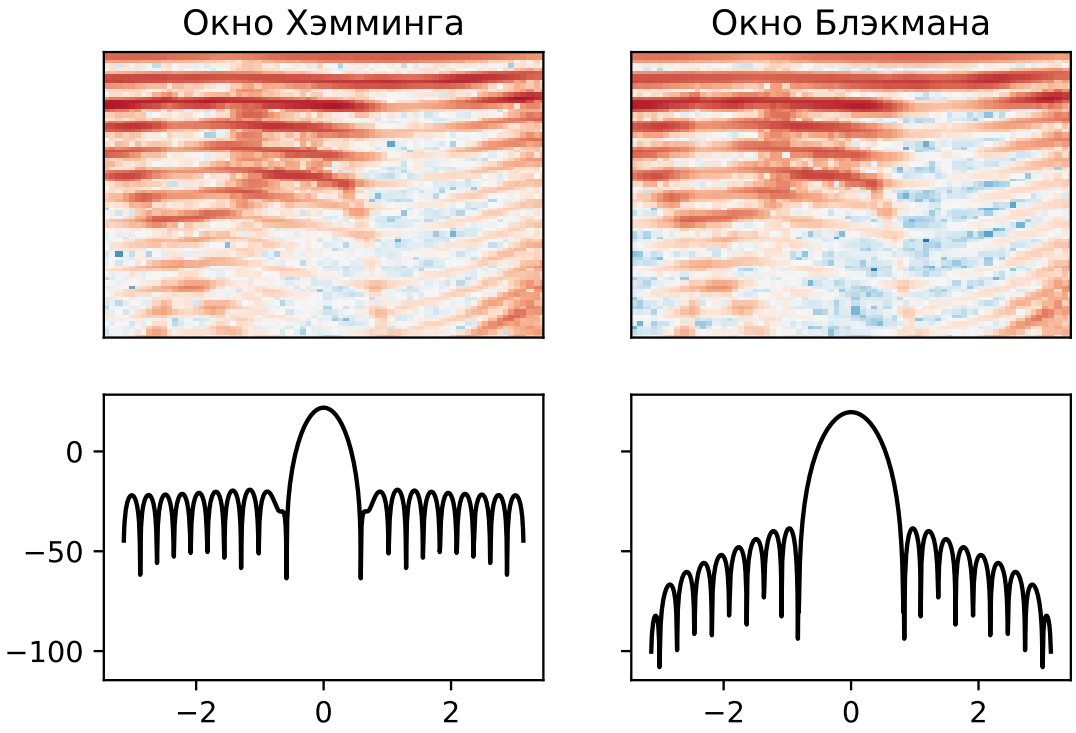

Рис. 3: Частотные отклики (снизу) различных оконных функций и фрагменты соответствующих им спектрограмм (сверху)

ющих спектрограмм для оконной функции Блэкмана и оконной функции Хэмминга, широко используемой в задачах анализа звука.

К сглаженному таким образом оконному сигналу s_i применим теперь дискретное преобразование Фурье

$$F(n+1) = \left| \sum_{k=0}^{L-1} s_i\,(k+1) \exp\left(-\frac{2\pi i}{L} nk\right) \right|,\ n \in \{0, \ldots, L-1\}$$

и получим 512 вещественных коэффициентов, из которых выберем первые 257.

Повторим описанную процедуру для каждого окна s_i и составим из полученных коэффициентов матрицу S размера $257 \times T$ — спектрограмму, где $T = \lfloor N/h \rfloor$. Останется лишь преобразовать абсолютные значения энергии сигнала в логарифмическую шкалу по формуле [7].

$$S'(i,j) = 20 \log_{10} \frac{S(i,j)}{10^{-6}}.$$

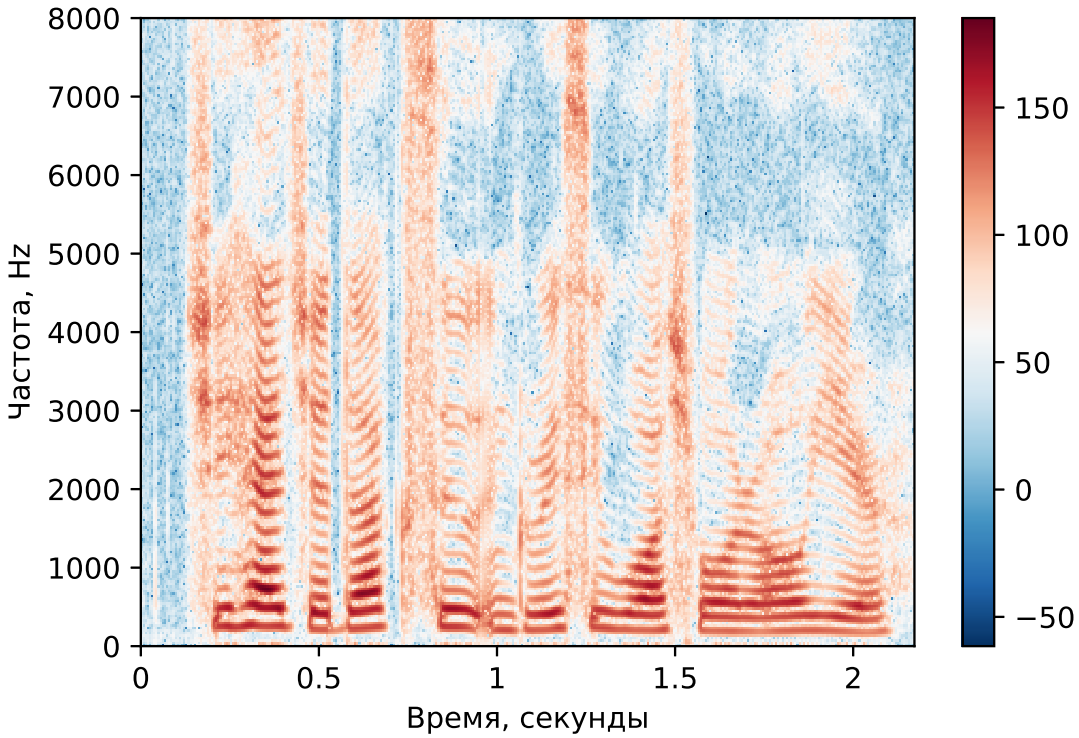

Рис. 4: Спектрограмма фразы «She had your dark suit in greasy wash water all year»

Пример спектрограммы полученной для фразы «She had your dark suit in greasy wash water all year» приведён на рисунке 4.

1.3.2 MFCC

Спектрограмма полностью описывает сигнал в частотно-временной области с некоторым фиксированным по обеим осям резрешением. Однако, существует две проблемы, не позволяющие использовать спектрограммы в качестве входных признаков для большинства классических алгоритмов машинного обучения:

1. Размерность спектрограммы слишком велика.
2. Такое представление сигнала никак не учитывает особенности человеческого восприятия, которые могли бы быть дополнительной априорной информацией, вносимой в систему.

Обе вышеперечисленные особенности учитываются в Мел-частотных кепстральных коэффициентах (MFCC). Для их получения зададимся не логарифмированной спектрограммой S сигнала s и вычислим спектр её мощ-

