

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий

институт

Вычислительная техника

кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

_____ А. И. Легалов

подпись инициалы, фамилия

«__» _____ 2016 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника

код и наименование специальности

Модуль голосовой идентификации диктора

тема

Пояснительная записка

Руководитель

подпись, дата

доцент

должность, учёная степень

М.С. Медведев

инициалы, фамилия

Выпускник

подпись, дата

И.В. Дашкевич

инициалы, фамилия

Нормоконтролер

подпись, дата

доцент, к.т.н

должность, учёная степень

В. И. Иванов

инициалы, фамилия

Красноярск 2016

СОДЕРЖАНИЕ

Введение.....	4
Задачи	6
1 Анализ системы	7
1.1 Анализ технического задания	7
1.2 Методы выделения признаков речевого сигнала.....	9
1.2.1 Преобразование Фурье	9
1.2.2 Вейвлет-преобразование.....	12
1.2.3 Преобразование Гильберта-Хуанга.....	19
1.2.4 Вывод.....	20
1.3 Средства разработки	22
2 Проектирование модуля голосовой идентификации диктора.....	25
2.1 Структурная схема	25
2.2 Алгоритм работы модуля	27
2.3 Методов классификации речевого сигнала	29
2.3.1 Динамическая трансформация шкалы времени.....	29
2.3.2 Скрытая Марковская модель.....	29
2.3.3 Вектор квантования.....	30
2.3.4 Нейронная сеть	30
2.3.5 Вывод.....	40
3 Программная реализация.....	41
3.1 Описание работы программы.....	41

3.2 Оценка качества работы модуля идентификации	46
3.3 Вывод.....	47
Заключение	48
Список используемых источников.....	49
Приложение А	51

ВВЕДЕНИЕ

Системы голосовой идентификации диктора (СГИД) быстро развиваются в последнее время. Причиной развития СГИД является их востребованность в таких областях, как биометрический поиск, голосовая верификация водителя и пассажиров, разграничение прав доступа к информации с помощью голосовой биометрии и т. д. [1]. Важным достоинством СГИД по сравнению с другими биометрическими системами идентификации является их дешевизна. Важно также, что современные СГИД по уровню надёжности идентификации не уступают, а бывает и превосходят, к примеру, системы идентификации человека по изображению [2]. Эволюция систем распознавания речи привела к созданию интеллектуальных систем, позволяющих не только распознавать, но и автоматически синтезировать человеческую речь.

Несмотря на уникальность голоса человека, ни одна из СГИД, как и любая другая биометрическая система, не может гарантировать 100% надёжность идентификации. Основными источниками ошибок в СГИД являются: окружение (шум, реверберация и т.д.); особенности речи (длительность, тональность, уровень голосового усилия и т.д.); канал связи (искажения микрофона и канала передачи, погрешности кодирования аудио сигнала и т.д.) [2].

В общем случае идентификация личности по голосу требует решения большого числа разнородных задач, основными из которых являются следующие:

- выделение вокализованных участков аудио сигнала путём отбрасывания пауз и участков, содержащих различного рода помехи;
- разделение речи дикторов (задача диаризации);
- выделение характерных признаков голоса диктора.

Применение технологии идентификации и верификации диктора

позволяет:

- осуществить тексто- и языконезависимую идентификацию диктора;
- выделить из общего объёма данных звуковые файлы, содержащие речь интересующего диктора;
- подтвердить принадлежность звуковых данных тому или иному диктору;
- снизить риск пропуска файла с речью диктора из-за перегруженности операторов;
- обработать большое количество речевой информации и подготовить данные для дальнейшей обработки оператором[14].

ЗАДАЧИ

- Разработать метод выделения признаков речевого сигнала, позволяющий проводить идентификацию дикторов;
- проанализировать существующие подходы и выбрать алгоритм классификации пользователей по их голосовым характеристикам;
- разработать модуль голосовой идентификации диктора на рабочей среде **MATLAB**, в дальнейшем подключаемого к системе распознавания речи, с целью повышения качества распознавания;
- внедрить дополнительный этап классификации диктора — повышение качества распознавания речи за счёт выбора оптимального классификатора, обученного на речевом материале, с схожими с диктором голосовыми характеристиками;
- определить качество распознавания модуля идентификации диктора.

1 Анализ системы

1.1 Анализ технического задания

Распознавание дикторов объединяет идентификацию и верификацию дикторов.

Идентификация диктора — процесс выявления личности по образцу голоса путём сравнения данного образца с образцами, сохранёнными в базе. Результатом процесса идентификации является список кандидатов. Реализующая система может выдавать список фиксированного размера либо принимать решение о включении пользователя в список кандидатов на основании заданного порога. Если предусмотрена возможность того, что в процессе идентификации будет участвовать пользователь, не зарегистрированный в системе, то говорят об идентификации на открытом множестве. Если все пользователи, проходящие процедуру идентификации, зарегистрированы в системе, то говорят об идентификации на замкнутом множестве.

