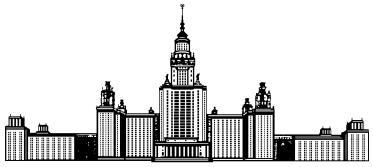


Московский государственный университет имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математических методов прогнозирования

УЛЬЯНОВ Дмитрий Владимирович

Исследование и разработка метода автоматической верификации личности на основе искусственных нейронных сетей

ДИПЛОМНАЯ РАБОТА

Научный руководитель:
к.ф.-м.н., в.н.с.
Чучупал Владимир Яковлевич

Москва, 2015

Содержание

1 Аннотация

В работе рассматривается задача верификации личности по голосу. Предлагается метод, в котором используются искусственные нейронные сети для построения модели диктора. В работе исследуется несколько алгоритмов принятия окончательного решения по построенным моделям дикторов. В качестве одного из алгоритмов сопоставления моделей используется метод, часто используемый в машинном зрении, однако редкий для задач распознавания диктора.

2 Проблема автоматического распознавания личности по голосу

За последние 50 лет было предложено большое количество способов снятия различных измерений и сигналов для систем биометрического распознавания личности. Одни из самых часто используемых способов распознавания – по отпечатку пальца, по снимку лица, сетчатке глаза, по голосу. У каждого метода существуют свои как положительные, так и отрицательные стороны. Например, распознавание личности по сетчатке глаза может быть произведено с большой точностью, но также требует больших затрат на оборудование, предъявляются жесткие требования к качеству данных, к действиям пользователя. Использование речи человека для задач распознавания имеет большой потенциал, так как единственным требованием, предъявляемым к оборудованию является наличие микрофона, а уровень сигнал/шум записи может быть довольно низок. Такие системы верификации и идентификации легко могут быть внедрены и использоваться как на телефонных линиях, так и на смартфонах и планшетах.

2.1 Задачи идентификации и верификации диктора

Задачи распознавания диктора делятся на два класса: задачи верификации и идентификации.

В задаче *идентификации* неизвестный диктор сравнивается с каждым из L имеющихся в базе данных дикторов. Ответом считается диктор из базы, наиболее похо-

жий на неизвестного. Такая постановка задачи идентификации называется задачей *идентификации с закрытым множеством решений*. В задаче *идентификации с открытым множеством решений* возможен отдельный исход: «диктор не является ни одним из известных». Такая задача сложна тем, что кроме определения наиболее похожего диктора требуется понять, входит ли вообще неизвестный диктор в множество дикторов базы данных.

Задача верификации диктора заключается в определении принадлежности данной записи проверяемому диктору. Задача верификации может рассматриваться как частный случай идентификации с открытым множеством, в которой число дикторов в базе данных $L = 1$.

Задачи распознавания также делят на две категории. В *текстозависимой* задаче текст, произносимый диктором известен заранее, в то время как в *текстонезависимой* задаче никаких предположений о тексте не делается. Текстозависимые системы верификации теоретически могут давать лучшие результаты, так как в этом случае с эталоном могут сравниваться как голос, так и текст.

2.2 Системы распознавания дикторов

Независимо от задачи (верификация, идентификация), работа системы делится на 2 этапа: настройка и распознавание.

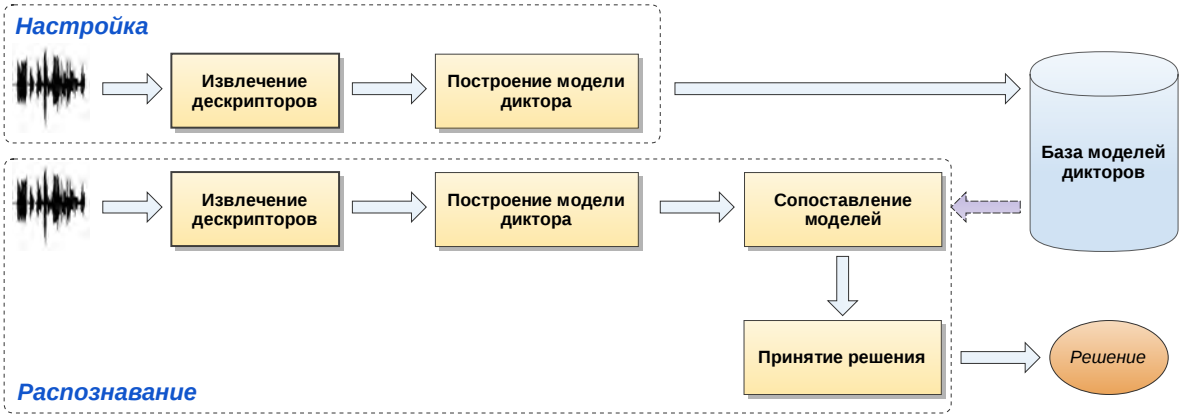


Рис. 1: Типовая схема системы распознавания диктора.

На *этапе настройки* система формирует модель диктора. На этапе распознавания модель диктора, построенная по входящей записи, сравнивается с моделями в базе данных.

Для построения модели диктора требуется найти компактное представление сигналов, такое, что сигналы с речью одного диктора оказывались бы «похожи» в некотором смысле, а сигналы разных дикторов «различны» в этом представлении. Использование компактного представления естественным образом обусловлено избыточностью исходных данных. Компактным представлением обычно служит последовательность дескрипторов небольшого размера, извлеченных из частей сигнала. По дескрипторам, в свою очередь, строится модель диктора. Основная сложность задач верификации и идентификации с открытым множеством решений – научиться строить модель диктора так, чтобы модель обладала дискриминативной способностью в том числе и для дикторов, не входящих в обучающую выборку.

На *этапе распознавания* из речи неизвестного диктора извлекается последовательность дескрипторов, строится модель диктора. Эта модель сравнивается со всеми моделями, находящимися в базе данных, с помощью некоторого алгоритма оценки близости. Полученные таким образом оценки используются далее для принятия окончательного решения. Окончательным решением для задачи верификации является один из двух возможных вариантов: подтверждение личности диктора или отвержение диктора. В задаче идентификации с закрытым множеством ответом считается наиболее похожий диктор из множества известных, в то время как в задаче с открытым множеством дополнительно вводится ответ «диктор не найден в базе».

3 Методы автоматической верификации личности говорящего

3.1 Признаковое описание речевых сигналов

Для работы с сигналами переменной длины невозможно напрямую применить классический аппарат машинного обучения. Для того, чтобы с сигналом можно было работать стандартными средствами, от сигнала вычисляется некоторое признаковое описание фиксированного размера. При этом признаковое описание всего сигнала обычно ищется как функция от признаковых описаний маленьких фрагментов сигнала, называемых дескрипторами (Рис. 2).

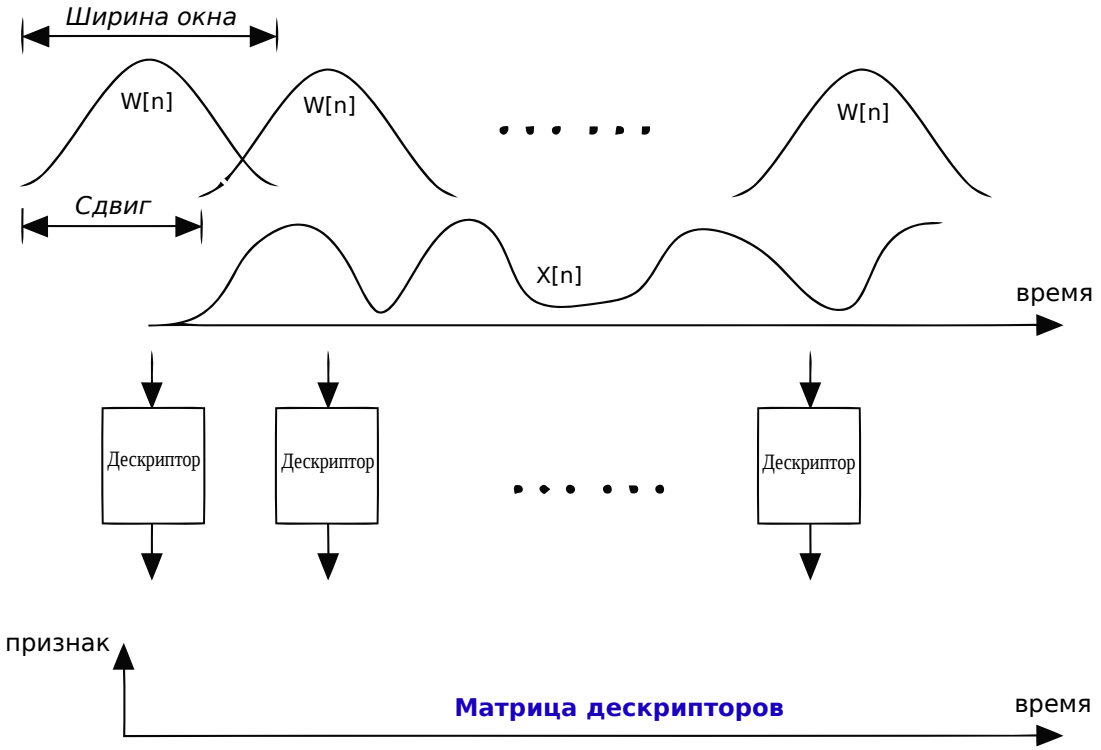


Рис. 2: Матрица дескрипторов.

Для того чтобы статистическая модель обладала обобщающей способностью, число объектов обучения должно быть достаточно большим по сравнению с количеством признаков. Известно, что количество необходимых объектов зависит экспоненциально от размерности признаков (этот эффект называется эффектом «проклятья размерности» [9]). Таким образом другой важной причиной использования признакового

описания сигнала вместо «сырого» сигнала является компрессия данных до значимой для задачи информации и сокращение размера модели, уменьшение эффекта «проклятья размерности».

В задаче распознавания диктора естественно потребовать от извлекаемого признакового описания сигнала следующих свойств. Признаки должны быть «похожи» для любых записей одного диктора, при этом сильно различаться для разных дикторов. Эти свойства являются необходимыми для того, чтобы было возможно построить качественный классификатор, разделяющий дикторов базы данных. Признаки должны быть легко вычислимы, так как процедура извлечения признаков может повторяться многократно. Признаки должны быть устойчивыми к шумам, должны отличать настоящего диктора и подражание диктору. Признаки должны быть как можно более независимыми друг от друга.