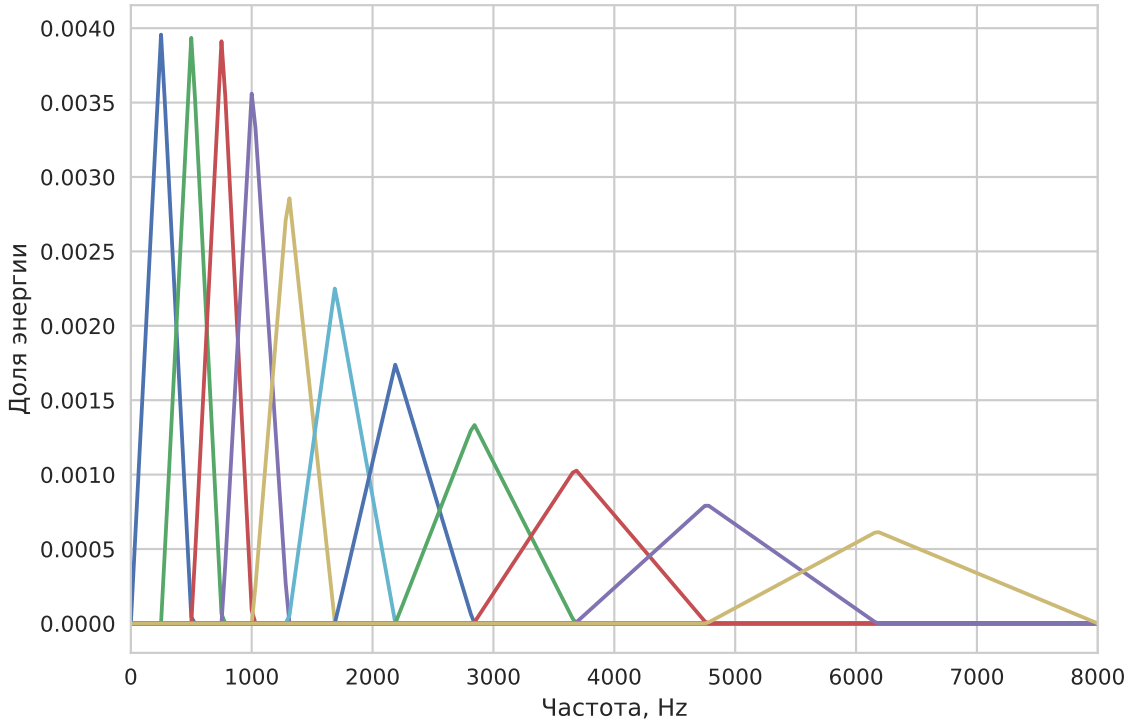

Рис. 5: 11 равномерных треугольных Мел-шкалированных фильтров в стандартной шкале

ности

$$P(i,j) = \frac{1}{L} S(i,j),\ i \in \{1,\dots,L\},\ j \in \{1,\dots,T\}.$$

Следующим шагом необходимо вычислить банк из $M = 26$ треугольных фильтров, преобразованных из Мел шкалы. Мел — единица измерения высоты звука, основанная на психофизическом восприятии звука человеком и построенная на основе анализа большого числа статистических данных. Равным дистанциям на мел-шкале соответствуют одинаковые разницы высоты звука, оцениваемые слушателями. Расположим равномерно линейно M фильтров на мел-шкале от 0 до F_h — максимальной частоты в нашем сигнале. Затем отобразим их в стандартную шкалу. Пример фильтров в стандартной шкале для $M = 11$ изображён на рисунке 5. Важно заметить, что каждый фильтр определён на всём возможном диапазоне частот $[0, F_h]$, а не только на тех учатках, где он не обращается в ноль.

Далее для каждого столбца $P_j(k) = P(k,j)$, $j \in \{1,\dots,T\}$ полученного ранее спектра:

1. Свернём $P_j(k)$ с каждым из мел-шкалированных фильтров, чтобы по-

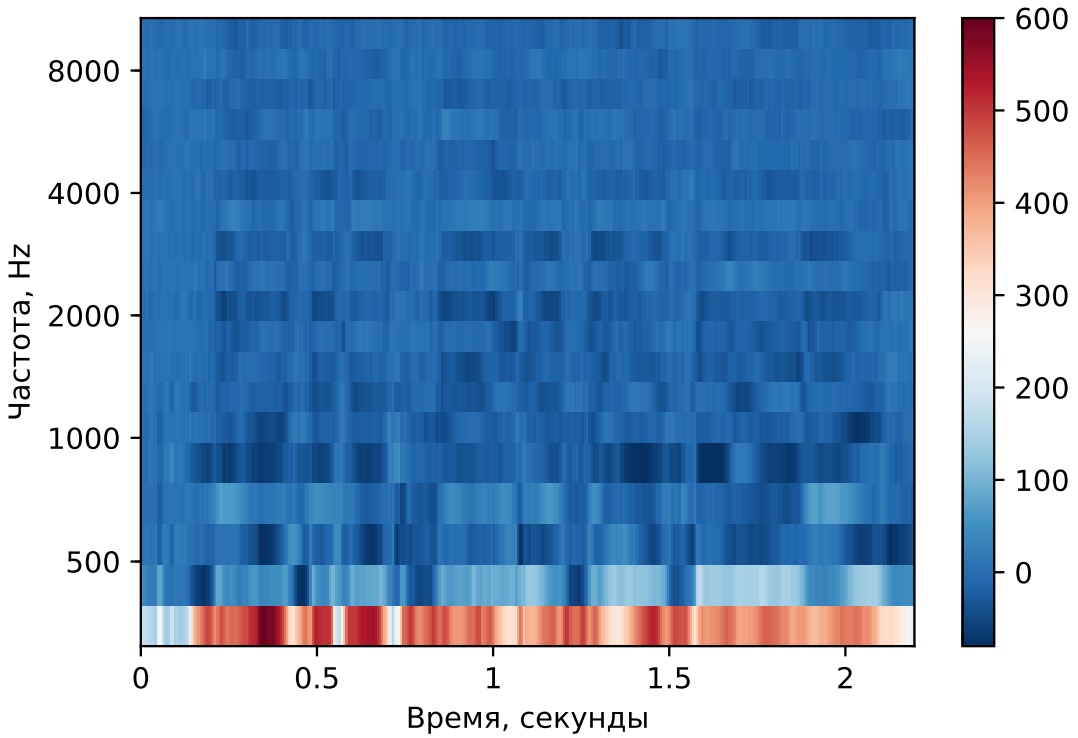

Рис. 6: 19 MFCC для фразы «She had your dark suit in greasy wash water all year»

лучить M вещественных коэффициентов.

2. Применим к ним фукнцию логарифма.

3. Трактуя полученные M коэффициентов как сигнал, применим к нему DCT и оставим лишь первые $D = 19$ коэффициентов.

Объединив полученные наборы признаков по столбцам в матрицу размера $D \times T$, получим MFCC. Пример для фразы «She had your dark suit in greasy wash water all year» приведён на рисунке 6.

1.3.3 Извлечение признаков

Для извлечения признаков в рамках данной работы был реализован программный модуль на языке Python 3. Загрузка wav-файлов осуществляется с помощью функции scipy.io.wavfile.read пакета scipy [11], а для построения спектрограмм и MFCC используются функции модуля scipy.signal.

Для всех имеющихся в базах произнесений были извлечены спектрограммы и MFCC по описанным выше алгоритмам.

1.4 Предварительная обработка признаков

1.4.1 Спектрограммы

Так как спектрограммы используются в качестве входных признаков для свёрточной нейронной сети, все они должны иметь одинаковый размер. Размер спектрограмм по частотной оси фиксирован и равен 257, однако размер по временной оси не фиксирован и меняется в зависимости от длины произнесения.

Нельзя допускать изменения пропорций, масштаба или угла поворота спектрограмм, так как это исказит значимые признаки. Единственной доступной операцией является обрезка спектрограммы и её дополнение. Исходя из этого, для приведения спктрограмм к единому размеру предложен следующий алгоритм.

1. Выбрать $\hat{T}$.
2. Для каждой спектрограммы S:
   (a) Если длина текущей спектрограммы по временной оси T равна $\hat{T}$, то оставить спектрограмму в исходном виде:

$$\hat{S}(i,j) = S(i,j),\ i \in \overline{1,L},\ j \in \overline{1,T}.$$

   (b) Если $T > \hat{T}$, то обрезать спектрограмму по временной оси до $\hat{T}$:

$$\hat{S}(i,j) = S(i,j),\ i \in \overline{1,L},\ j \in \overline{1,\hat{T}}.$$

   (c) Если $T < \hat{T}$, то дополнить спектрограмму справа собственными копиями до требуемого размера:

$$\hat{S}(i,j) = S(i, j \bmod T),\ i \in \overline{1,L},\ j \in \overline{1,\hat{T}}.$$

Отдельно стоит обсудить вопрос выбора $\hat{T}$. Очевидно, что $\hat{T}$ должно быть меньше, чем максимальная длина спектрограммы по временной оси в обучающей выборке. Как будет обсуждено позднее, преимуществом свёрточных сетей является их инвариантность к топологии входных признаков: благодаря архитектуре таких сетей, расположение и повторяемость

отдельных паттернов не имеют большого влияния на результат работы сети. Таким образом, повторённая несколько раз подряд спектрограмма не является проблемой для свёрточных нейронных сетей. С другой стороны, обрезка слишком длинных спектрограмм может привести к потере важной дикторской информации. Исходя из этих рассуждений стоило бы сделать $\hat{T}$ максимально возможным, однако в выборе его значения стоит учитывать ещё один фактор — размерность признаков. С увеличением размера спектрограмм увеличивается и время обучения сети и необходимое для этого количество памяти.