Верификация диктора — процесс, при котором с помощью сравнения представленного образца с хранимым в базе шаблоном проверяется запрошенная идентичность. Результатом верификации является положительное либо отрицательное решение. Иногда используется термин «обнаружение по голосу» (speaker detection [4]). В задаче обнаружения используются несколько иные термины и приоритеты, но, по сути, верификация и обнаружение являются одной той же задачей [3].

В нашей работе большее внимание уделено задаче выделения характерных признаков голоса (ХПГ). Вообще говоря, для выделения ХПГ можно использовать спектрально-формантный анализ, статистические характеристики основного тона голоса, параметрическое представление

аудиосигнала. ХПГ можно искать в амплитуде и диапазоне частот звукового сигнала, а также в его спектре. Обширный список публикаций, посвящённых перечисленным методам, представлен в работе [5].

Существует два способа распознавания аудиосигнала, дикторозависимый и дикторонезависимый.

Дикторозависимая система предназначена для использования одним пользователем, в то время как дикторонезависимая система предназначена для работы с любым диктором. Дикторонезависимость – труднодостижимая цель, так как при обучении системы, она настраивается на параметры того диктора, на примере которого обучается. Частота ошибок распознавания таких систем обычно в 3-5 раз больше, чем частота ошибок дикторозависимых систем[9].

1.2 Методы выделения признаков речевого сигнала

Существует несколько способов преобразования сигнала, о самых распространённых из них будет описано ниже.

1.2.1 Преобразование Фурье

Преобразование Фурье — операция, превращающая одну функцию вещественной переменной в другую функцию вещественной переменной. Эта новая функция описывает коэффициенты («амплитуды») при разложении исходной функции на элементарные составляющие — гармонические колебания с разными частотами (подобно тому, как музыкальный аккорд может быть выражен в виде амплитуд нот, которые его составляют).

Преобразование Фурье функции f вещественной переменной является интегральным и задаётся следующей формулой:

$$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x)e^{-ix\omega} dx. \quad (1.1)$$

Хотя формула, задающая преобразование Фурье, имеет понятный смысл только для функций класса $L_1(\mathbb{R})$, преобразование Фурье может быть использовано и для более широкого класса функций и даже обобщённых функций. Это возможно благодаря ряду свойств преобразования Фурье:

Преобразование Фурье является линейным оператором:

$$(\alpha f + \beta g)^\wedge = \alpha \hat{f} + \beta \hat{g}.$$

Справедливо равенство Парсеваля: если $f \in L_1(\mathbb{R}) \cap L_2(\mathbb{R})$, то преобразование Фурье сохраняет L_2 -норму:

$$\int_{-\infty}^{\infty} |f(x)|^2 dx = \int_{-\infty}^{\infty} |\hat{f}(w)|^2 dw. \quad (1.2)$$

Это свойство позволяет по непрерывности распространить определение преобразования Фурье на всё пространство $L_2(\mathbb{R})$. Равенство Парсеваля будет при этом справедливо для всех $f \in L_2(\mathbb{R})$.

Формула обращения:

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{f}(w) e^{ixw} dw \quad (1.3)$$

справедлива, если интеграл в правой части имеет смысл. В частности, это верно, если функция f является достаточно гладкой. Если $f \in L_2(\mathbb{R})$, то формула также верна, поскольку равенство Парсеваля позволяет придать интегралу в правой части смысл с помощью предельного перехода.

Эта формула объясняет физический смысл преобразования Фурье: правая часть — (бесконечная) сумма гармонических колебаний $e^{i\omega x}$ с

частотами ω , амплитудами $\frac{1}{\sqrt{2\pi}} |\hat{f}(\omega)|$ и фазовыми сдвигами $\arg \hat{f}(\omega)$ соответственно.

Теорема о свёртке: если $f, g \in L_1(\mathbb{R})$, тогда

$$\begin{aligned} (\widehat{f * g}) &= \sqrt{2\pi} \hat{f} \hat{g}, \text{ где} \\ (f * g)(t) &= \int_{-\infty}^{\infty} f(t-s)g(s) ds. \end{aligned} \quad (1.4)$$

Эта формула может быть распространена и на случай обобщённых функций.

Преобразование Фурье и дифференцирование. Если $f, f' \in L_1(\mathbb{R})$, то

$$\widehat{(f')} = i\omega \widehat{f}. \quad (1.5)$$

Из этой формулы легко выводится формула для n -й производной:

$$\widehat{(f^{(n)})} = (i\omega)^n \widehat{f}. \quad (1.6)$$

Формулы верны и в случае обобщённых функций.

Преобразование Фурье и сдвиг.

$$\widehat{f(x - x_0)} = e^{-i\omega x_0} \widehat{f}(\omega). \quad (1.7)$$

Эта и предыдущая формула являются частными случаями теоремы о свёртке, так как сдвиг по аргументу — это свёртка со сдвинутой дельта-функцией $\delta(x - x_0)$, а дифференцирование — свёртка с производной дельта-функции.