В задачах распознавания диктора часто используются те же признаки, что и в задаче распознавания речи. Примером могут служить мел-кепстральные коэффициенты (MFCC), коэффициенты линейного предсказания [3]. Интересно, что успешное применение MFCC в задаче распознавания речи, часто объясняется «дикторонезависимостью» этих признаков. Однако оказывается, что эти же признаки могут быть использованы и для распознавания диктора, что ставит под сомнения данную гипотезу.

3.2 Модели диктора

Существует два основных подхода к построению модели диктора по набору дескрипторов сигнала с речью: параметрический и непараметрический. В рамках *параметрического подхода* плотность распределения дескрипторов обучающей выборки приближается некоторым распределением, которое выбирается из некоторого параметрического семейства распределений посредством максимизации определенного критерия. Методы *непараметрического подхода*, в свою очередь, делают минимум предположений о плотности распределения дескрипторов (или не делают их вовсе).

3.2.1 Метод векторного квантования

Метод векторного квантования предполагает разбиение множества дескрипторов каждого диктора на небольшое число кластеров. Совокупность средних дескрипторов каждого кластера (центроид) $\{\vec{c_1}, \vec{c_2}, \ldots, \vec{c_k}\}$ называется кодовой книгой и представляет собой модель диктора. Вектора кодовой книги подчиняются тому же закону распределения, что и дескрипторы, поэтому кодовую книгу можно рассматривать как сжатое представление последовательности дескрипторов записи диктора. Размер кодовой книги является структурным параметром алгоритма.

Предложено несколько различных методов составления кодовой книги, которые можно разделить на два типа. В методах первого типа кодовая книга каждого диктора вычисляется независимо от кодовых книг других дикторов. Самым популярным алгоритмом такого типа является обобщенный алгоритм Ллойда (GLA)[16] также известный как алгоритм Линде-Бузо-Грея (LBG). Этот алгоритм очень похож на метод K-средних [7]. На практике выбор конкретного алгоритма кластеризации не сильно влияет на ошибку распознавания [13]. Методы второго типа стараются учитывать кодовые книги других дикторов при построении кодовой книги для данного диктора. Метод группового векторного квантования (GVQ) [8] сначала строит кодовые книги для каждого диктора независимо, после чего пересчитывает их так, чтобы кодовые книги разных дикторов как можно больше различались.

Для нахождения ближайшего диктора к данному используется функция «шум квантования». Пусть $X = \{\vec{x_1}, \ldots, \vec{x_T}\}$ – исследуемая последовательность дескрипторов, $C = \{\vec{c_1}, \ldots, \vec{c_K}\}$ – модель некоторого диктора. Шум квантования вектора x_i множеством C определяется как

$$d_q(\vec{x_i}, C) = \min_{\vec{c_j} \in C} d(\vec{x_i}, \vec{c_j}), \tag{1}$$

где d – некоторая мера близости, определенная в пространстве дескрипторов.

Средний шум квантования множества X множеством C определяется как:

$$D_Q(X, C) = \frac{1}{T} \sum_{i=1}^{T} d_q(\vec{x_i}, C). \tag{2}$$

3.2.2 Универсальная фоновая модель, GMM-UBM

Пусть требуется определить, является ли $\hat{Y}$ диктором фрагмента $X = \{\vec{x_1}, \ldots, \vec{x_T}\}$. Проверяется гипотеза H_0 против альтернативы H_1, где H_0 и H_1 определяются следующим образом:

$$H_0 : \hat{Y} \text{ является диктором } X, \tag{3}$$

$$H_1 : \hat{Y} \text{ не является диктором } X. \tag{4}$$

Для выбора между гипотезами используется тест отношения правдоподобия.

$$\frac{p(X|H_0)}{p(X|H_1)} \begin{cases} > \theta, & \text{принять } H_0, \\ < \theta, & \text{принять } H_1. \end{cases} \tag{5}$$

Предполагается ([20]):

$$p(X|H_0) = \prod_{i=1}^{T} p(x_i|\lambda_{H_0}) \tag{6}$$

где в качестве моделей распределения вероятностей $p(x|\lambda_{H_0})$ используется модель смеси гауссиан (GMM). Распределение $\log p(X|\lambda_{H_1})$ может быть определено различными способами. Например, используя модели других дикторов базы:

$$p(X|\lambda_{H_1}) = f(p(X|\lambda_1), \ldots, p(X|\lambda_l)), \tag{7}$$

где функция f задается, например, как максимум или минимум.

Подход универсальной фоновой модели (UBM) предполагает настройку одной модели $p(X|\lambda_{H_1})$, не зависящей от проверяемого диктора (универсальной для дикторов). Для этого выделяется отдельное множество дикторов, по записям которых строится распределение плотности вероятностей $p(X|H_1)$ UBM модели. Чаще всего искомое распределение аппроксимируется смесью гауссиан.

Понятно, что информации, извлеченной из короткой записи может не хватить для устойчивой настройки модели $p(X|H_0)$. В работе [19] предложено настраивать модель диктора $p(X|H_0)$, адаптируя UBM модель $p(X|H_1)$ с помощью метода, называемого MAP адаптацией. где m дикторонезависимая компонента, а T – некоторая низкоранговая матрица. Идентификационным вектором называется полученный вектор w. Идентификационные векторы могут быть впоследствии преобразованы с

целью компенсации различий между сессиями. Сравнение идентификационных векторов обычно выполняется с помощью косинусного расстояния или с использованием аппарата вероятностного линейного дискриминантного анализа.

3.2.3 Глубинные вектора, d-vector

В [22] был предложен метод текстозависимой верификации диктора основанный на нейронных сетях. Было предложено использовать глубинную (deep) нейронную сеть для извлечения дескрипторов.

На подмножестве обучающей выборки обучается глубинная нейронная сеть, в качестве объектов используются окна $O = \{\vec{o_1}, \ldots, \vec{o_T}\}$ MFCC последовательности, целевым признаком является идентификатор диктора записи, которой принадлежит окно. Последовательность дескрипторов $X = \{\vec{x_1}, \ldots, \vec{x_T}\}$ получается путем извлечения активаций последнего скрытого слоя нейросети (см. 4.2.4). Модель диктора рассчитывается как среднее всех дескрипторов, и называется глубинным вектором (d-vector). Модели сравниваются аналогично другим методам.

3.3 Принятие решений

Последним этапом в задаче распознавания диктора является принятие решения. Пусть $\hat{M}$ – модель неизвестного диктора, $S = \{M_1, \ldots, M_L\}$ – модели известных дикторов. Обозначим за $f(\hat{M}, M)$ некоторую функцию оценки близости модели $\hat{M}$ к модели M.

В задаче идентификации с закрытым множеством решений ответ строится с помощью простого правила:

$$y^* = \arg\min_l f(\hat{M}, M_l). \tag{8}$$

Например для модели диктора, полученной с помощью метода векторного квантования, чем «лучше» описывается множество X множеством C, тем меньше средний шум квантования. Ближайшим диктором считается диктор с моделью, доставляющей наименьший шум квантования: $y^* = \arg\min_l \sum_{i=1}^{T} d_q(\vec{x_i}, C_l)$.

В случае идентификации с открытым множеством решающее правило определяется как:

$$P = \{ \quad j \quad | \quad f(\hat{M}, M_j) \leq \theta_j, j = 1, \ldots, L\} \tag{9}$$

$$y^* = \begin{cases} \arg\min_l f(\hat{M}, M_l), & \text{если } |P| > 0, \\ \text{диктор не найден}, & \text{иначе.} \end{cases} \tag{10}$$

В задаче верификации возможны лишь два исхода:

$$y^* = \begin{cases} \text{подтвердить}, & \text{если } f(\hat{M}, M) \leq \theta, \\ \text{отказать}, & \text{иначе.} \end{cases} \tag{11}$$

В d-vector системе в качестве функции f используется косинусное расстояние, определяемое как

$$d(\vec{u}, \vec{v}) = 1 - \frac{< \vec{u}, \vec{v} >}{||\vec{u}|| ||\vec{v}||}. \tag{12}$$

Другими возможными функциями близости могут служить, например:

Расстояние Брэя-Кёртиса [2]

$$d(\vec{u}, \vec{v}) = \frac{\sum_i |u_i - v_i|}{\sum_i |u_i + v_i|}, \tag{13}$$

расстояние Канберры [15]

$$d(\vec{u}, \vec{v}) = \sum_i \frac{|u_i - v_i|}{|u_i| + |v_i|}, \tag{14}$$

Евклидово расстояние

$$d(\vec{u}, \vec{v}) = ||\vec{u} - \vec{v}||_2. \tag{15}$$

4 Необходимые понятия

4.1 Обработка сигнала

4.1.1 Дискретное Преобразование Фурье

Для анализа сигналов часто используется преобразование Фурье. Это преобразование позволяет перейти из временной области в частотную, для анализа частотной характеристики сигнала.

Для дискретных аудиосигналов применяется *Дискретное Преобразование Фурье* (ДПФ). Для данного сигнала $s[n]$ преобразование Фурье задается следующим образом.

$$\hat{S}[k] = F_{s[n]} = \sum_{n=0}^{N-1} s[n] e^{-2\pi i k \frac{n}{N}}, \quad 0 \leq k \leq N-1. \tag{16}$$

Дискретное преобразование Фурье является линейным преобразованием и обратимо. Сложность преобразования в соответствии с формулой составляет $O(N^2)$, существует алгоритм, позволяющий вычислить ДПФ за $O(Nlog_2N)$, что позволяет вычислять преобразование в реальном времени.