В данной работе $\hat{T} = 800$ для базы RSR2015 и $\hat{T} = 622$ для базы NIST. Все спектрограммы были обработаны вышеизложенным алгоритмом и приведены тем самым к единому размеру.

В ходе экспериментов были рассмотрены различные способы нормализации спектрограмм, однако лучшие результаты были достигнуты при нормализации на среднее и стандартное отклонение вдоль временной оси:

$$S(i,j) = \frac{S(i,j) - m(i)}{\sigma(i)},\ i \in \{1,\dots,L\},\ j \in \left\{1,\dots,\hat{T}\right\},$$

$$m(i) = \frac{1}{\hat{T}} \sum_{j=1}^{\hat{T}} S(i,j),$$

$$\sigma(i) = \left[\frac{1}{\hat{T}} \sum_{j=1}^{\hat{T}} \left(S(i,j) - m(i)\right)^2\right]^{\frac{1}{2}}.$$

Последний этап в предварительной обработке спектрограмм — подсчёт среднего изображения по обучающей выборке и вычитание его из каждого примера.

1.4.2 MFCC

В качестве предварительной обработки MFCC использовалась их нормализация на среднее и стандартное отклонение вдоль временной оси полностью аналогично случаю спектрограмм и последующая аугментация.

Аугментация признаков производилась путём конкатенации к ним конечных разностей первого и второго порядков, подсчитанных вдоль временной оси. Пусть S — MFCC размерности $D \times T$. Тогда конечные разности

первого и второго порядка определены для неё соответственно как

$$\Delta_S(i,j) = \frac{1}{10}\Big[2S(i,j+2) - 2S(i,j-2) + S(i,j+1) - S(i,j-1)\Big],$$
$$i \in \{1,\ldots,D\},\ j \in \{3,\ldots,T-2\},$$
$$\Delta\Delta_S(i,j) = \frac{1}{10}\Big[2\Delta_S(i,j+2) - 2\Delta_S(i,j-2) + \Delta_S(i,j+1) - \Delta_S(i,j-1)\Big],$$
$$i \in \{1,\ldots,D\},\ j \in \{5,\ldots,T-4\}.$$

Конкатенация производится вдоль частотной оси. Так как получившиеся матрицы не совпадают в размере по временной оси, конечные разности дополняются слева и справа копиями соответствующих крайних векторов до нужного размера. Таким образом, результирующие признаки имеют размер $3D \times T$.

Глава 2. Классическая базовая система

2.1 Универсальная фоновая модель

Универсальная фоновая модель (universal background model, UBM) в задаче распознавания диктора представляет собой некоторую модель, обученную на конечном множестве произнесений определённых дикторов и содержащую в себе апостерирные знания об устройстве человеческого голоса в целом. Как правило, в качестве UBM применяется смесь гауссовых распределений (gaussian mixture model, GMM) [1]. UBM обучается методом Expectation-maximization на обучающем множестве. Вектор средних, извлечённый из модели после обучения, называется супервектором средних. Когда необходимо получить признаки для вновь пришедшего произнесения, параметры UBM подстраиваются методом оценки апостериорного максимума (maximum a posteriori probability estimate) и полученный вектор средних модели уже называется дикторским супервектором средних. Принцип обучения и работы UBM изображён на рисунке 7.

2.2 Извлечение i-векторов

Система на основе i-векторов моделирует речевое произнесение как вектор, содержащий дикторскую и канальную (устройство записи и канал передачи) ифнормацию с помощью модели toal variability:

$$s = \mu + Tw,$$

где s — дикторский супервектор средних, μ — супервектор средних, извлечённый из UBM, T — матрица низкого ранга, а w — искомый i-вектор, получаемый методом факторного анализа [20].

2.3 Сравнение i-векторов

Извлечённые i-вектора сравниваются между собой подсчётом косинусной дистанции между ними по формуле

$$h\left(x, y\right) = \frac{xy}{\|x\|\|y\|}$$

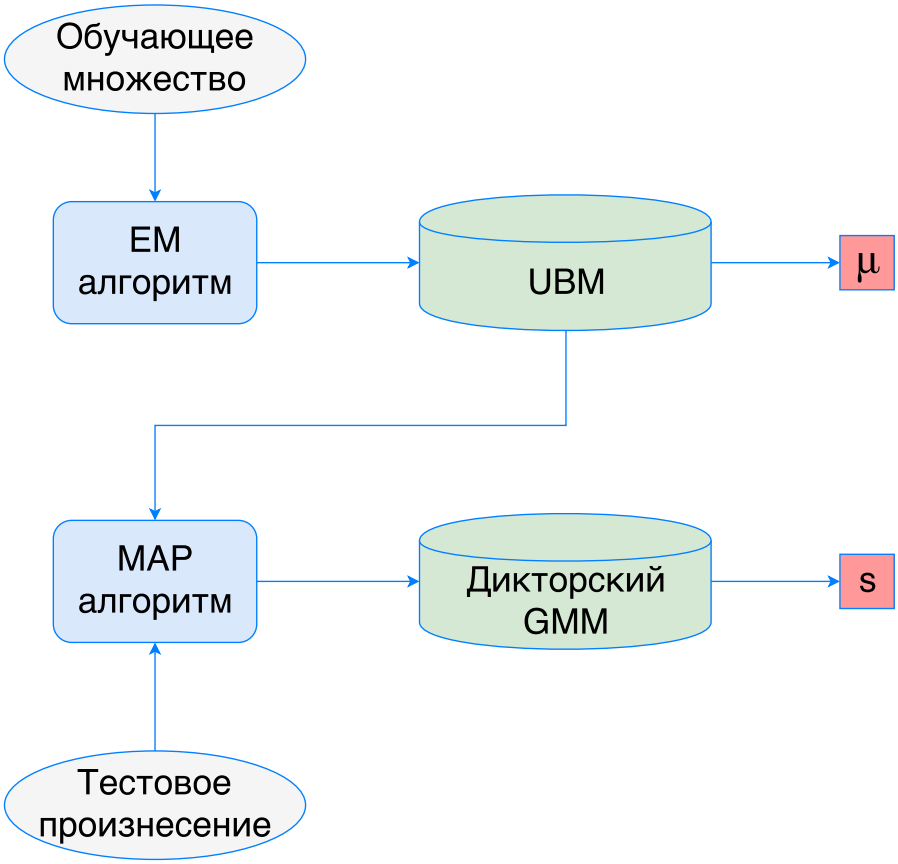

Рис. 7: Схема обучения и функционирования UBM-GMM

с последующей s-нормализацией [29].

Если $h(x, y) = 1$, то вектора x и y идентичны и, исходя из однозначности построенного идентификатора F, получены из одной и той же фонограммы. В любом другом случае нельзя сказать с определённостью об идентичности дикторов, чьим произнесениям соответствуют данные вектора.

При практическом применении системы выбирается некоторое пороговое значение $d \in (-1, 1)$. Вектора x и y, удовлетворяющие

$$h(x, y) > d$$

признаются принадлежащими одному диктору, не удовлетворяющие — разным. Выбор d зависит от конкретной решаемой практической задачи и предъявляемых к системе требований.

После извлечения все i-вектора нормируются по длине и к ним применяется регуляризация методом внутриклассовой ковариационной нормали-

зации (Within-class Covariance Normalization, WCCN) [19].

Глава 3. Система на основе глубоких нейронных сетей

3.1 Свёрточные нейронные сети

Свёрточная нейронная сеть — одна из архитектур искусственных нейронных сетей, которая основывается на особенностях зрительной коры головного мозга и выгодно отличается от полносвязных искусственных нейронных сетей в определённом ряде задач.

Слой полносвязной нейронной сети определяется как

$$y = Wx + b,$$

где x — входной вектор размерности n_i, y — выходной вектор размерности n_o, W — матрица весов размерности $n_o \times n_i$, b — вектор смещения размерности n_o. Попытка применения полносвязной нейронной сети к двумерным структурам, таким как спектрограммы, приводит к двум серьёзным проблемам. Во-первых, спектрограммы имеют большой размер и полносвязная нейронная сеть будет иметь слишком большое количество параметров и связей , что приведёт к переобучению и невозможности использовать такую модель на практике. К примеру, спектрограмма размера 257×800, будучи представленной как вектор, будет содержать 205600 элементов, и, при количестве нейронов скрытого слоя равном 512, полносвязная однослойная нейронная сеть будет иметь 106 миллионов параметров. Во-вторых, спектрограммы содержат большое число локальных паттернов, примеры которых можно увидеть на рисунке 8. Таким образом, кроме численных значений признаков так же важна и их тополгия. Для полносвязной нейронной сети возможность обнаруживать такие повторения означает наличие одинаковых или крайне похожих наборов параметров в различных строках матрицы W.