Преобразование Фурье и растяжение.

$$\widehat{f(ax)} = |a|^{-1} \widehat{f}(\omega/a). \quad (1.8)$$

Преобразование Фурье обобщённых функций. Преобразование Фурье можно определить для широкого класса обобщённых функций. Определим вначале пространство гладких быстро убывающих функций (пространство Шварца):

$$S(\mathbb{R}) := \left\{ \varphi \in C^\infty(\mathbb{R}) : \forall n, m \in \mathbb{N} \ x^n \varphi^{(m)}(x) \xrightarrow{x \rightarrow \infty} 0 \right\}. \quad (1.9)$$

Ключевым свойством этого пространства является то, что это инвариантное подпространство по отношению к преобразованию Фурье[7].

1.2.2 Вейвлет-преобразование

Широко используемое преобразование Фурье для анализа сигналов, как непрерывное, так и дискретное, оказывается недостаточно эффективным при обработке сложных сигналов. Например, Фурье спектры для сигналов из двух синусоид с разными частотами, первый из которых представляет собой сумму синусоид, а второй – последовательно следующие друг за другом синусоиды, одинаковы и будут выглядеть как два пика на двух фиксированных частотах (рисунок 1). Из этого следует, преобразование Фурье в своём обычном виде не приспособлено для анализа нестационарных сигналов, в том числе расположенных на некотором временном интервале, так как теряется информация о временных характеристиках сигнала. Речевой сигнал является примером нестационарного процесса, в котором информативным является сам факт изменения его частотно-временных характеристик.

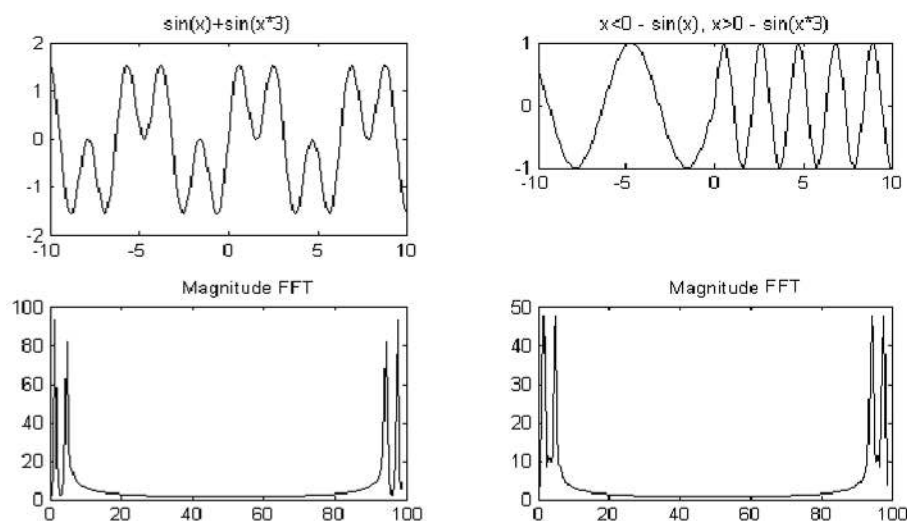


Рисунок 1– Пример неинформативности преобразования Фурье

Для анализа таких процессов требуются базисные функции, способные выявлять в исследуемом сигнале как частотные, так и его временные характеристики, т. е. функции со свойствами частотно-временной локализации. Такие возможности предоставляют вейвлеты, являющиеся обобщением спектрального анализа.

Вейвлеты – функции двух аргументов – масштаба и сдвига. В отличие от стандартного преобразования Фурье, они позволяют обрабатывать сигнал одновременно в физическом пространстве – время, координата, и частотном пространстве

$$[W_{\psi} f](x, a) = |a|^{-1/2} \int_{-\infty}^{+\infty} f(t) \psi\left(\frac{t-x}{a}\right) dt \quad (1.10)$$

здесь $\psi\left(\frac{t-x}{a}\right)$ – вейвлет, a – масштабный коэффициент, x – параметры сдвига. Таким образом, вейвлет-преобразование обеспечивает двумерное представление исследуемого сигнала в частотной области в плоскости частота-положение. Аналогом частоты при этом является масштаб аргумента базисной функции (чаще всего – времени), а положение характеризуется её сдвигом. Это позволяет найти особенности сигналов, одновременно локализуя их на временной шкале. Другими словами, вейвлет-анализ можно охарактеризовать как спектральный анализ локальных возмущений.

В теории вейвлет-анализа существует множество направлений. Например, используя многомасштабный (кратномасштабный) вейвлет-анализ сигнал можно представить как последовательность образов с разной степенью детализации, что позволяет найти локальные особенности сигнала и классифицировать их по интенсивности.