4.1.2 Оконное Дискретное Преобразование Фурье, ОДПФ

Если характеристики сигнала в значительной степени меняются со временем, сигнал следует разбить на маленькие фрагменты, так чтобы параметры каждого фрагмента можно было бы считать стационарными (не зависящими от времени). Вычисление ДПФ от всего речевого сигнала приводит к потере локальных особенностей сигнала. Чтобы сохранить локальную информацию используется *Оконное Дискретное Преобразование Фурье* (ОДПФ). Фиксируется ширина окна (выбирается в зависимости от свойств сигнала, обычно от 10 до 100 миллисекунд). К скользящему окну применяется оконная функция w и вычисляются модули коэффициентов ДПФ. Процесс повторяется для всего речевого сигнала вместе с движением окна (окно обычно двигается на 25-75 процентов своей ширины), см. Рис. 3.

$$\text{ОДПФ}[t, f] = \sum_{n=-\inf}^{\inf} s[n] w[n-m] e^{-jfn}. \tag{17}$$

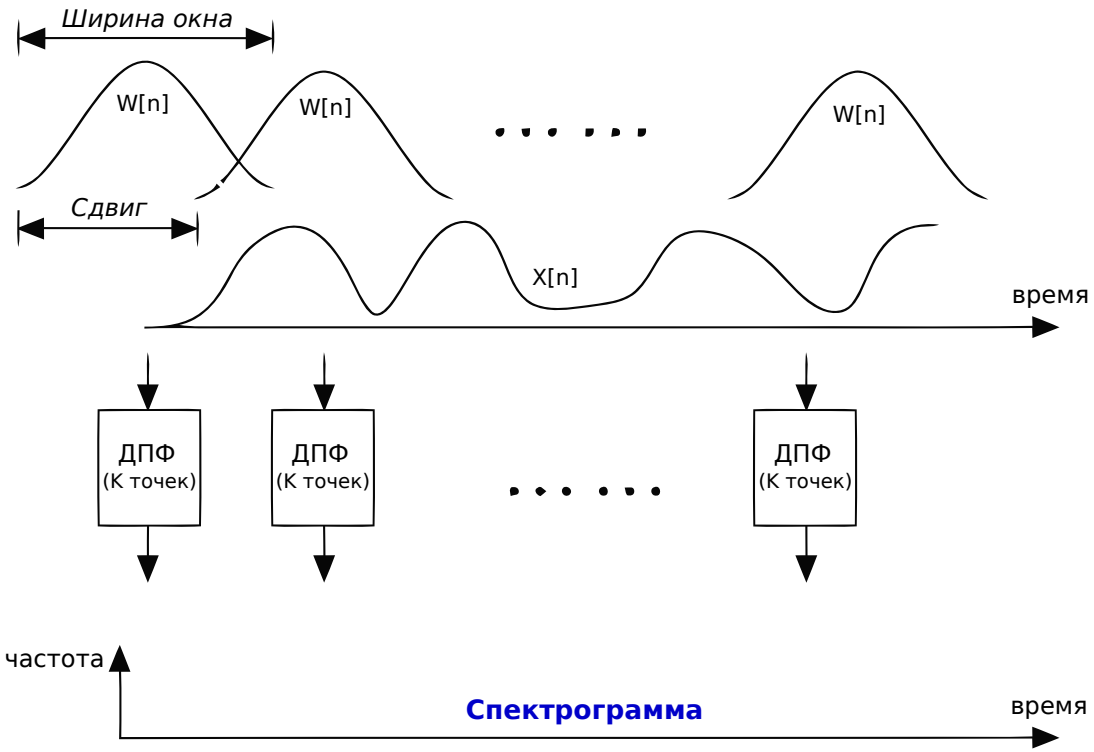


Рис. 3: Получение спектрограммы сигнала.

При определенных ограничениях на оконную функцию ОДПФ преобразование обратимо.

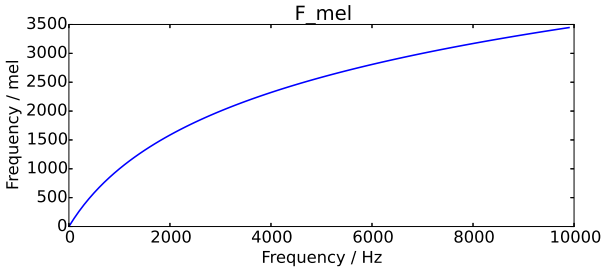
Такой вид сигнала называется спектрограммой сигнала. На практике обычно работают с логарифмом спектрограммы, что примерно соответствует восприятию громкости человеком.

4.1.3 Мел-спектрограмма

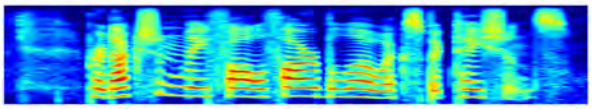
Известно, что разрешающая способность человеческого уха нелинейна по частотам [4]. Исследования показали, что на высоких частотах фундаментальная частота должна измениться сильнее, чтобы человек заметил различие. Единицей психофизической частоты называется *мел*. Формула, связывающая частоту в герцах и мелах, была получена на основе экспериментальных данных, поэтому существует несколько определений, отличающихся константами. Для примера в [5] предложен такой вид функции преобразования:

$$F_{mel} = 1000 \log_2(1 + \frac{F_{Hz}}{1000}). \tag{18}$$

Это преобразование применяется к спектрограмме сигнала для получения *мел-спектрограммы*.



(a) Функция преобразования частоты в мелы.



(b) Типичный вид мел-спектрограммы.

4.1.4 Мел-кепстральные коэффициенты

Мел-кепстральные коэффициенты[18] замечательно зарекомендовали себя в качестве признаков в задачах распознавания и анализа речи. Алгоритм получения мел-кепстральных коэффициентов состоит из 5 основных шагов.

1. Разбиение сигнала на окна.

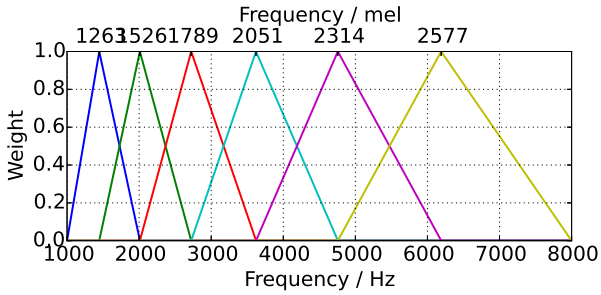
2. Получение модулей коэффициентов ДПФ.

3. Преобразование в мел пространство частот.

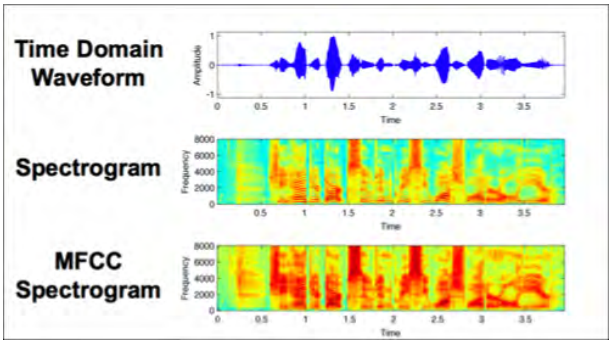
4. Сглаживание применением банка фильтров.

5. Применение дискретного косинусного преобразования.

Первые два пункта представляют собой вычисление ОДПФ. Преобразование в мел-пространство описано в параграфе 4.1.3. Применение банка фильтров обусловлено необходимостью сокращения размерности и сглаживания признаков. Дискретное косинусное преобразование применяется для уменьшения коррелированности признаков.



(a) Типичный вид банка мел фильтров.



(b) Визуализация различных признаковых описаний сигнала.

4.2 Машинное обучение

4.2.1 Классическая постановка

Кратко дадим основные понятия машинного обучения. Более подробная информация может быть найдена, например, в [23].

Пусть задано множество объектов X и множество допустимых ответов Y, также задана целевая функция $y^* : X \to Y$, значения которой известны на подмножестве объектов $x_1, ..., x_t \in X$, обозначим $y_i = y^*(x_i)$ Совокупность пар $X^l = (x_i, y_i)_{i=1}^l$ называется *обучающей выборкой*, пара (x_i, y_i) называется *прецедентом*.

Задача обучения по прецедентам заключается в восстановлении зависимости y^*, то есть построении функции $a : X \to Y$, приближающей целевую функцию.

Функцию a чаще всего задают в параметрическом виде $a(\Lambda)$, параметры Λ выбираются так, чтобы минимизировать некоторый функционал средней ошибки $Q(a, X^l) = \frac{1}{L}\sum_{i=0}^{L}\mathcal{L}(a, x_i)$. Таким образом решается задача $Q(a(\Lambda), X^l) \to \min_\Lambda$.

Функция потерь $\mathcal{L}$ задается в зависимости от задачи. Наиболее часто встречаемые функции потерь:

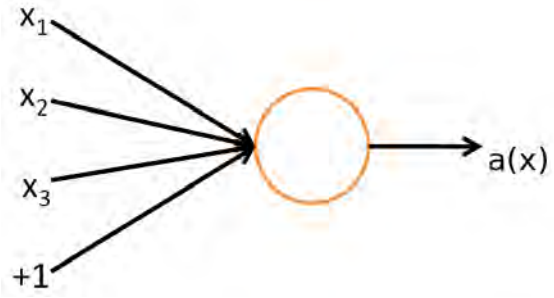
Среднеквадратичная ошибка, MSE: $\mathcal{L}(a, x) = (a(x) - y^*(x))^2$ для $Y \in \mathbb{R}$ (задача регрессии).

Логарифмические потери, LogLoss: $\mathcal{L}(a, x) = \sum_{i=1}^{N}[y^*(x) = i] \log a(x)$ для $Y = \{1, \ldots, N\}$, N – количество классов.

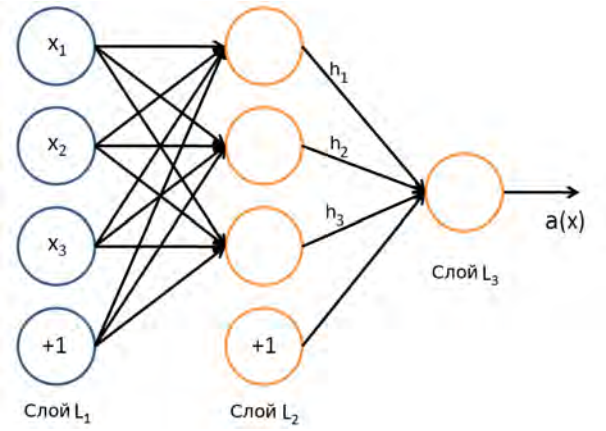
MSE также используется и для задач классификации, несмотря на то, что в этом случае отсутствует четкая интерпретация оптимизируемого функционала.

4.2.2 Искусственные нейронные сети

Пусть $a(x;w) = f(\sum_{i=1}^{N} x_i w_i + w_0)$, где f – некоторая функция, называемая функцией активации. Модель такого вида, где в качестве функции f берется функция Хевисайда была предложена Ф. Розенблатом в 1957 году и называется перцептроном (нейроном).



(a) Графическое представление перцептрона.



(b) Графическое представление нейронной сети.