3.1.1 Свёрточный слой

Свёрточные нейронные сети, впервые предложенные Яном Лекуном [8], решают описанные проблемы, применяя локальные ядра свёртки ко всевозможным маленьким участкам изображения. Веса у всех таких ядер общие,

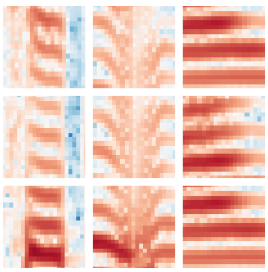

Рис. 8: Примеры паттернов на изображении спектрограммы (по столбцам)

что позволяет искать паттерны, не зависящие от топологии спектрограммы, используя при этом относительно малое количество параметров. Зависимость входа от выхода в свёрточном слое с ядром свёртки размера $k_v \times k_h$ описывается формулой

$$
\begin{aligned}
y_{ijk} &= \sum_{u=0}^{k_v} \sum_{v=0}^{k_h} \sum_{n=0}^{n(x)} W_{uvk} \cdot x_{(i+u)(j+v)n} + b_k, \\
i &\in \{as_v + 1 \mid a \in \{0, \ldots, h(x) \bmod s_v\}\}, \\
j &\in \{as_h + 1 \mid a \in \{0, \ldots, w(x) \bmod s_h\}\}, \\
k &\in \{1, \ldots, n(y)\},
\end{aligned}
$$

где W — трёхмерный тензор весов, b — вектор смещений, x и y — входной и выходной трёхмерные тензоры, $h(x)$ и $w(x)$ обозначают высоту и ширину входного изображения, s_v и s_h — шаг по вертикали и по горизонтали, а $n(x)$ и $n(y)$ — количество каналов входного и выходного изображений соответственно.

Вход и выход свёрточного слоя в общем случае являются трёхмерными тензорами. Первые две размерности интуитивно понятно соотносятся с высотой и шириной изображения (спектрограммы), а третья размерность отвечает количеству каналов (карт признаков). Множество таких карт позволяет использовать множество ядер свёртки для поиска различных паттернов. Количество каналов в свёрточном слое аналогично количеству нейронов в скрытом слое классической полносвязной сети. Если входом свёрточной сети служит цветное изображение, то оно уже является трёхмерным тензором, где третьему измерению отвечают красный, зелёный и синий цветовые каналы. В случае же спектрограмм, на вход сети подаётся тензор с одним каналом, что аналогично случаю серого изображения.

3.1.2 Слой линейной ректификации

Слой активации в свёрточных нейронных сетях полностью аналогичен таковому в полносвязных и состоит в поэлементном применении функции активации σ к входному тензору:

$$\begin{aligned} &y_{ijk} = \sigma\left(x_{ijk}\right), \\ &i \in \{1, \ldots, h(x)\}, \\ &j \in \{1, \ldots, w(x)\}, \\ &k \in \{1, \ldots, n(x)\}. \end{aligned}$$

В качестве функции активации после свёрточных слоёв в подавляющем большинстве случаев используется блок линейной ректификации, исследованный во многих работах [12–15] и показавший свою эффективность во всех задачах, где применяются свёрточные нейронные сети. Функция линейной ректификации выглядит следующим образом:

$$\sigma(x) = \max(0, x) = \begin{cases} x, & x \geqslant 0 \\ 0, & x < 0 \end{cases}.$$

График функции изображён на рисунке 9.

После применения такой функции активации сигналы с отрицательным знаком отсеиваются и не распространяются далее по сети, а положительные сигналы остаются неизменными. Производная функции линейной ректификации

$$\frac{d\sigma}{dx} = \begin{cases} 1, & x \geqslant 0 \\ 0, & x < 0 \end{cases}$$

имеет следующий физический смысл: ошибка распространяется неизменной по тем путям, где сигнал был положителен и не распространяется вовсе по тем путям, где сигнал был отрицателен.

3.1.3 Слой субдискретизации

Слоями субдискретизации называются нелинейные преобразования, действующие не на отдельные компоненты тензора, а на их группы. Нами будут использованы два типа таких слоёв: максимум и среднее по каналам.

Рис. 9: График функции линейной ректификации

Слой субдискретизации функцией максимума (maximum pooling) с окном $m \times m$ и величиной шага s_v и s_h по вертикали и горизонтали соответственно определяется как

$$
\begin{aligned}
&y_{ijk} = \max\left\{x_{(i+u)(j+v)k} \;\middle|\; u, v \in \{0, \ldots, m\}\right\}, \\
&i \in \{as_v + 1 \mid a \in \{0, \ldots, h(x) \bmod s_v\}\}, \\
&j \in \{as_h + 1 \mid a \in \{0, \ldots, w(x) \bmod s_h\}\}, \\
&k \in \{1, \ldots, n(x)\},
\end{aligned}
$$

где x — входной тензор размера $h(x) \times w(x) \times n(x)$, y — результирующий тензор. Подобное отображение оставляет лишь сигнал с максимальным значением среди всех квадратов размера $m \times m$ взятых с определённым шагом. Тем самым, подобный слой субдискретизации уменьшает размерность признаков, оставляя среди них лишь самые «важные» в смысле максимума.

Слой субдискретизации функцией среднего по каналам (average pooling) можно выразить формулой

$$
y_k = \frac{1}{h(x)w(x)} \sum_{u,v=1}^{h(x),w(x)} x_{uvk}, \; k \in \{1, \ldots, n(x)\},
$$

где все обозначения эквивалентны ранее введённым. Результат работы такого слоя — одномерный вектор, зависящий лишь от количества карт признаков исходного тензора, но не от его высоты и ширины. Таким образом,

данная операция может быть применена для получения признаков высокого уровня некоторой фиксированной размерности из изображения (спектрограммы) произвольного размера.

3.1.4 Слой нормализации

Обучение глубоких нейронных сетей сопряжено с серьёзной проблемой: распределение входных признаков каждого слоя меняется во время обучения, так как меняются параметры предыдущего слоя. Из-за этого обучение замедляется и требует более точной настройки параметров оптимизирующего алгоритма и более низких значений скорости обучения. Авторы [18] для решения данной проблемы предлагают сделать нормализацию признаков частью глубокой архитектуры и вводят понятие слоя нормализации, который связывает входную и выходную группы тензоров $\{x\}_{i=1}^{L}$ и $\{y\}_{i=1}^{L}$ соотношением

$$
\begin{aligned}
y_{ijk}^{l} &= \gamma_k \frac{x_{ijk}^{l} - \mu_{ijk}}{\sqrt{\sigma_{ijk}^{2} + \varepsilon}} + \beta_k, \\
i &\in \{1, \ldots, h(x)\}, \\
j &\in \{1, \ldots, w(x)\}, \\
k &\in \{1, \ldots, n(x)\}, \\
l &\in \{1, \ldots, L\},
\end{aligned}
$$

где верхний индекс l обозначает l-ый пример из текущей группы, μ и σ — трёхмерные тензоры выборочного среднего и стандартного отклонения, посчитанные по всей текущей группе $\{x\}_{i=1}^{L}$, γ и β — обучаемые вектора параметров, которые используются для приведения признаков к требуемому распределению, а ε — некоторая константа, предназначенная для предотвращения деления на ноль.

3.2 Residual отображения

Главное преимущество и сила свёрточных нейронных сетей заключается в их многослойности [8, 9, 16, 17]. Относительно небольшое количсетво параметров на каждом слое позволяет строить глубокие архитектуры, где на каждом последующем слое извлекаются всё более высокоуровневые при-

знаки.

Однако, эксперименты [9] показывают, что невозможно увеличивать глубину нейронных сетей бесконечно. Например, хотя множество всевозможных значений параметров 34-слойной нейронной сети и содержит в себе множество всевозможных значений параметров 18-слойной сети той же архитектуры, на практике, обучаясь на одних и тех же данных, вторая сеть достигает меньших значений функции потерь и больших значений точности классификации.

Для решения подобных проблем авторы [9] предлагают вместо некоторого отображения $\mathcal{H}(x)$ внутри свёрточной нейронной сети обучать отображение заранее определённого вида

$$\mathcal{F}(x) = f(x) + \mathcal{H}(x).$$

Тогда исходное отображение запишется в виде

$$\mathcal{H}(x) = \mathcal{F}(x) - f(x).$$

Отображение такого вида называется residual отображением. Использующая этот подход архитектура достигла лучших результатов в соревновании по классификации изображений ImageNet [9] на момент публикации работы в декабре 2015 года.