Анализ основывается на разложении сигнала по функциям, образующим ортонормированный базис. Каждую функцию можно разложить на некотором заданном уровне разрешения (масштабе) j_n в ряд вида:

$$f(x) = \sum_{k=0}^{2M-1} s_{j_n, k} \varphi_{j_n, k} + \sum_{j \geq j_n}^{j_{\max}} \sum_{k=0}^{2M-1} d_{j, k} \psi_{j, k} \quad (1.11)$$

Здесь:

$\varphi_{j_n, k}$ и $\psi_{j, k}$ – масштабированные и смещённые версии скейлинг-функции (масштабной функции) φ и "материнского вейвлета" ψ ;

$s_{j, k}$ – коэффициенты аппроксимации;

$d_{j, k}$ – детализирующие коэффициенты.

На рисунке 2-5 представлены аналитические графики функций φ и ψ указанных вейвлетов.

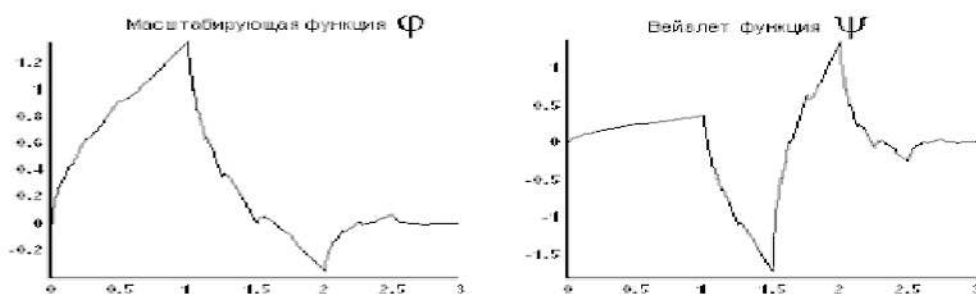


Рисунок – 2 Аналитические графики функций φ и ψ , вейвлет Симлета

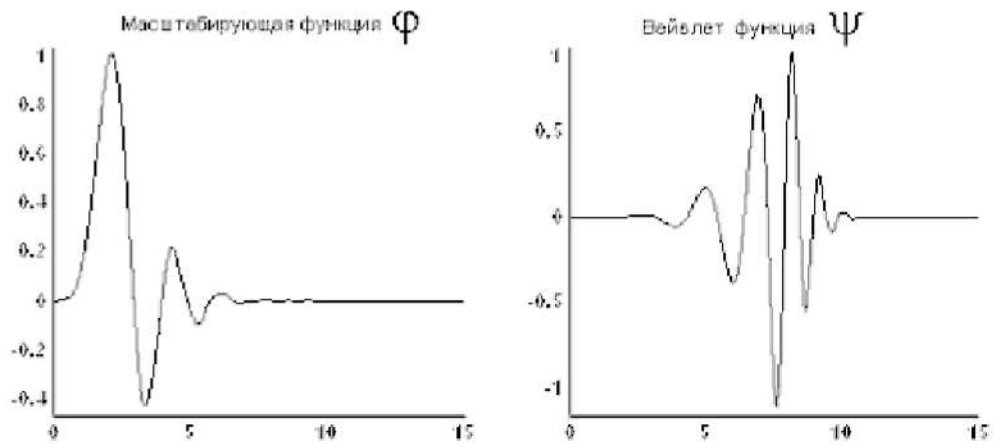


Рисунок – 3 Аналитические графики функций φ и ψ , вейвлет Добеши 8

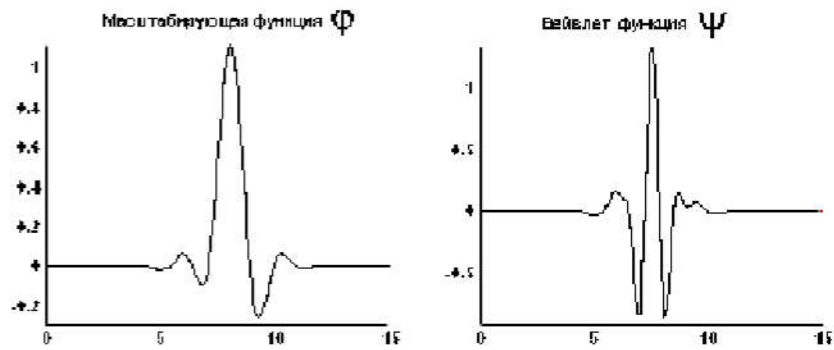


Рисунок – 4 Аналитические графики функций φ и ψ , вейвлет Симлета 8

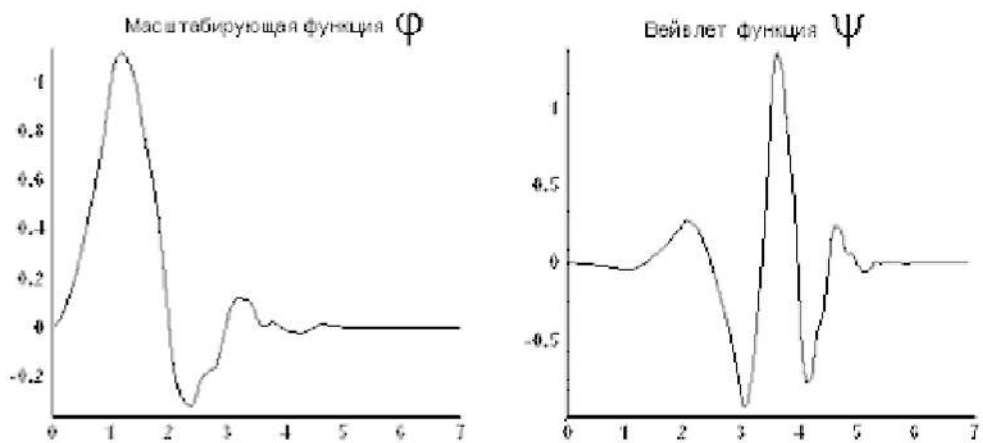


Рисунок 5- Аналитические графики функций φ и ψ , вейвлет Добеши 4

Масштабирование и смещение функций φ и ψ находится по законам

$$\begin{aligned}\varphi_{j,k} &= 2^{j/2} \varphi(2^j x - k), \\ \psi_{j,k} &= 2^{j/2} \psi(2^j x - k).\end{aligned}\tag{1.12}$$

В свою очередь сами функции φ и ψ определяются как:

$$\begin{aligned}\varphi(x) &= \sqrt{2} \sum_{k=0}^{2M-1} h_k \varphi(2x - k) \\ \psi(x) &= \sqrt{2} \sum_{k=0}^{2M-1} g_k \varphi(2x - k)\end{aligned}\tag{1.13}$$

где $g_k = (-1)^k h_{2M-k-1}$