Рассмотрим композицию нейронов. Пусть $h_i{}_{i=1}^{N}$ – перцептроны, т.е.

$$h_i(x;W) = f_1(\sum_{j=1}^{N} x_j W_{i,j} + W_{i,0}) \tag{19}$$

Композиция перцептронов строится следующим образом:

$$a(x;W^{(1)},w^{(2)}) = f_2(\sum_{i=1}^{N} h_i(x;W^{(1)})w_i^{(2)} + w_0^{(2)}) \tag{20}$$

$$h_i(x;W) = f_1(\sum_{j=1}^{N} x_j W_{i,j} + W_{i,0}) \tag{21}$$

Представленная на Рис. 6b конфигурация называется нейронной сетью с одним скрытым слоем. Скрытых слоев, как и выходных нейронов, может быть значительно больше чем один.

Для обучения нейронных сетей предложено большое количество методов, самый распространенный из них – метод обратного распространения ошибки [21]. В методе используется правило дифференцирования сложных функций для вычисления градиента, что позволяет пересчитать параметры за время, сходное с временем, требуемым для получения выходных значений нейросети.

Последние исследования показали, что конкретный вид функции активации не сильно влияет на итоговое качество нейронной сети, из-за чего на практике в качестве функций активаций используют легко вычислимые функции (но обязательно нелинейные). В настоящее время в качестве функции активации чаще всего используется Rectified Linear Unit (ReLU): $f(x) = max(x, 0)$

В задаче классификации с N классами классы $\{1, \ldots, N\}$ кодируются бинарными векторами $\bar{y}$ размера N, такими, что: $\bar{y}_i(y) = [i = y]$. Количество нейронов на последнем слое соответствует количеству классов, каждый нейрон выдает оценку принадлежности объекта к соответствующему классу. Для того чтобы выход нейронов последнего слоя являлся оценкой принадлежности к классам (то есть являлся некоторым распределением вероятностей), в качестве функции активации последнего слоя используют функцию *софтмакс* (softmax):

$$\text{softmax}(z) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \tag{22}$$

4.2.3 Сверточные нейросети

В работе [14] представлена структура и методы обучения нейросети, с помощью которых были достигнуты лучшие, на тот момент, результаты в задаче распознавания изображений. Позже было показано, что нейросети такой структуры могут быть использованы и в задачах распознавания речи, в том числе и в задачах распознавания диктора [17]. Заметим, что спектрограмму или последовательность MFCC признаков можно рассматривать как изображение, что позволяет использовать методы распознавания изображений и компьютерного зрения в задачах связанных со звуком.

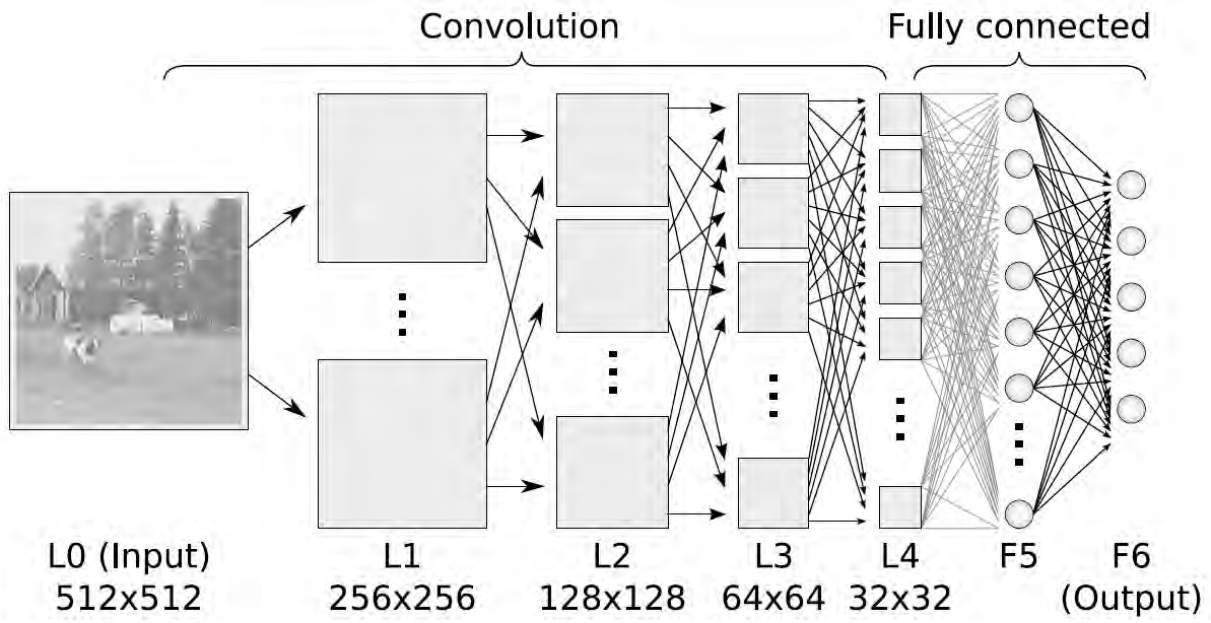


Рис. 7: Схема часто используемой конфигурации нейросети.

Структура предложенной в [14] нейросети выглядит следущим образом. Первая часть сети представляют собой комбинацию из сверточных, «выборочных» (pooling) и нормализационных слоев. Вторая часть состоит из комбинации полносвязных и дропаут (dropout) слоев. К выходу последнего слоя нейросети обычно применяется софтмакс функция.

4.2.4 Нейронные сети для извлечения признаков

Рассмотрим структуру сети и активации, полученные с K-ого скрытого слоя. Существует гипотеза, что активации обладают все большей дискриминативной способностью с увеличением K. Кроме того, можно заметить, что использование кросс-энтропийной функции потерь в паре с софтмакс активациями последнего слоя равноценно использованию логистической регрессии над активациями, полученными с последнего скрытого слоя нейросети. Таким образом, если нейросеть показывает хорошую точность, можно считать, что активации последнего слоя являются хорошими дискриминативными признаками объектов (заметим, что в определенных задачах в качестве признаков все-таки выбирают активации и других слоев).

4.3 Векторы локальных агрегированных дескрипторов

Рассмотрим задачу построения супервектора по набору дескрипторов объекта. В задачах распознавания образов в компьютерном зрении очень хорошо зарекомендовали себя методы, основанные на векторах локальных агрегированных дескрипторов (VLAD)[10]. VLAD идейно является расширением модели «мешок слов» (Bag of Words), которая предполагает кластеризацию пространства дескрипторов с последующим подсчетом количества дескрипторов относящихся к каждому кластеру. Очевидно, что просто подсчитывая количества дескрипторов, похожих на центры кластеров, информация о степени похожести теряется. В векторах VLAD учитывается не только информация о ближайших центрах кластеров, но также и информация о пространственном положении дескрипторов относительно центров.

Пусть пространство дескрипторов размерности N разбито на кластеры с центрами $C = \{\vec{c_1}, \ldots, \vec{c_K}\}$. Тогда VLAD(X,C) представляет из себя вектор размера KN. Удобно работать с VLAD как с матрицей $v \in \mathbb{R}^{K \times N}$ с последующим вытягиванием в вектор.

Для каждой центроиды (кодового элемента) $\vec{c_i}$ накапливается разность между центром кластера и дескриптором $\vec{x} - \vec{c_i}$, если данная центр является ближайшим к дескриптору ($\vec{c_i} = Nearest(\vec{x})$).

$$VLAD(\{\vec{x_1}, \ldots, \vec{x_T}\}, C)_{j,:} = \sum_{i=1}^{T} I[Nearest(\vec{x_i}) = \vec{c_j}](\vec{x_i} - \vec{c_j}). \tag{23}$$

В [1] обсуждается несколько способов улучшения техник на основе VLAD. В частности предлагается сначала нормализовать остатки для каждого кластера отдельно, а после этого применять нормализацию ко всему VLAD вектору.

Структурными параметрами алгоритма являются число кластеров, метрика, используемая для нахождения ближайшего кластера, метрика используемая при сравнении супервекторов.

5 Разработка метода верификации личности говорящего на основе нейронной сети

5.1 Обоснование

Нейронные сети отлично зарекомендовали себя в большом количестве задач, d-vector системы показали применимость нейросетей и в задаче текстозависимой верификации диктора.

В данной работе исследуется качество работы d-vector подобной системы в задаче текстонезависимой верификации. Проверяется, какие исходные признаки лучше использовать и каким образом следует сравнивать модели дикторов. В качестве одного из алгоритмов сравнения моделей дикторов используется метод, не используемый в задачах анализа речи.

5.2 Описание системы

Опишем используемую схему получения решения в задаче верификации. Входной сигнал преобразовывается в одно из двух представлений: мел-спектрограмма (6.2) или последовательность мел-кепстральных коэффициентов (6.1). MFCC признаки извлекаются из 25мс окон с помощью утилиты HTK, параметры представлены в таблице 8

Мел-спектрограмма извлекалась с помощью пакте LibRosa, использовалось 125мс окно с 4мс сдвигом. К спектрограмме сигнала применялись 227 мел фильтров, после чего последние 20 частот были отброшены, поскольку содержали только нули.

Для улучшения качества из последовательности вырезаются фрагменты с низкой энергией (такие фрагменты соответствуют тишине между словами). Используемое представление разбивается на окна и подается на вход нейронной сети. Нейронная сеть обучается предсказывать правильного диктора на тренировочной выборке, после чего последний (софтмакс) слой отбрасывается.

Тестовый сигнал разбивается на окна и прогоняется через нейронную сеть, таким образом получается последовательность дескрипторов, соответствующая тестовому сигналу. Эта последовательность используется для построения модели диктора –

используется один из двух алгоритмов: простое усреднение (6.1.1), VLAD (6.1.3). Модели сравниваются используя некоторую функцию близости или их комбинацию.

5.3 Экспериментальная установка

5.3.1 Речевой корпус

Для тестирования алгоритмов использовался речевой корпус TIMIT [6]. Корпус был подготовлен совместными усилиями компании Texas Instruments и института MIT. В нем представлено 6300 записей 630 дикторов. Вместе с корпусом подставляется разнородная метаинформация, не интересная в контексте решаемой задачи, но позволяющая использовать корпус и, например, для задач распознавания речи. В корпусе предусмотрено разбиение на тестовую и обучающую выборки, что в некоторых случаях упрощает сравнение результатов с результатами других работ.

5.3.2 Экспериментальный протокол

С точки зрения исследования и разработки в задаче верификации диктора обычно выделяется три этапа. Первый этап – обучение алгоритма верификации по размеченному корпусу данных (train). Второй – инициализация модели диктора, на которого необходимо настроиться (enroll) и третий этап – тестирование (test) системы на записях, как принадлежащих интересующему нас диктору, так и не принадлежащих.