Авторы [10] исследуют residual архитектуру более детально и уточняют, что лучшие результаты достигаются при выполнении следующих условий:

- $f(x) = x$ везде, где область определения и область значений $\mathcal{H}$ совпадают.
- Слои активации и нормализации всегда предшествуют свёрточным слоям в отличии от привычных архитектур, где они обычно применяются после.

На рисунке 10 схематично изображено отображение $\mathcal{F}$, являющееся основным строительным блоком residual сетей.

Успех подобной архитектуры можно объяснить двумя факторами, описанными ниже. Если предположить, что $f(x) \equiv x$ всюду, то уравнение для $\mathcal{F}$ запишется как

$$\mathcal{F}(x) = x + \mathcal{H}(x).$$

Тогда зависимость выхода L-го слоя сети относительно выхода l-го слоя может быть выражена следующим образом:

$$x_L = x_l + \sum_{i=l}^{L-1} \mathcal{F}_i(x_i)$$

Градиент функции ошибки ϵ в этом случае можно записать как

$$\frac{\partial \epsilon}{\partial x_l} = \frac{\partial \epsilon}{\partial x_L}\frac{\partial x_L}{\partial x_l} = \frac{\partial \epsilon}{\partial x_L}\left(1 + \frac{\partial}{\partial x_l}\sum_{i=l}^{L-1}\mathcal{F}_i(x_i)\right)$$

Такая форма градиента обеспечивает беспрепятственное распространение ошибки от функции потерь к входному слою благодаря аддитивному члену $\frac{\partial \epsilon}{\partial x_L}$ и избавляет от таких проблем глубоких сетей как угасание или разрастание градиента при его распространении.

Второе наблюдение заключается в том, что если выбранная глубина сети окажется избыточной, очевидно, что методу оптимизации гораздо легче прийти к константному отображению $\mathcal{H}(x) \equiv 0$, чем к отображению $\mathcal{F}(x) \equiv x$.

Конечно, условие $f(x) = x$ не может выполняться для всей сети для хоть сколько нибудь больших входных признаков (таких как спектрограммы), так как его использование исключит возможность редукции размерности данных и приведёт к внутреннему представлению слишком большого размера. Однако, если данное условие и связанные с ним свойства будут верны на достаточно больших локальных участках сети, то и для всей сети эти свойства будут выполняться в некоторой мере. Действительно, хоть ошибка, распространяющаяся по сети, и будет претерпевать нелинейные изменения там, где $f(x) \neq x$, таких участков будет гораздо меньше, чем слоёв во всей сети и количество таких участков будет сравнимо с глубиной стандартных (не residual) сетей, для которых экспериментально подтверждена сходимость.

Для тех $\mathcal{H}$, где размерность данных изменяется, в качестве f используется свёрточный слой с ядром свёртки размера 1×1 и одинаковым размером шага в обоих направлениях $s_h = s_w = 2$. Таким образом, возможно использование $\mathcal{H}$, изменяющей размер исходного тензора в два раза по пространственным измерениям и в произвольное число раз по последнему

измерению, отвечающему за число карт признаков.

В данной работе $\mathcal{H}$ представляет собой композицию шести слоёв в следующем порядке:

1. Слой нормализации 1.
2. Слой линейной ректификации 1.
3. Свёрточный слой 1.
4. Слой нормализации 2.
5. Слой линейной ректификации 2.
6. Свёрточный слой 2.

Исключение составляет самый первый residual блок в сети, в котором отсутствуют слои 1 и 2. Высота k_v и ширина k_h ядер свёртки обоих свёрточных слоёв равна 3. Если размер входного и выходного тензора residual блока совпадают, то для обоих свёрточных слоёв используется шаг $s_h = s_w = 1$, иначе для первой свёртки используется шаг $s_h = s_w = 2$, что уменьшает размер признаков вдвое.

3.3 Глубокая архитектура

В качестве архитектуры глубокой свёрточной сети для проведения экспериментов была выбрана 18-слойная сеть, исследованная в [9]. Причины выбора именно этой архитектуры следующие:

- Сеть данной архитектуры достигла высоких результатов в задаче распознавания изображений конкурса ImageNet.
- Сеть имеет небольшое число параметров по сравнению с другими архитектурами схожей глубины.
- Сеть является самой простой из всех исследованных ранее residual архитектур, что делает её отличным кандидатом для проведения первых экспериментов по применению таких архитектур к исследуемой задаче.

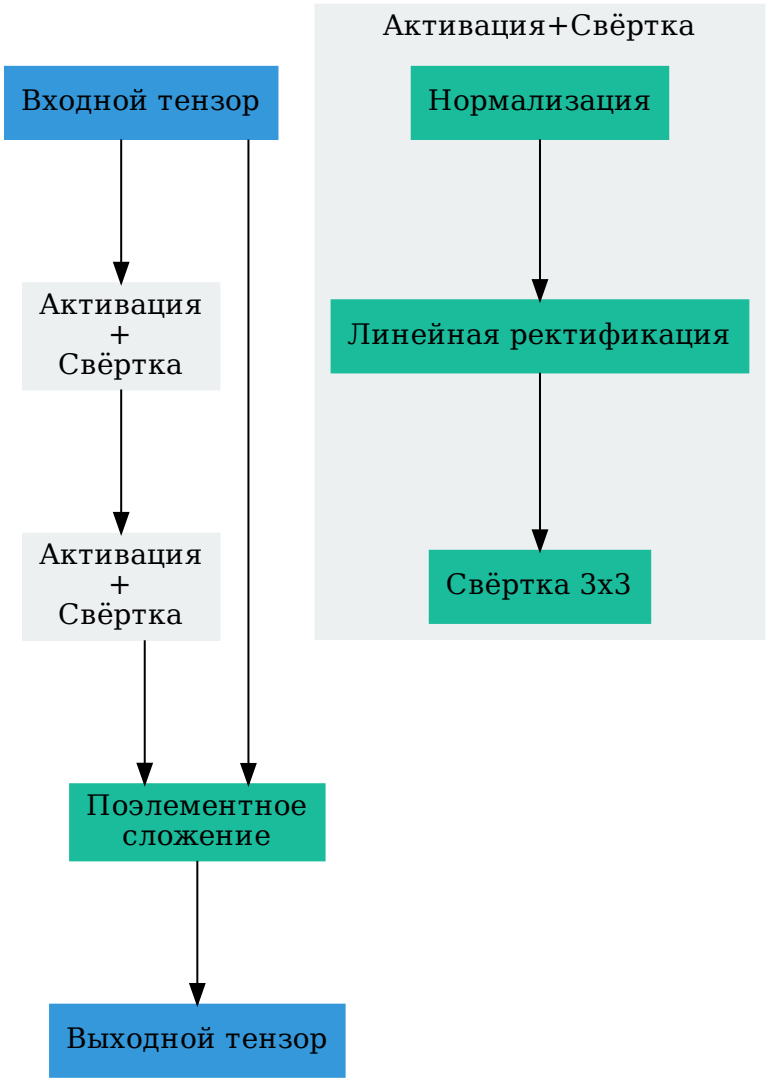

Рис. 10: Схематичное изображение residual блока с улучшениями из [10] и при условии $f(x) = x$

- Размер предпоследнего слоя зафиксирована и не зависит от размера входных признаков, что позволяет применять сеть к большим изображениям и использовать одинаковую архитектуру для разных признаков.
- Размер предпоследнего слоя, равный 512, близок к типичному размеру i-вектора, что позволяет использовать его выход как высокоуровневый признак для исходного произнесения.

Архитектура сети для спектрограмм размера 257×800 представлена в таблице 1. Введены следующие обозначения:

- **Conv** — свёрточный слой.
- **BN** — слой нормализации.
- **ReLU** — слой линейной ректификации.
- **Residual** — residual блок.
- **MaxPooling**, **AvgPooling** — слои субдискретизации: максимум и среднее по каналам соответственно.
- **Linear** — полносвязный слой.
- **SoftMax** — функция активации SoftMax.

Количество классов, равное количеству нейронов на последнем слое, обозначено как N_c. Общее количество параметров посчитано без учёта последнего слоя.