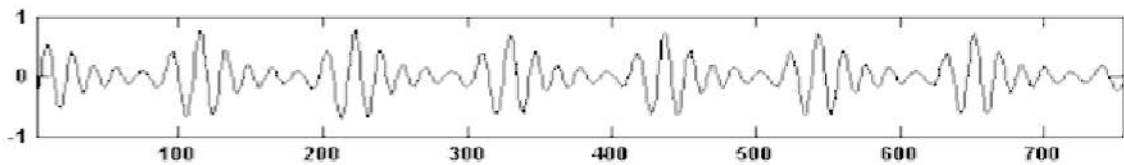


Рисунок – 6 Скелет максимумов фонемы "а", линия коэффициентов вейвлет-преобразования фонемы "а",

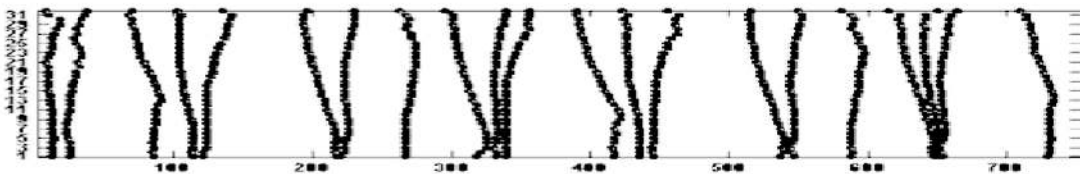


Рисунок – 7 Скелет максимумов фонемы "а":
скелет максимумов фонемы "а"

Выполняя свойства ортогональности масштабных функций и задавая значения M , можно вычислять конкретные значения коэффициентов h_k , определяющие ортогональные вейвлеты. Например, задав $M=2$ получаем ряд коэффициентов h_k , определяющий вейвлет Добеши 4.

Таким образом, ортогональный вейвлет-анализ сводится к нахождению коэффициентов аппроксимации $S_{j,k}$ и детализирующих коэффициентов $d_{j,k}$ в разложении сигнала $f(x)$ по формуле (1.11). Выше приведён пример скелета максимумов для фонемы "а" (рисунок 6,7).

Множество точек на (x, a) , в которых находятся локальные "пики" вейвлет-преобразования, образуют скелет максимумов. Этим точкам часто очень много в области малых масштабов. Их появлением вейвлет-преобразование реагирует на любые негладкости сигнала. При росте масштаба мелкие негладкости исчезают, а вместе с ними и точки максимумов. Оставшиеся сливаются в довольно гладкие кривые, которые при дальнейшем росте масштаба тоже сливаются друг с другом. При этом они либо "аннигилируют", либо продолжают "расти" в область ещё более крупных масштабов, и т. д. В определённом смысле, вся существенная информация о сигнале находится в значениях скелета максимумов вейвлет-преобразования (рисунок 6). Использование этого механизма позволяет более эффективно выделять фонемы из речевого потока.