Изначально корпус с записями делится на две непересекающиеся по дикторам части, чаще всего неравных размеров. Подвыборка большего размера используется на этапе обучения. Другая же часть используется для тестирования системы (enroll + test).

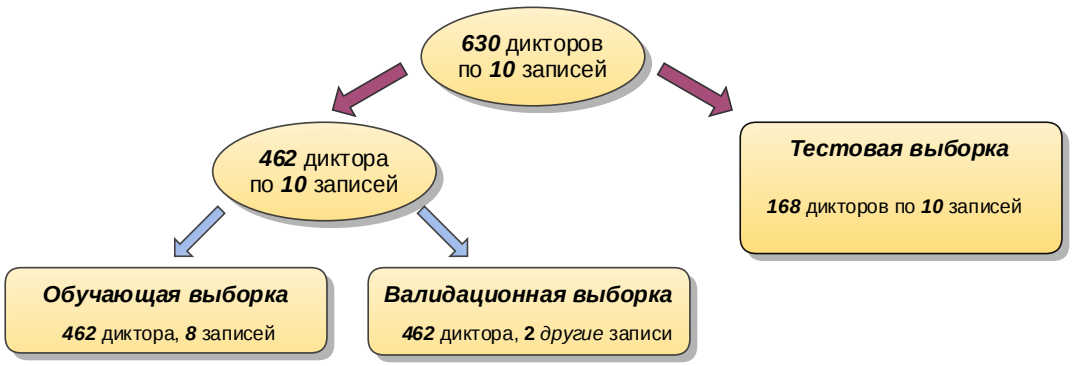


Рис. 8: Деление базы TIMIT на части.

Большая выборка в свою очередь делится еще на две части – обучающую и контрольную. Обучающая выборка используется для обучения используемого алгоритма получения дескрипторов, валидация (контроль) которого выполняется на контрольной выборке.

Деление на обучающую и контрольную выборку лучше производить не случайным образом, а таким, чтобы минимизировать вероятность настройки алгоритма не на диктора, а, например, на фразу, которую диктор произносит. Валидационная выборка была составлена из записей sa_1, sa_2, в то время как остальные 8 записей использовались в обучающей выборке.

5.3.3 Оценка системы

В работе приводятся результаты работы системы для двух случаев. В первом случае дикторы тестовой и обучающей выборок совпадают. Для этого тестирование системы производится на валидационной выборке (тестовая выборка не используется).

Во втором случае для тестирования используется тестовая выборка. Второй случай является практически значимым, так как показывает, насколько хорошо алгоритм работает на неизвестных на этапе обучения дикторах.

Сравнение результатов работы алгоритма в таких постановках позволяет анализировать обобщающую способность алгоритма.

5.3.4 Метрика качества

Для тестирования алгоритма формируется отдельные задачи оценки близости моделей дикторов, количество которых равно совпадает с количеством аудиозаписей в выбранной выборке тестирования и равно L. В рамках каждой задачи некоторая запись выбирается в качестве настроечной и тестируется против всех остальных. Таким образом получается матрица оценок $S \in \mathbb{R}^{L \times L}$ и соответствующая оценкам матрица истинных ответов $Y \in \mathbb{R}^{L \times L}$ (0 – «отказ», 1 – «допуск»).

Для примера:

Таблица 1: Матрица запись-диктор.

a	1
b	1
c	2

Таблица 2: Матрица оценок.

0	d(a,b)	d(a,c)
d(b,a)	0	d(b,c)
d(c,a)	d(c,b)	0

Таблица 3: Матрица истинных ответов.

1	1	0
1	1	0
0	0	1

Для выбранного порога θ: количество ложных отказов FR (False Rejection), доля ложных отказов FRR (False Rejection Rate), количество ложных допусков FA (False Acceptance) и доля ложных допусков FAR (False Acceptance Rate) могут быть вычислены как

$$FR(\theta) = \sum_{i=1}^{L} \sum_{j=i+1}^{L} [S_{ij} > \theta][Y_{ij} = 1], \qquad FRR(\theta) = \frac{FR(\theta)}{\sum_{i=1}^{L} [Y_{ij} = 1]}, \tag{24}$$

$$FA(\theta) = \sum_{i=1}^{L} \sum_{j=i+1}^{L} [S_{ij} < \theta][Y_{ij} = 0], \qquad FAR(\theta) = \frac{FA(\theta)}{\sum_{i=1}^{L} [Y_{ij} = 0]}. \tag{25}$$

$$\tag{26}$$

Таким образом может быть построена ROC (Receiver operating characteristic) кривая. DET (Detection error tradeoff) кривой называется ROC кривая в логарифмическом по обеим осям масштабе (с осями FAR,FRR) . DET кривая нагляднее показывает результаты экспериментов и позволяет легче проводить визуальный анализ.

На практике порог выбирается в зависимости от ограничений на FR и FA, в то время как в исследовании важно оптимизировать обе характеристики. Для оценки качества системы в целом, в качестве порога выбирается порог $\bar{\theta}$, при котором доля

ложных отказов равна доле ложных допусков. Таким образом определяется метрика EER (Equal Error Rate):

$$EER = FA(\bar{\theta}) = FR(\bar{\theta}). \tag{27}$$

5.3.5 Извлечение дескрипторов

В экспериментах использовались две конфигурации нейросети. Первая использовалась для извлечения дескрипторов из последовательности мел-кепстральных коэффициентов и представлена в таблице 4 (заметим, что все слои кроме одного используют линейные функции активации), вторая конфигурация применялась к окнам, полученным из мел-спектрограммы и совпадает с конфигурацией сети AlexNet[14] (реализация Caffe[11]) с уменьшенным до размера 1024 последним скрытым слоем (Таблица 5).

Таблица 4: Конфигурация первой нейросети.

#	Слой	Параметры
0	Данных	40x40
1	Сверточный	32 5x5 фильтров
2	Pooling	усредение области 3x3, выбор каждого второго
3	Сверточный	32 5x5 фильтров
4	Pooling	усредение области 3x3, выбор каждого второго
5	Сверточный + ReLU	64 5x5 фильтров
6	Pooling	усредение области 3x3, выбор каждого второго
7	Полносвязный	1000 нейронов
8	Dropout	$p = 0.5$
9	Полносвязный	200 нейронов
10	Dropout	$p = 0.5$
11	Софтмакс	462 выходных нейрона

Обучение нейросети происходило на тренировочной выборке с контролем функции потерь на валидационной выборке. Использовался критерий раннего останова для минимизации эффекта переобучения.

Таблица 5: Конфигурация второй нейросети.

#	Слой	Параметры
0	Данных	207x207
1	Сверточный + ReLU	96 11x11 фильтров, выбор каждого четвертого
2	Нормировочный + ReLU	по 5x5 областям
3	Pooling	усредение области 3x3, выбор каждого второго
4	Сверточный + ReLU	256 5x5 фильтров
5	Нормировочный + ReLU	по 5x5 областям
6	Pooling	усредение области 3x3, выбор каждого второго
7	Сверточный + ReLU	384 3x3 фильтров
7	Сверточный + ReLU	384 3x3 фильтров
7	Сверточный + ReLU	256 3x3 фильтров, выбор каждого второго
8	Нормировочный + ReLU	по 5x5 областям
9	Pooling	усредение области 3x3, выбор каждого второго
7	Полносвязный + ReLU	4096 нейронов
8	Dropout	$p = 0.5$
7	Полносвязный + ReLU	1024 нейронов
8	Dropout	$p = 0.5$
11	Софтмакс	462 выходных нейрона

6 Экспериментальная проверка эффективности метода верификации

6.1 Первая серия экспериментов

В данной серии экспериментов используется первая конфигурация нейросети, примененная к окнам размера 40x40 последовательности мел-кепстральных коэффициентов.

6.1.1 Сравнение функций близости

Исследуем поведение ERR в зависимости от конкретной функций близости. Будем использовать функции, описанные в 3.3.

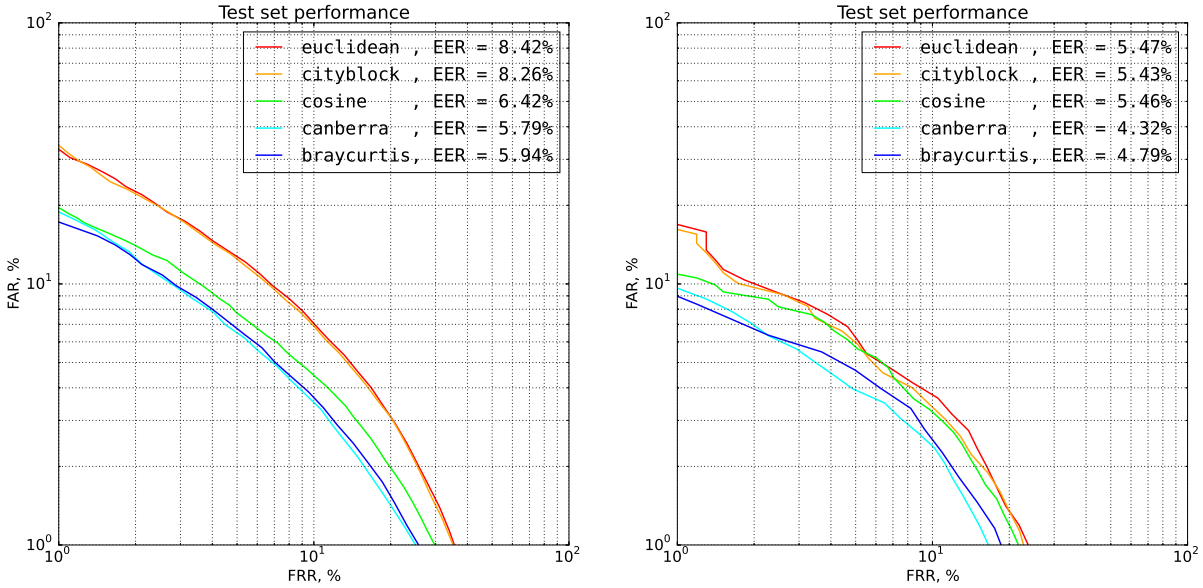


Рис. 9: Сравнение различных метрик на тестовой и валидационной выборках.

Обратим внимание, что функции «Канберра» и «Брэй-Кертис» дали наилучший результат. Обычно применяемое косинусное расстояние оказалось немного хуже.