3.4 Извлечение высокоуровневых признаков

На этапе обучения сеть представленной архитектуры обучается решать закрытую задачу классификации на обучающем множестве с N_c классами. В качестве функции потерь используется категориальная кроссэнтропия:

$$L(t, p) = -\frac{1}{N} \sum_{i=1}^{N} t_i \log(p_i),$$

Таблица 1: Архитектура используемой глубокой нейронной сети

Тип слоя	Размер ядра/шага	Размер выхода	Параметры
Вход	–	$257 \times 800 \times 1$	0
Conv	$7 \times 7/2 \times 2$	$129 \times 400 \times 64$	3.2K
BN	–	$129 \times 400 \times 64$	256
ReLU	–	$129 \times 400 \times 64$	0
MaxPool	$3 \times 3/2 \times 2$	$65 \times 200 \times 64$	0
Residual	$3 \times 3/1 \times 1$	$65 \times 200 \times 64$	74.1K
	$3 \times 3/1 \times 1$		
Residual	$3 \times 3/1 \times 1$	$65 \times 200 \times 64$	74.1K
	$3 \times 3/1 \times 1$		
Residual	$3 \times 3/2 \times 2$	$33 \times 100 \times 128$	230.1K
	$3 \times 3/1 \times 1$		
Residual	$3 \times 3/1 \times 1$	$33 \times 100 \times 128$	296.2K
	$3 \times 3/1 \times 1$		
Residual	$3 \times 3/2 \times 2$	$17 \times 50 \times 256$	919.8K
	$3 \times 3/1 \times 1$		
Residual	$3 \times 3/1 \times 1$	$17 \times 50 \times 256$	1 182.2K
	$3 \times 3/1 \times 1$		
Residual	$3 \times 3/2 \times 2$	$9 \times 25 \times 512$	3 674.7K
	$3 \times 3/1 \times 1$		
Residual	$3 \times 3/1 \times 1$	$9 \times 25 \times 512$	4 723.7K
	$3 \times 3/1 \times 1$		
AvgPool	–	512	0
Linear	N_c	N_c	$N_c \times 512$
SoftMax	–	N_c	0
Всего			11 178.4K

где p — вектор предсказаний сети (выход softmax слоя), а t — вектор правильных ответов. Обучение останавливается по достижении приемлемой точности классификации на валидационном множестве.

На этапе использования модели, выход её последнего softmax слоя игнорируется, а в качестве выхода сети используется 512-мерный вектор, сформировавшийся на предпоследнем слое сети. Предположение, лежащее в основе такого метода обучения и использования нейронной сети состоит в том, что предпоследний слой обученной сети содержит некоторое репрезентативное высокоуровневое представление исходных признаков в пространстве относительно низкой размерности. Действительно, если зафиксировать всю сеть кроме последнего слоя, получившаяся модель будет линейной с малым относительно общего числа количеством параметров (0.45% для $N_c = 100$, 4.6% для $N_c = 1000$, 9.1% для $N_c = 2000$), а значит, для успешного решения задачи классификации, данные, служащие входом для данной модели, должны быть хорошо линейно разделимы.

3.5 Сравнение высокоуровневых признаков

Извлечённые высокоуровневые 512-мерные признаки сравниваются так же, как и в случае базовой системы: подсчётом косинусной дистанции между ними.

Глава 4. Эксперименты и результаты

4.1 Проведение экспериментов

4.1.1 Программная реализация

Для проведения экспериментов с базовой системой была использована её закрытая реализация, предоставленная ООО «ЦРТ».

Программная реализация глубокой свёрточной нейронной сети была реализована в рамках данной работы на языке Python 3 с использованием библиотек TensorFlow [36] и Keras [37]. Python 3, будучи высокоуровневым интерпретируемым языком программирования, позволяет быстро проводить эксперименты, используя при этом малое количество кода. Реализация глубоких нейронных сетей требует вычисления огромного числа матричных операций. Библиотека TensorFlow позволяет перенести все такие операции на графический процессор (Graphics Processing Unit, GPU), который, благодаря встроенному параллелизму и большому числу вычислительных ядер, позволяет ускорить расчёты и сократить тем самым время обучения глубоких моделей во много раз. Библиотека Keras является высокоуровневой абстракцией над TensorFlow и оперирует уже такими понятиями, описанными в третьей главе, как свёрточный слой, слой линейной ректификации, слои субдискретизации и т. д. Нейронная сеть, являющаяся чистой функцией, представляется в TensorFlow в виде вычислительного графа, где узлы обозначают переменные и операции над ними, а рёбра — потоки данных. Методом обратного распространения ошибки библиотека способна в полностью автоматическом режиме вычислить градиент описанной функции по всем необходимым параметрам. Используя градиент легко затем минимизировать данную функции любым методом многомерной оптимизации первого порядка.

Обучение сети производилось с помощью модифицированного метода стохастического градиентного спуска ADAM [38]. Нейронная сеть обучалась на протяжении 100 эпох. Затем набор параметров, показавший наилучший результат на валидационном множестве во время обучения, использовался для проверки модели на тестовом множестве. Скорость обучения равнялась 10^{-4} в начале обучения и уменьшалась в 10 раз каждые 30 эпох.

Для проведения экспериментов использовалась графическая карта Nvidia GeForce GTX 980 Ti, содержащая 2816 вычислительных ядер и 6 гигабайт оперативной памяти. Время обучения глубокой нейронной сети при этом составило 26 часов на базе RSR2015 и 14 часов на базе NIST.

4.1.2 Методика тестирования

Протоколы тестирования обоих баз состоят из двух списков:

1. **Target-сравнения**. Моделируют попытку доступа оригинального пользователя к системе и должны завершиться удачно.
2. **Imposter-сравнения**. Моделируют попытку авторизации пользователя, не имеющего доступа к системе и должны завершиться неудачей.

Сравнение представляет собой либо пару фонограмм (x_1, x_2), либо четвёрку (e_1, e_2, e_3, x). Второй случай моделирует сравнение произнесения x с эталонной моделью голоса диктора, выраженной тремя регистрационными записями e_1, e_2 и e_3. В первом случае для получения вероятности принадлежности произнесений одному и тому же диктору необходимо извлечь с помощью обученной модели F два высокоуровневых представления фонограмм и сравнить их зафиксированной функцией h:

$$s = h(F(x_1), F(x_2)).$$

Второй случай аналогичен первому с тем лишь отличием, что в качестве вектора для сравнения берётся средний эталонный вектор:

$$s = h\left(\frac{1}{3}\left[F\left(e_1\right) + F\left(e_2\right) + F\left(e_3\right)\right], F\left(x_2\right)\right).$$

Полученные для каждого сравнения результаты образуют два множества: множество результатов target-сравнений M_T и множество результатов imposter-сравнений M_I. Величина

$$\mathrm{FRR}(d) = \frac{|\{x \in M_T \mid x < d\}|}{|M_T|}$$

носит название False Rejection Rate (FRR) и равна доле target-сравнений,

которые будут неверно отвергнуты при заданном пороге d. Аналогично величина False Acceptance Rate (FAR) равна доле неправильно принятых imposter-сравнений:

$$\mathrm{FAR}(d) = \frac{|\{x \in M_I \mid x > d\}|}{|M_I|}.$$

График взаимной зависимости этих величин является важной и репрезентативной характеристикой системы и носит название **DET-кривой**. Метрика **Equal Error Rate** (EER) характеризует точку, в которой доли ошибок обоих типов равны:

$$\mathrm{EER} = \mathrm{FRR}\left(\arg\min_{d\in(-1,1)}\left|\mathrm{FRR}(d) - \mathrm{FAR}(d)\right|\right).$$

4.1.3 Композиция систем

Наравне с отдельными сравниваемыми системами иногда бывает полезно рассмотреть и их композицию: объединение двух независимых подходов может продемонстрировать лучший результат, чем каждый из них по отдельности. В данной работе рассматривается композиция базовой и исследуемой систем на уровне результатов сравнений. Пусть s_1 и s_2 — результаты некоторого сравнения, полученные для двух систем. Тогда соответствующий результат композиции этих систем вычисляется по формуле

$$s_c = \omega_1 s_1 + \omega_2 s_2.$$

Коэффициенты ω_1 и ω_2 могут быть либо установлены вручную эмпирическим путём, либо найдены с помощью логистической регрессии на некотором настроечном подмножестве обучающих данных. В данной работе используется второй подход.