Достоинства:

- Вейвлетные преобразования обладают всеми достоинствами преобразований Фурье.
- Вейвлетные базисы могут быть хорошо локализованными как по частоте, так и по времени. При выделении в сигналах хорошо локализованных разномасштабных процессов можно рассматривать только те масштабные уровни разложения, которые представляют интерес.

- Базисные вейвлеты могут реализоваться функциями различной гладкости.

Недостатки:

- Можно выделить один недостаток, это относительная сложность преобразования.

1.2.3 Преобразование Гильберта-Хуанга

Под преобразованием Гильберта-Хуанга (**ННТ**) понимается метод эмпирической модовой декомпозиции (EMD) нелинейных и нестационарных процессов и Гильбертов спектральный анализ (HSA). ННТ представляет собой частотно-временной анализ данных (сигналов) и не требует *априорного* функционального базиса преобразования. Функции базиса получаются адаптивно непосредственно из данных процедурами отсеивания функций «эмпирических мод». Мгновенные частоты вычисляются от производных фазовых функций Гильбертовым преобразованием функций базиса. Заключительный результат представляется в частотно-временном пространстве Преобразование Гильберта $\text{TH}[x(t)]$ действительной функции $x(t)$, $-\infty < t < \infty$, - есть действительная функция, определённая как

$$\begin{aligned}\tilde{x}(t) &= \text{TH}[x(t)] = x(t) * (1/\pi t), \\ \tilde{x}(t) &= \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x(\tau)}{t-\tau} d\tau.\end{aligned}\tag{1.14}$$

Функция $1/(t-\tau)$ называется *ядром преобразования Гильберта*. Преобразование имеет особую точку при $t-\tau \Rightarrow 0$, в которой при вычислении используется главное значение интеграла по Коши. Функции $x(t)$ и $\tilde{x}(t)$ обычно называют сопряжёнными по Гильберту.

Физически, преобразование Гильберта может быть интерпретировано как естественный $\tau/2$ фазовращатель, который при прохождении через систему сигнала $x(t)$ изменяет фазу всех частотных составляющих сигнала на $\tau/2$, и тем самым делает сигнал $\tilde{x}(t)$ ортогональным сигналу $x(t)$. Это позволяет сформировать из сигналов $x(t)$ и $\tilde{x}(t)$ комплексный аналитический сигнал $z(t)$,

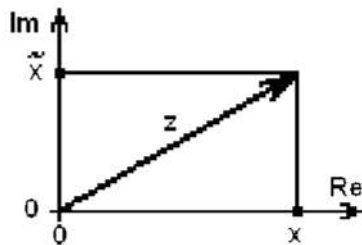


Рисунок – 8 $z(t) = x(t) + j \tilde{x}(t)$

где $z(t)$ представлен вектором на комплексной плоскости с проекциями на действительной и мнимой оси соответственно $x(t)$ и $\tilde{x}(t)$ (1.14). Преимущество этого представления состоит в том, что возникает возможность однозначно определять текущие временные параметры сигнала $z(t)$, а именно – мгновенные значения его амплитуды и фазы.

Допустим, что имеем зарегистрированный радиосигнал $x(t)$ с несущей частотой ω_0 , который содержит определённую информацию, заключённую в огибающей сигнала $u(t)$ и его фазе $\tilde{x}(t)$:

$$x(t) = u(t) \cos(\omega_0 t + \tilde{x}(t)). \quad (1.15)$$

В другой форме:

$$\begin{aligned} x(t) &= a(t) \cdot \cos(\omega_0 t) + b(t) \cdot \sin(\omega_0 t), \\ a(t) &= u(t) \cos \tilde{x}t, \quad b(t) = u(t) \sin \tilde{x}t, \quad u(t) = \sqrt{a^2(t) + b^2(t)}, \quad \text{tg } \tilde{x}(t) = b(t)/a(t). \end{aligned} \quad (1.16)$$

1.2.4 Вывод

Проведя небольшой анализ и оценив все достоинства и недостатки представленных методов преобразования, метод вейвлет-преобразования сигналов является наиболее предпочтительным для применения при создании требуемого модуля. Данный метод не содержит сложных

последовательностей действий, признаки, получаемые в результате, характеризуют сигнал и во временной плоскости и в частотной, что даёт хорошие результаты для классификации сигналов. По сравнению с преобразованием Фурье, Вейвлет-преобразование имеет лучшее частотно-временное разрешение, а при сравнении с преобразованием Гильберта-Хуанга, у Вейвлет-преобразования быстродействие выше.

1.3 Средства разработки

Среда разработки Matlab 12

В качестве платформы для исследований и разработки модуля голосовой идентификации диктора был выбран пакет MATLAB. Данный выбор объясняется следующим:

MATLAB - пакет прикладных программ для решения задач технических вычислений, а также используемый в этом пакете язык программирования. MATLAB используют более 1 000 000 научных и инженерных работников, он работает на большинстве современных операционных систем, включая GNU/Linux, Mac OS, Solaris и Microsoft Windows.