6.1.2 Макс-мин эвристика

Автором был найден очень простой способ улучшить результат косинусного расстояния, не усложнив вычисления. Пусть требуется оценить близость моделей m_1 и m_2. Вместо прямого сравнения с помощью некоторой функции близости d: $d(m_1, m_2)$ будем вычислять оценку близости как

$$d_{max-min} = \frac{d(max(m_1, 0), max(m_2, 0)) + d(-min(m_1, 0), -min(m_2, 0))}{2}. \tag{28}$$

Результаты применения такого подхода показаны на Рис. 10.

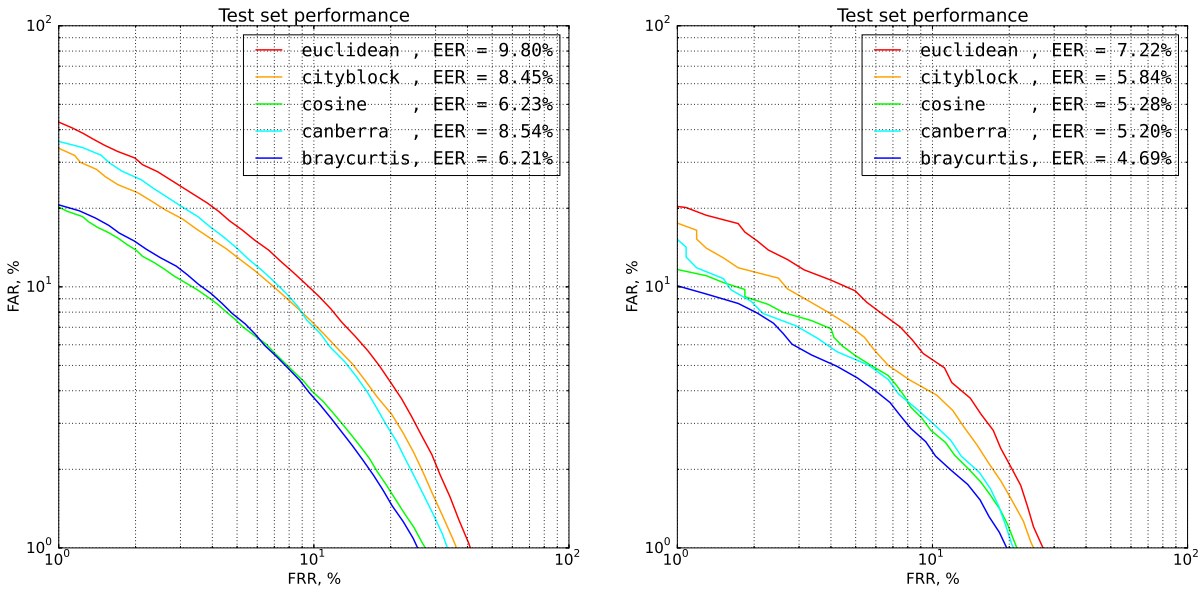


Рис. 10: Результаты применения макс-мин подхода.

Интересно, что положительный эффект наблюдается только при использовании косинусного расстояния. Использование данного макс-мин подхода только ухудшает результат других функций расстояния.

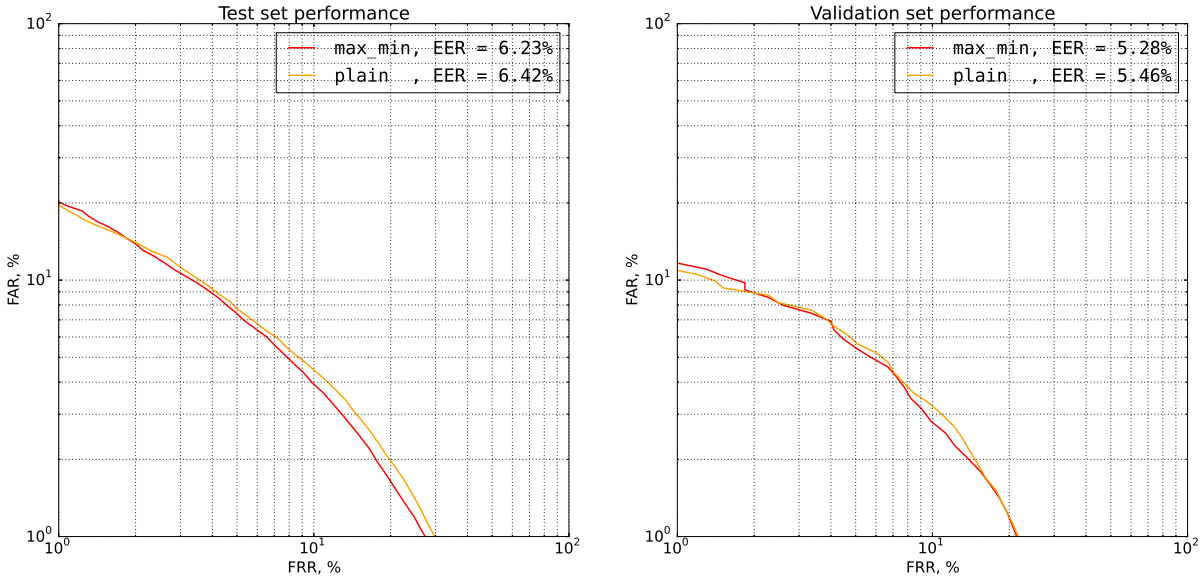


Рис. 11: Сравнение макс-мин подхода для косинусного расстояния.

6.1.3 Применение VLAD

Будем строить VLAD супервекторы вместо простого усреднения дескрипторов. Параметрами алгоритма является пара функций расстояния. Рис 12 показывает, что все комбинации метрик дают в общем либо «хороший» результат, либо «плохой».

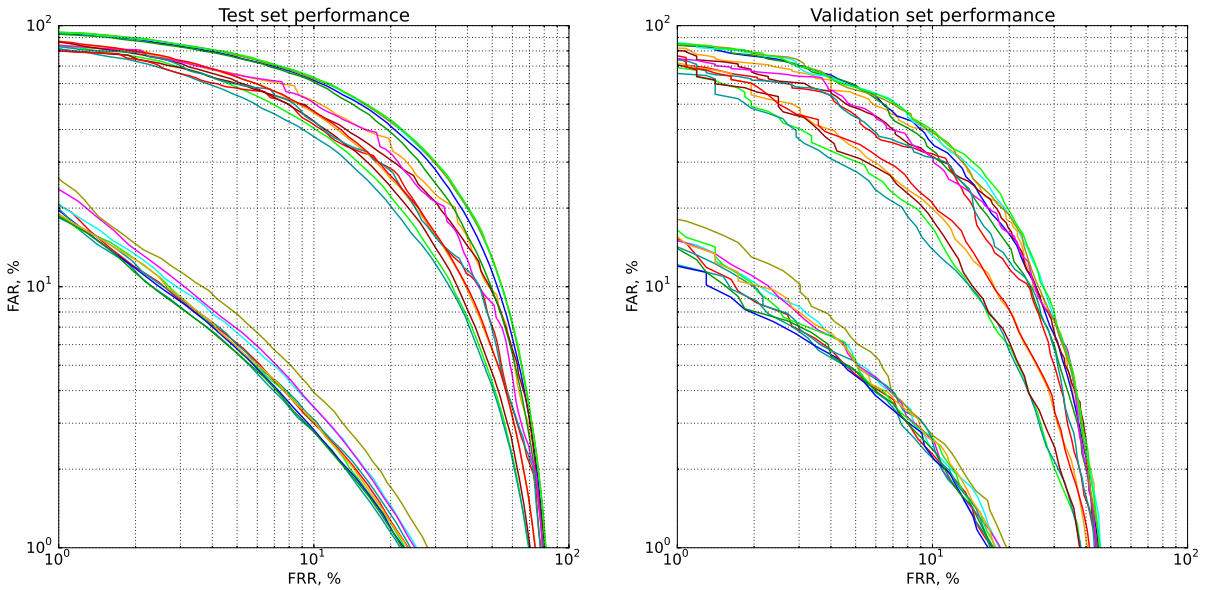


Рис. 12: DET кривые при использовании различных пар функций близости. Иллюстрируется дифференциация по качеству.

Группа пар метрик с лучшим качеством использует косинусное расстояние или расстояние Брэя-Кертиса в качестве второй функции меры близости. Конкретный вид первой функции близости не так важен. На рис 13 изображены 5 лучших комбинаций на тестовой и валидационной выборках.