4.2 Результаты

Результаты тестирования базовой и исследуемой систем на базе RSR2015 приведены в таблице 2. На рисунке 11 изображена соответствующая DET-кривая. В целях изучения зависимости результата обучения глубокой нейронной сети от количества входных данных, дополнительный эксперимент

Таблица 2: Результаты на базе RSR2015

Система	EER (%)
Базовая система	0.79
Исследуемая система	6.02
Исследуемая система (расширенная)	5.23
Композиция	**0.64**

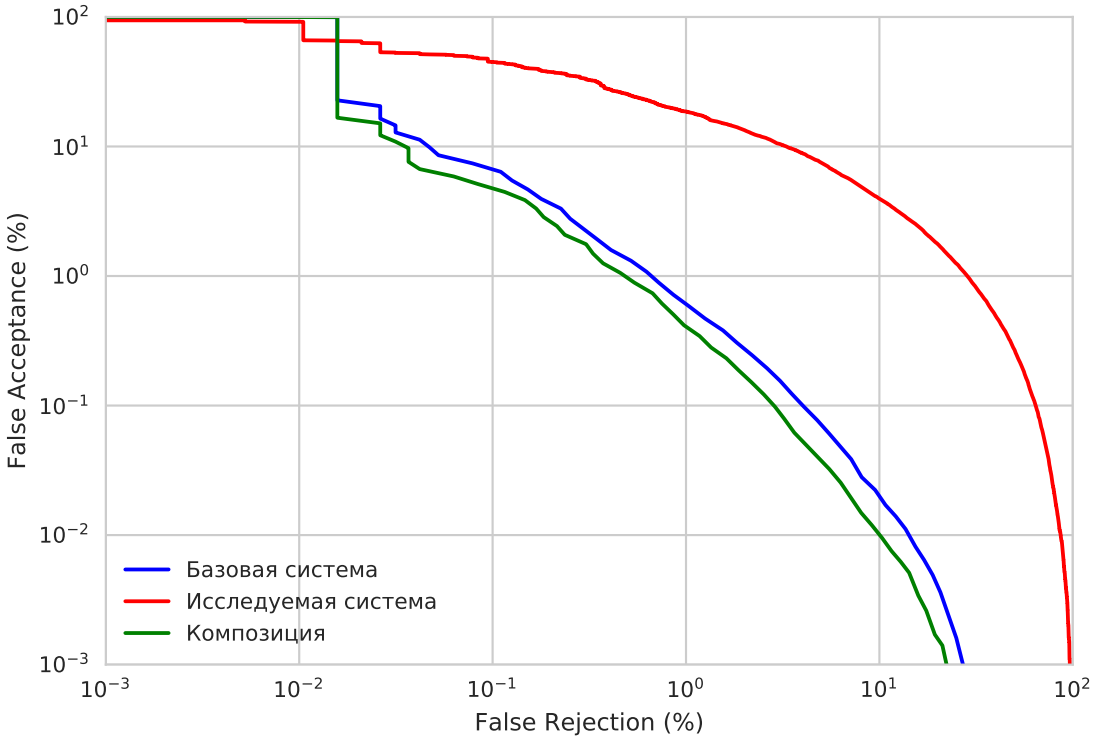

Рис. 11: DET-кривая результатов на базе RSR2015

был проведён на данной базе. Результат обучения модели на расширенном background множестве отражён в таблице под названием «Исследуемая система (расширеная)». Рисунок 12 иллюстрирует PCA-проекцию в трёхмерное пространство высокоуровневых векторов признаков, полученных с помощью глубокой нейронной сети.

В таблице 3 отражены результаты экспериментов на базе NIST, рисунок 13 отображает соответствующую DET-кривую.

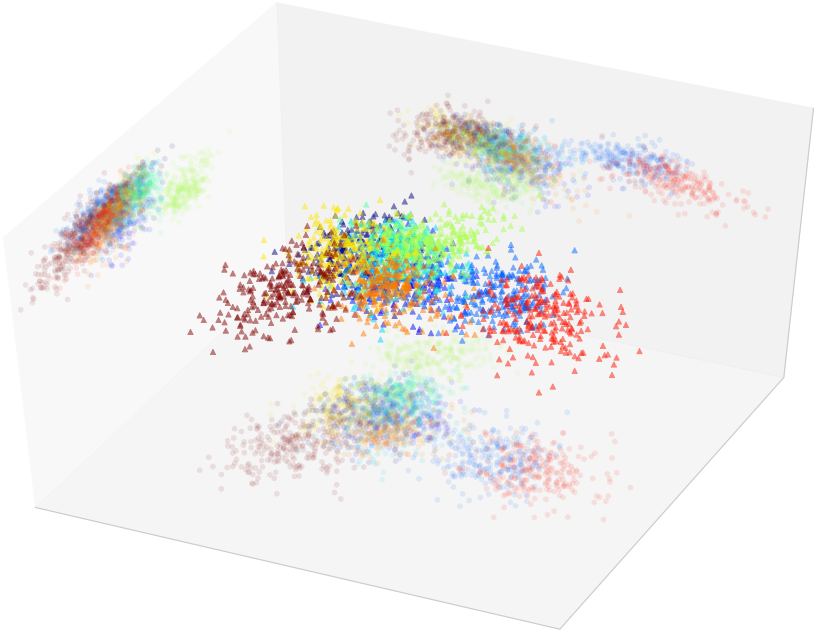

Рис. 12: PCA-проекция в трёхмерное пространство векторов признаков десяти дикторов, извлечённых исследуемой глубокой архитектурой

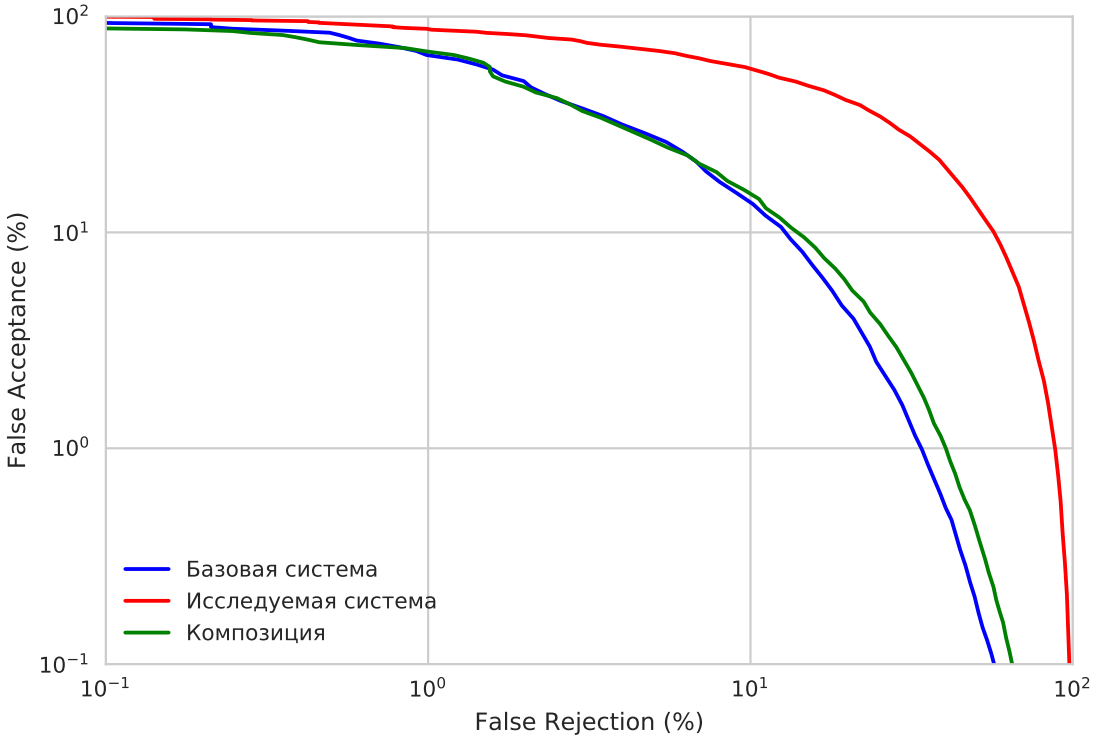

Рис. 13: DET-кривая результатов на базе NIST

Таблица 3: Результаты на базе NIST

Система	EER (%)
Базовая система	12.45
Исследуемая система	29.40
Композиция	**12.00**

4.3 Анализ результатов

Базовая система продемонстрировала высокое качество на тестовой части базы RSR2015, достигнув значения EER меньше одного процента. Качество же исследуемой модели оказалось хуже в 7.5 раз, достигнув 6.02% EER. Такой результат может быть объяснён недостаточным объёмом обучающей выборки. Глубокие нейронные сети в смежной задаче распознавания субъекта по лицу, например, обучаются на выборках из десятков тысяч различных субъектов. Хоть база RSR2015 и содержит достаточно большое количество фонограмм, её дикторская вариативность крайне мала и ограничена 300 дикторами. Для проверки выдвинутой гипотезы был проведён эксперимент на расширенной обучающей выборке с вдвое большим числом дикторов, который показал результат, равный 5.23% EER. Подобное улучшение свидетельствует о том, что дикторская вариативность важна в обучении глубоких архитектур для текстозависимой задачи.