Язык MATLAB является высокоуровневым интерпретируемым языком программирования, включающим основанные на матрицах структуры данных, широкий спектр функций, интегрированную среду разработки, объектно-ориентированные возможности и интерфейсы к программам, написанным на других языках программирования.

MATLAB предоставляет пользователю большое количество (несколько сотен) функций для анализа данных, покрывающие практически все области математики.

MATLAB - это удобные средства для разработки алгоритмов, включая высокоуровневые, с использованием концепций объектно-ориентированного программирования. В нём имеются все необходимые средства интегрированной среды разработки, включая отладчик и профайлер. Функции для работы с целыми типами данных облегчают создание алгоритмов для микроконтроллеров и других приложений, где это необходимо.

Встроенная среда разработки позволяет создавать графические интерфейсы пользователя с различными элементами управления, такими как кнопки, поля ввода и другими. С помощью компонента MATLAB Compiler

эти графические интерфейсы могут быть преобразованы в самостоятельные приложения, для запуска которых на других компьютерах необходима установленная библиотека MATLAB Component Runtime.

В пакет MATLAB входят различные интерфейсы для получения доступа к внешним подпрограммам, написанным на других языках программирования, данным, клиентам и серверам, общающимся через технологии Component Object Model или Dynamic Data Exchange, а также периферийным устройствам, которые взаимодействуют напрямую с MATLAB. Многие из этих возможностей известны под названием MATLAB API.

Система Matlab предоставляет мощный язык программирования, ориентированный на математические преобразования и превосходящий по возможностям и скорости вычислений традиционные языки программирования [8].

Для решения проблемы классификации, родственные функции и процедуры системы собираются в специальные папки. Это приводит к концепции пакетов прикладных программ, которые представляют собой коллекции M-файлов для решения определённой задачи или проблемы. Именно пакеты прикладных программ – Matlab Application Toolboxes, входящие в состав семейства продуктов Matlab, позволяют находиться этой системе на уровне самых современных приложений.

Для моделирования структуры нейронной сети был выбран пакет Neural Networks. В пакет включены более полутора десятков известных типов искусственных нейронных сетей и обучающих правил, позволяющих пользователю выбрать наиболее подходящую для конкретного приложения или исследовательской задачи парадигму. Для каждого типа архитектуры и обучающего алгоритма имеются функции инициализации, обучения, адаптации, создания и моделирования, демонстрации и примеры применения. Один из пакетов системы - Wavelet Toolbox, предоставляет разнообразные

возможности обработки сигналов с помощью вейвлетов. С его помощью реализуются принципиально новые виды декомпозиции и реконструкции сигналов и изображений с повышенной эффективностью и новыми качественными возможностями – например, в идентификации тонких локальных особенностей функций и сигналов.

В стандартный пакет аудио поддержки системы **Matlab** 12 включены функции, позволяющие произвести запись звукового сигнала с возможностью настройки значений частоты дискретизации и разрядности. Встроенная в **Matlab** среда разработки пользовательского интерфейса **Matlab GUIDE** позволяет реализовывать элементы визуально-ориентированного программирования (кнопки, меню, инструментальные панели и т.д.). В системе **Matlab** присутствует уникальная возможность сохранения значений всех обрабатываемых переменных из оперативной памяти в постоянную. Переменные сохраняются в специальные *mat*-файлы и могут быть в дальнейшем загружены. Т.к. одновременно-файл могут быть сохранены переменные разных типов и структура самого хранения является упорядоченной, то данную возможность системы **Matlab** удобнее применять для создания базы данных эталонов.

2 Проектирование модуля голосовой идентификации диктора

Использование модуля голосовой идентификации диктора в системе распознавания речи предполагает появление коэффициента качества распознавания, а так же повышение дикторонезависимости системы.