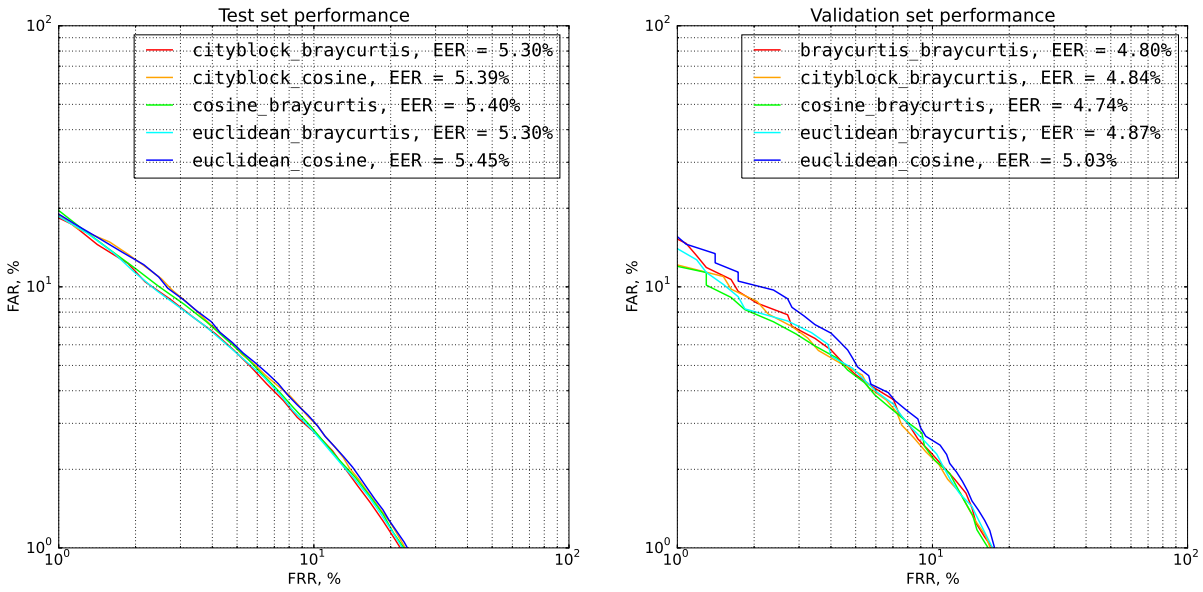
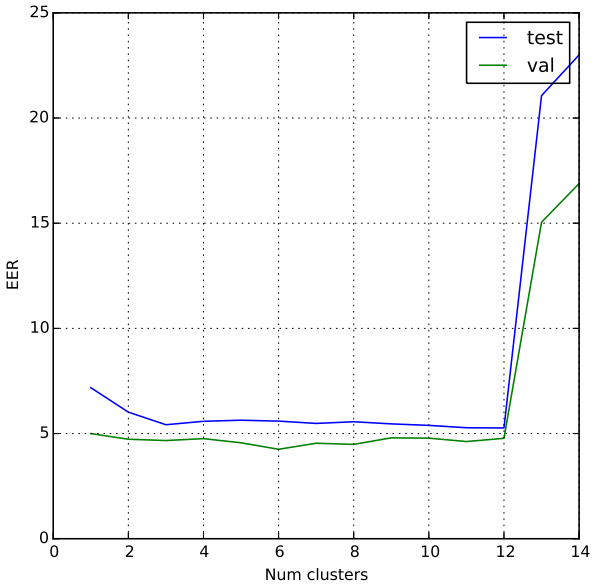
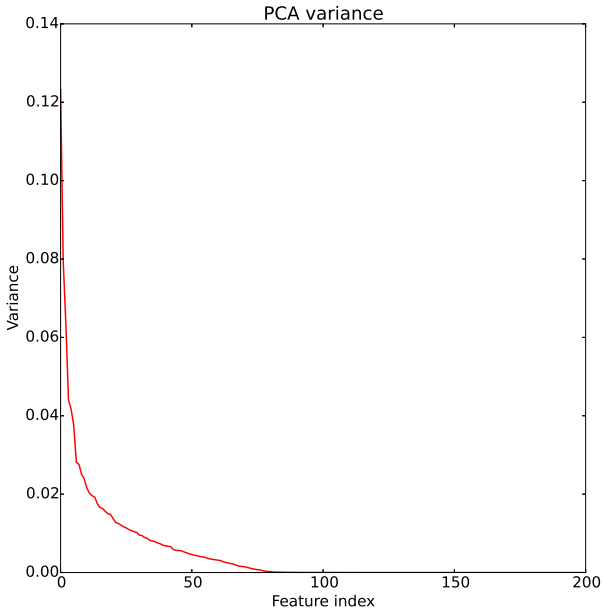


Рис. 13: Сравнение 5 лучших вариантов метрик.

Структурным параметром алгоритма является число кластеров, на которое разбивается пространство дескрипторов, в данной задаче лучшие результаты $ERR = 5.25$ на тестовой выборке достигались при $K = 3, \ldots, 11$ (Рис. 14a). При большем количестве кластеров результаты по какой-то причине резко портились, что, скорее всего, связано с необходимостью лучшей настройки метода кластеризации K-средних.



(a) EER в зависимости от числа кластеров.



(b) Анализ объясненной дисперсии для метода главных компонент.

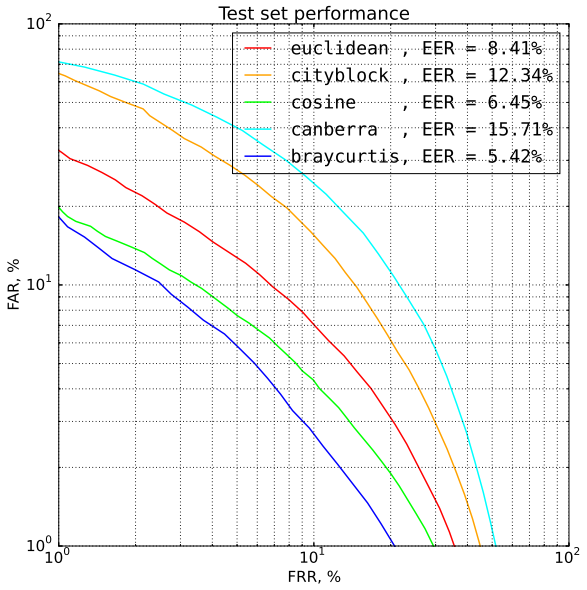
Рис. 14

Ни один вариант со сложными схемами нормализации не давал улучшения качества применительно к этой задаче.

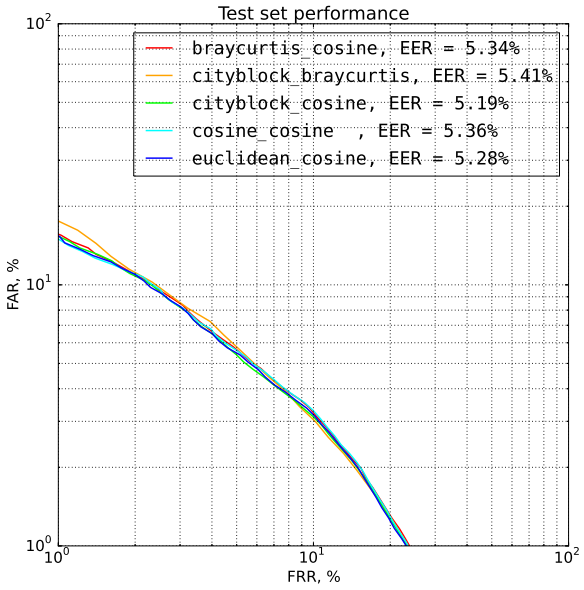
6.1.4 Использование метода главных компонент

Естественно предположить, что информация, закодированная в дескрипторах избыточна. Применим метод главных компонент (PCA) к набору дескрипторов для уменьшения размерности данных. Анализ дисперсии 14b показывает, что уменьшение размерности до 100 совсем незначительно уменьшит полезную информацию, заложенную в дескрипторах.

Результаты верификации по моделям, рассчитанным по первым 100 главным компонентам представлены на рис 15a. Качество верификации значительно вырастает (с 5.94% до 5.42%), если используется расстояние Брэя-Кертиса.



(a) Результаты верификации по моделям после применения PCA.



(b) Использование VLAD после PCA. Для некоторых пар метрик заметен положительный эффект от сокращения размерности.

Рис. 15

Применение VLAD к PCA данным уменьшает ошибку верификации. На Рис 15b приведены результаты работы такого алгоритма (количество центроид $K = 3$). Именно с использованием PCA+VLAD достигается наилучшее качество на тестовой выборке ERR = 5.19%.

6.2 Вторая серия экспериментов

Считается, что глубинные нейронные сети являются очень «сильными» моделями, которые способны извлекать нужные признаки и одновременно настраивать классификатор поверх этих признаков. Таким образом предпочтительно подавать на вход нейросети «сырые» данные. Заметим, что мел-кепстральные коэффициенты получаются путем обширных преобразований мел-спектрограммы. Кроме того, например, дискретное косинусное преобразование мотивировано необходимостью декорреляции признаков – таким образом сокращается число параметров в часто используемой модели смеси гауссиан. Такая предобработка данных может быть из-

лишней, поэтому в данных экспериментах используются окна мел-спектрограммы размера 207x207 в качестве входных данных нейросети второй конфигурацией.

6.2.1 Сравнение функций близости

Также исследуем поведение ERR в зависимости от конкретной функций близости и для этой конфигурации.

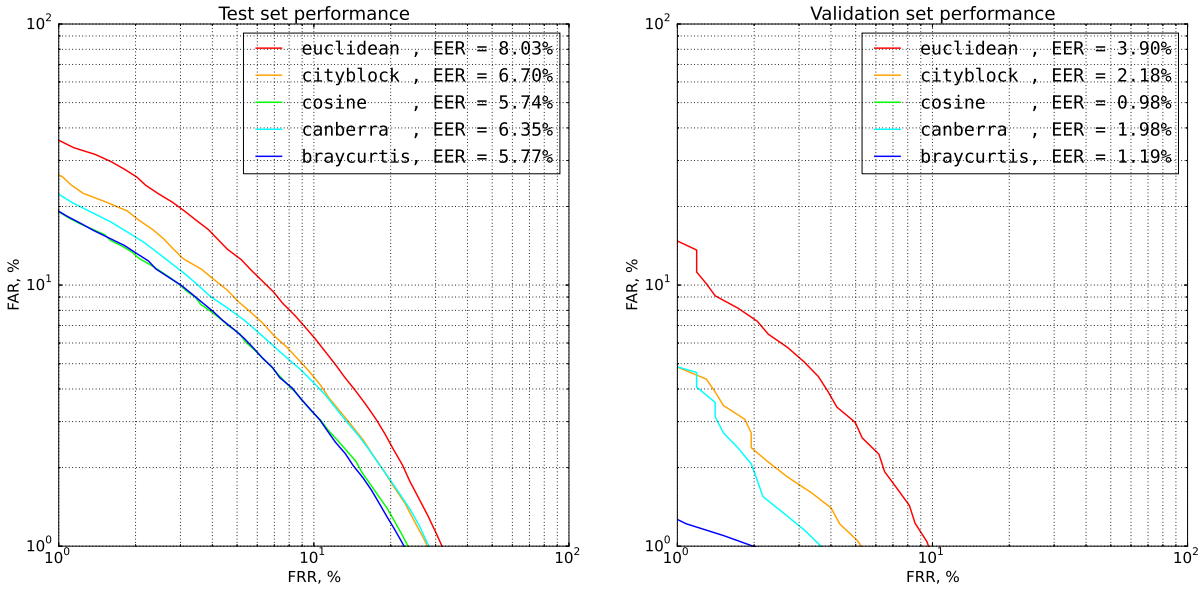


Рис. 16: Сравнение различных метрик на тестовой и валидационной выборках. Обратите внимание на масштаб графиков.

На Рис 16 видна проблема несоответствия ошибки на валидационной выборке ошибке на тестовой выборке. Таким образом обобщающая способность нейросети очень высока уроне для дикторов, встречающихся в обучающей выборке и хуже на тестовой выборке, состоящих из незнакомых дикторов. Заметим, что в 6.1 разница также присутствовала, однако была заметно меньше.

Косинусное расстояние показывает наилучшие результаты для данного протокола.

6.2.2 Улучшение результата

В качестве функции активации последнего скрытого слоя используется функция $ReLU = max(x, 0)$. Таким образом информация об отрицательном полупростран-

стве дескрипторов теряется, хотя и, наверняка, представляет ценность. На рис. 17 представлены результаты для случая, когда дескрипторы извлекаются с последнего скрытого слоя нейросети, но до применения функции активации слоя.