Композиция базовой и исследуемой моделей достигла результата в 0.64% EER, улучшив результат базовой модели на 19% относительно. Этот факт свидетельствует о декоррелированности признаков, извлекаемых разными моделями и о наличии в обученной глубокой нейросетевой модели информации, важной для разделения дикторов, но при этом не присутствующей в базовой модели.

Результаты на текстонезависимой базе NIST соотносятся с результатами на RSR2015. Исследуемая модель достигла 29.40% EER, что в 2.3 раза хуже базовой модели, продемонстрировавшей 12.45% EER. Композиция систем улучшила результат базовой системы на 4% относительно. Результаты текстонезависимого эксперимента оказались в целом хуже результатов текстозависимого, что может быть объяснено её большой лингвистической вариативностью и меньшим при этом количеством фонограмм.

Выводы

По результатам исследования могут быть сделаны следующие выводы:

1. Глубокая свёрточная нейросетевая архитектура может быть успешно использована в задачах автоматического текстозависимого и текстонезавсисимого распознавания диктора по голосу.

2. Спектрограммы могут быть успешно применены в качестве исходных низкоуровневых признаков для рассматриваемых задач.

3. Возможно построить систему автоматического распознавания диктора по голосу, не внося в неё большого числа априорной информации о речи, звуке, восприятии звука человеком и о разложимости высокоуровневого представления фонограммы на дикторскую и канальную составляющие.

4. Модель на основе глубоких свёрточных нейронных сетей может улучшить результаты базовой системы в обоих рассматриваемых задачах.

Заключение

В рамках исследования были выполнены все поставленные задачи и подтверждены выдвинутые гипотезы. Система распознавания диктора по голосу действительно может быть построена на основе глубоких свёрточных нейронных сетей и использовать спектрограммы в качестве входных низкоуровневых признаков. Не смотря на то, что такая модель проявляет себя хуже базовой, их композиция позволяет улучшить результат базовой модели на 19% и 4% относительно в текстозависимой и текстонезависимой задачах соответственно. Также было замечено, что увеличение объёма обучающей выборки улучшает результат в текстозависимой задаче. Предполагается, что, имея базу с достаточно высокой дикторской и внутридикторской вариативностями, возможно построить систему на основе глубоких свёрточных нейронных сетей, которая превзойдёт базовую систему по качеству. Проверка этого предположения может стать темой дальнейших исследований. По результатам работы была написана и предложена к публикации на международной конференции SPECOM 2017 научная статья [39].

Список литературы

[1] Reynolds D. A., Quatieri T. F., Dunn R. B. Speaker verification using adapted Gaussian mixture models //Digital signal processing. – 2000. – Т. 10. – №. 1-3. – С. 19-41.

[2] Larcher A. et al. RSR2015: Database for Text-Dependent Speaker Verification using Multiple Pass-Phrases //INTERSPEECH. – 2012. – С. 1580-1583.

[3] Garofolo J. S. et al. DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1 //NASA STI/Recon technical report n. – 1993. – Т. 93.

[4] NIST Speaker Recognition Evaluation // National Institute of Standards and Technology URL: National Institute of Standards and Technology | NIST (дата обращения: 01.04.2017).

[5] Cieri C., Miller D., Walker K. The Fisher Corpus: a Resource for the Next Generations of Speech-to-Text //LREC. – 2004. – Т. 4. – С. 69-71.

[6] Welch P. The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms //IEEE Transactions on audio and electroacoustics. – 1967. – Т. 15. – №. 2. – С. 70-73.

[7] Freeman R. L. Telecommunication transmission handbook. – Wiley-Interscience, 1981. – Т. 100.

[8] LeCun Y. et al. Gradient-based learning applied to document recognition //Proceedings of the IEEE. – 1998. – Т. 86. – №. 11. – С. 2278-2324.

[9] He K. et al. Deep residual learning for image recognition //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2016. – С. 770-778.

[10] He K. et al. Identity mappings in deep residual networks //European Conference on Computer Vision. – Springer International Publishing, 2016. – С. 630-645.

[11] Jones E., Oliphant T., Peterson P. SciPy: open source scientific tools for Python. – 2014.

[12] He K. et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification //Proceedings of the IEEE international conference on computer vision. – 2015. – С. 1026-1034.

[13] Nair V., Hinton G. E. Rectified linear units improve restricted boltzmann machines //Proceedings of the 27th international conference on machine learning (ICML-10). – 2010. – С. 807-814.

[14] Maas A. L., Hannun A. Y., Ng A. Y. Rectifier nonlinearities improve neural network acoustic models //Proc. ICML. – 2013. – Т. 30. – №. 1.

[15] Zeiler M. D. et al. On rectified linear units for speech processing //Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. – IEEE, 2013. – С. 3517-3521.

[16] Szegedy C. et al. Rethinking the inception architecture for computer vision //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2016. – С. 2818-2826.

[17] Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition //arXiv preprint arXiv:1409.1556. – 2014.

[18] Ioffe S., Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift //arXiv preprint arXiv:1502.03167. – 2015.

[19] Zeinali H. et al. Deep Neural Networks and Hidden Markov Models in i-vector-based Text-Dependent Speaker Verification //Odyssey-The Speaker and Language Recognition Workshop. – 2016.

[20] Dehak N. et al. Front-end factor analysis for speaker verification //IEEE Transactions on Audio, Speech, and Language Processing. – 2011. – Т. 19. – №. 4. – С. 788-798.

[21] Kenny P. et al. A study of interspeaker variability in speaker verification //IEEE Transactions on Audio, Speech, and Language Processing. – 2008. – Т. 16. – №. 5. – С. 980-988.

[22] Lei Y. et al. A novel scheme for speaker recognition using a phonetically-aware deep neural network //Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. – IEEE, 2014. – C. 1695-1699.

[23] Novoselov S. et al. Non-linear PLDA for i-vector speaker verification //INTERSPEECH. – 2015. – C. 214-218.

[24] Kudashev O. et al. Usage of DNN in speaker recognition: advantages and problems //International Symposium on Neural Networks. – Springer International Publishing, 2016. – C. 82-91.

[25] McLaren M., Lei Y., Ferrer L. Advances in deep neural network approaches to speaker recognition //Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on. – IEEE, 2015. – C. 4814-4818.

[26] Stafylakis T. et al. Text-dependent speaker recognition using PLDA with uncertainty propagation //matrix. – 2013. – T. 500. – C. 1.

[27] Larcher A. et al. Text-dependent speaker verification: Classifiers, databases and RSR2015 //Speech Communication. – 2014. – T. 60. – C. 56-77.

[28] Aronowitz H. Text dependent speaker verification using a small development set //Odyssey 2012-The Speaker and Language Recognition Workshop. – 2012.

[29] Novoselov S. et al. Text-dependent GMM-JFA system for password based speaker verification //Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. – IEEE, 2014. – C. 729-737.

[30] Novoselov S. et al. Plda-based system for text-prompted password speaker verification //Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on. – IEEE, 2015. – C. 1-5.

[31] Heigold G. et al. End-to-end text-dependent speaker verification //Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on. – IEEE, 2016. – C. 5115-5119.

[32] Zhang S. X. et al. End-to-End Attention based Text-Dependent Speaker Verification //arXiv preprint arXiv:1701.00562. – 2017.

[33] Li C. et al. Deep Speaker: an End-to-End Neural Speaker Embedding System //arXiv preprint arXiv:1705.02304. – 2017.

[34] Zhu Q., Alwan A. An efficient and scalable 2D DCT-based feature coding scheme for remote speech recognition //Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on. – IEEE, 2001. – T. 1. – C. 113-116.

[35] Grezl F., Fousek P. Optimizing bottle-neck features for LVCSR //Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on. – IEEE, 2008. – C. 4729-4732.

[36] Abadi M. et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems //arXiv preprint arXiv:1603.04467. – 2016.

[37] Chollet F. Keras. – 2015.

[38] Kingma D., Ba J. Adam: A method for stochastic optimization //arXiv preprint arXiv:1412.6980. – 2014.

[39] Malykh E. et al. On Residual CNN in text-dependent speaker verification task //Submitted to Specom. - 2017.