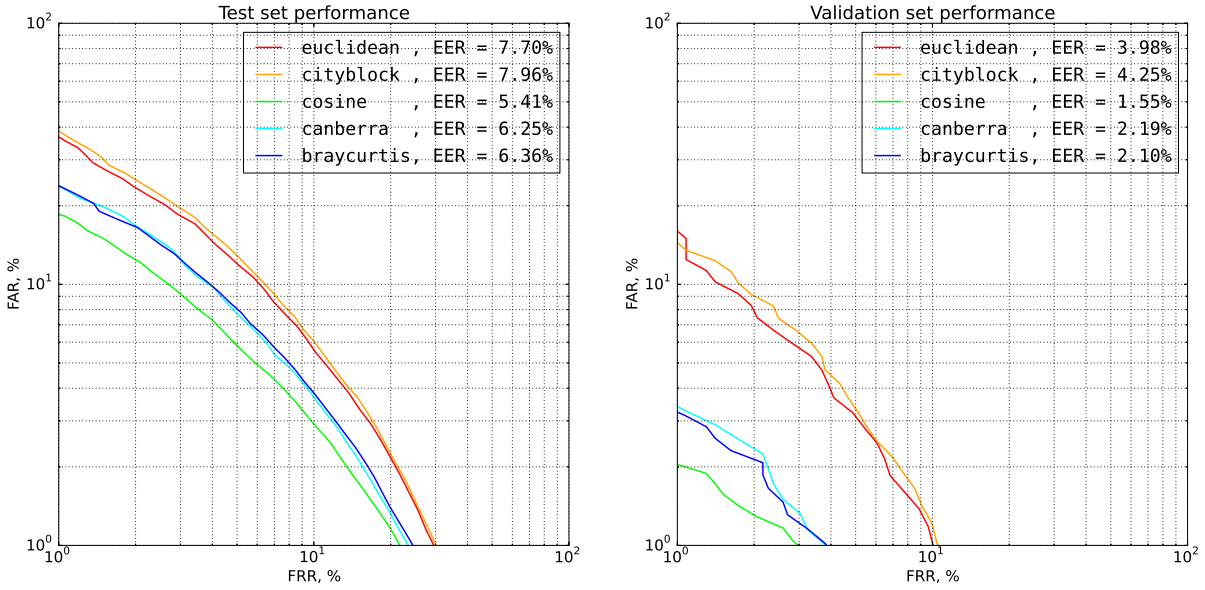


Рис. 17: Результаты при использовании линейной функции активации последнего скрытого слоя.

Падение качества на валидации легко объяснимо, улучшение качества на тесте может говорить либо о переобучении метода на дикторов из обучения, либо о том, что в отрицательном полупространстве все-таки находится важная информация.

6.2.3 Сокращение размерности

Заметим, что размерность дескрипторов (а значит, и модели диктора) равна 1024. Это в 5 раз больше, чем в экспериментах из раздела 6.1. Возможно сокращение размерности модели случайным выбором подмножества компонент дескрипторов. Сравнительные результаты для косинусной метрики представлены в таблице 6.

Использование метода главных компонент позволяет значительно повысить качество распознавания. Вариация главных компонент отображена на Рис. 18a. Результаты верификации с использованием первых 200 главных компонент представлены на Рис 18b.

Таблица 6: Эффект от сэмплирования. Результаты представлены для понимания порядка уменьшения качества от размера.

Размерность	EER Тест, %	EER Валидация, %
256	6.12	1.21
512	5.98	1.13
768	5.92	1.05
1024	5.74	0.98

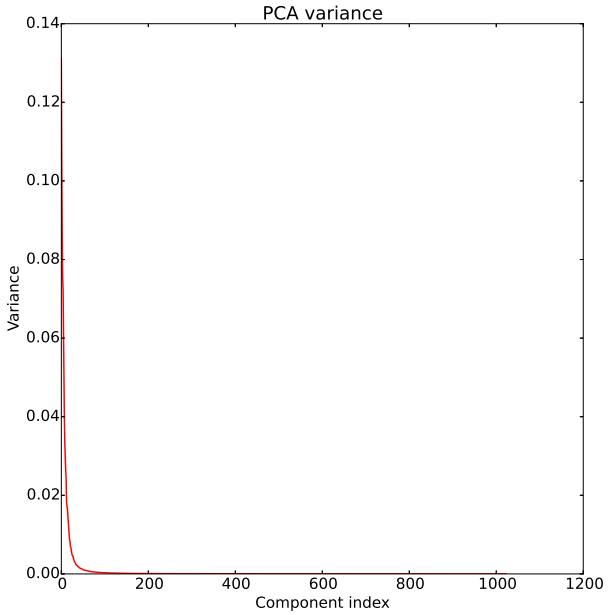
6.3 Сравнение моделей

В таблице 7 представлены результаты некоторых моделей для сравнения. Кроме того предлагается сравнить полученные результаты с методом GMM-UMB (реализация spear [12]).

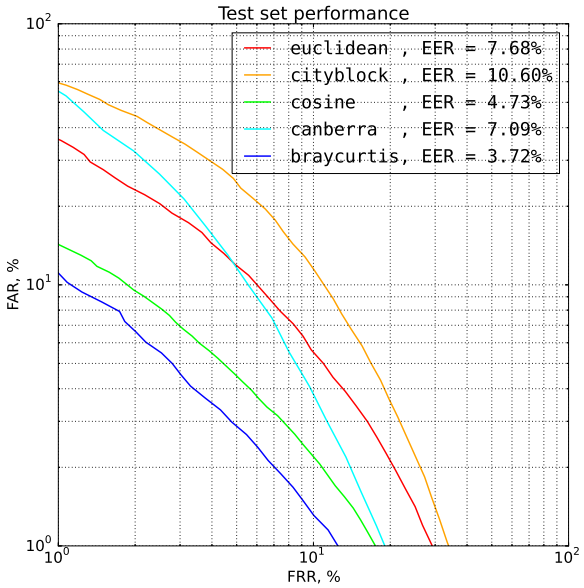
Таблица 7: Сравнение моделей.

Метод	**EER Тест,** %
GMM-UBM-250g	5.52
MFCC+canberra	5.79
MFCC+PCA+braycurtis	5.42
MFCC+VLAD+cityblock_braycurtis	5.30
MFCC+PCA+VLAD+cityblock_cosine	*5.19*
Mel+cosine	5.74
Mel+no_relu+cosine	5.41
Mel+no_relu+PCA+braycurtis	**3.72**

Таким образом лучшей системой является модель, в которой используется мел-спектрограмма в качестве исходных данных для нейросети типа AlexNet, дескрипторы извлекаются до применения ReLU на последнем слое, размерность сокращается с помощью PCA в несколько раз, а полученные модели дикторов сравниваются с помощью метрики Брэя-Кертиса.



(a) Вариацая главных компонент.



(b) Результаты с использованием 200 главных компонент.

Рис. 18

7 Заключение

В ходе выполнения дипломной работы получены следующие результаты:

- Исследована и реализована система верификации диктора на основе искусственных нейронных сетей.
- Проведено численное исследование моделей, а также сравнение с существующей системой.
- Предложена эвристика, позволяющая улучшить качество распознавания с использованием косинусного расстояния.
- Показана применимость метода VLAD, никогда не использовавшегося в этой задаче.
- Получен результат $ERR = 3.72\%$ на базе TIMIT при использовании одной записи для настройки на диктора.

Таблица 8: Параметры HTK.

$SOURCEFORMAT = WAV$
$TARGETKIND =$ MFCC_0_D_A_Z
$TARGETRATE = 100000.0$
$SAVECOMPRESSED = T$
$SAVEWITHCRC = T$
$WINDOWSIZE = 250000.0$
$USEHAMMING = T$
$PREEMCOEF = 0.97$
$NUMCHANS = 26$
$CEPLIFTER = 2$
$NUMCEPS = 12$
$ENORMALIZE = T$

Список литературы

[1] R. Arandjelović and A. Zisserman. All about VLAD. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[2] J Roger Bray and John T Curtis. An ordination of the upland forest communities of southern wisconsin. *Ecological monographs*, 27(4):325–349, 1957.

[3] Namrata Dave. Feature extraction methods lpc, plp and mfcc in speech recognition. *International Journal for Advance Research in Engineering and Technology*, 1, 2013.

[4] John R. Deller, Jr., John G. Proakis, and John H. Hansen. *Discrete Time Processing of Speech Signals*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 1993.

[5] Gunnar Fant. Acoustic theory of speech production : with calculations based on X-ray studies of russian articulations, 1970.

[6] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren. DARPA TIMIT acoustic phonetic continuous speech corpus CDROM, 1993.

[7] J. A. Hartigan and M. A. Wong. A K-means clustering algorithm. *Applied Statistics*, 28:100–108, 1979.

[8] Jialong He, Li Liu, and G. Palm. A discriminative training algorithm for vq-based speaker identification. *Speech and Audio Processing, IEEE Transactions on*, 7(3):353–356, May 1999.

[9] Anil K. Jain, Robert P. W. Duin, and Jianchang Mao. Statistical pattern recognition: A review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(1):4–37, January 2000.

[10] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, 2012.

[11] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[12] E. Khoury, L. El Shafey, and S. Marcel. Spear: An open source toolbox for speaker recognition based on Bob. In *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[13] Tomi Kinnunen, Teemu Kilpelinen, and Pasi FrÄnti. Comparison of clustering algorithms in speaker identification.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[15] G. N. Lance and W. T. Williams. Mixed-data classificatory programs i - agglomerative systems. *Australian Computer Journal*, 1(1):15–20, 1967.

[16] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28:84–95, 1980.

[17] Mitchell McLaren, Yun Lei, Nicolas Scheffer, and Luciana Ferrer. Application of convolutional neural networks to speaker recognition in noisy conditions. In Haizhou Li, Helen M. Meng, Bin Ma, Engsiong Chng, and Lei Xie, editors, *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*, pages 686–690. ISCA, 2014.

[18] Lawrence R Rabiner. *Fundamentals of speech recognition*, volume 14.

[19] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1):19–41, 2000.

[20] Douglas A Reynolds and Richard C Rose. Robust text-independent speaker identification using gaussian mixture speaker models. *Speech and Audio Processing, IEEE Transactions on*, 3(1):72–83, 1995.

[21] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.

[22] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez. Deep neural networks for small footprint text-dependent speaker verification. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 4052–4056. IEEE, 2014.

[23] КВ Воронцов. Математические методы обучения по прецедентам (теория обучения машин).