2.1 Структурная схема

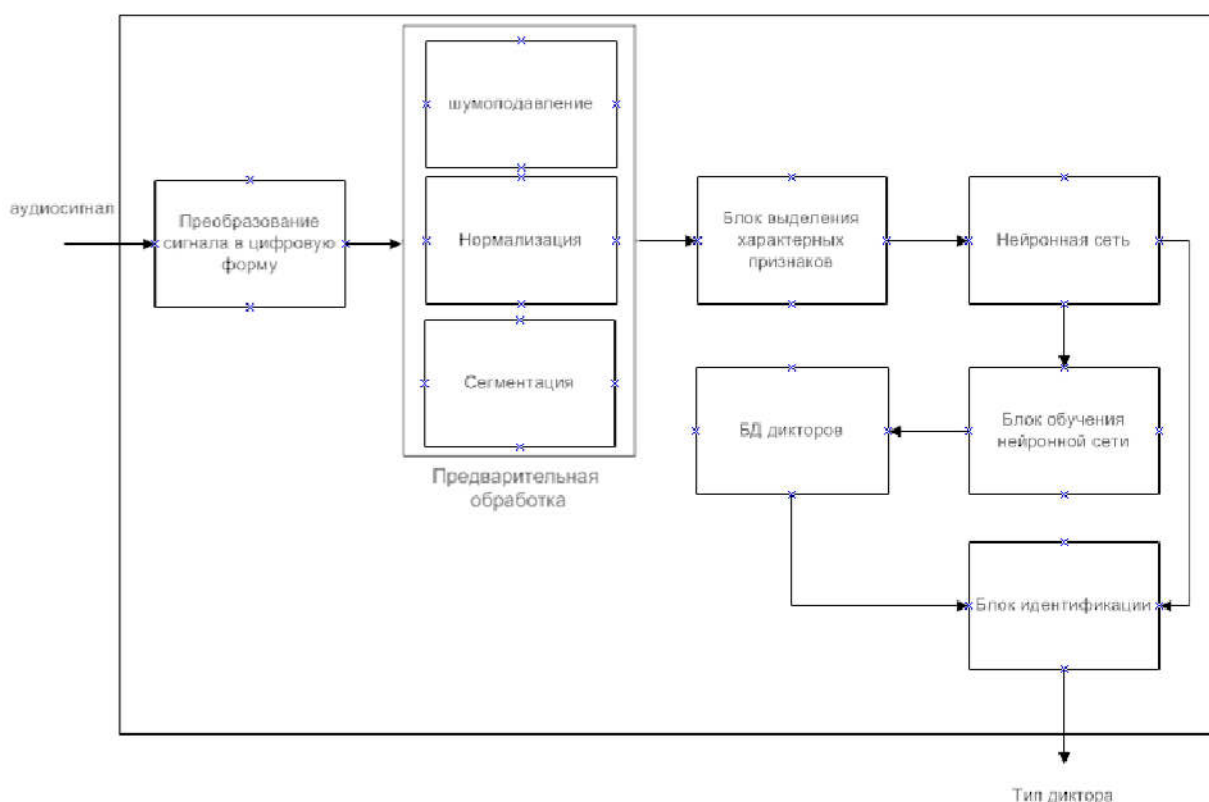


Рисунок 9 - Структурная схема модуля

На вход модуля голосовой идентификации диктора поступает аудиосигнал запись которого происходит через микрофон, подключённый к входу звуковой карты компьютера, или используется уже ранее записанный сигнал. Акустический компонент преобразует сигнал в цифровую форму с заданными параметрами частоты дискретизации. Затем осуществляется

очистка сигнала от шума, удаление не несущих информации участков, проводится нормализация сигнала и его разбиение на фиксированные интервалы во временной области, на которых будут определяться характеристики. На блоке выделение характерных признаков происходит выделение характерных признаков сигнала с помощью Вейвлет-преобразования. В блок нейронная сеть попадает результат с предыдущего блока, дальнейшие действия зависят от выбора режима работы. При выборе «обучение» данные передаются на блок обучения нейронной сети, где происходит её обучение и передача в базу данных. При выборе режима «распознавание» данные переходят на блок идентификации, так же в этот блок приходят данные с блока база данных с уже сохранёнными данными. Далее происходит идентификация и классификация (определение принадлежности к определённому классу), затем следует отправление результата типа диктора.

2.2 Алгоритм работы модуля



Рисунок 10 – Алгоритм работы модуля

Для начала работы подаётся аудиосигнал, затем происходит выбор режима работы модуля. При выборе режима «Обучение», первым делом, происходит Вейвлет-преобразование сигнала, после преобразованный сигнал подаётся на нейронную сеть, где происходит обучение нейронной сети на характерные признаки голоса диктора и сохранение её. При выборе режима «Распознавание» происходит загрузка обученной нейронной сети и выполняется Вейвлет-преобразование сигнала. Преобразованный сигнал подаётся на нейронную сеть для сравнение с характерными признаками голоса диктора. Далее происходит выбор класса диктора, по принципу максимально схожих характерных признаков и выводится результат.

2.3 Методов классификации речевого сигнала

В данном подразделе рассматриваются основные существующие решения задачи идентификации диктора по голосу. Несмотря на то, что методы во многом отличаются, в целом можно выделить следующие основные этапы, присущие каждому из рассматриваемых методов:

1. Извлечение признаков из входного речевого сигнала.
2. Построение модели (шаблона) диктора на основе полученных на предыдущем шаге векторов признаков.

Процесс определения диктора, зарегистрированного в системе, по входному речевому сигналу во всех рассматриваемых методах состоит в поиске наиболее подходящей сохранённой модели на основе каких-либо критериев[10].

2.3.1 Динамическая трансформация шкалы времени

Dynamic Time Warping (DTW) – метод динамического программирования, позволяющий найти близость между двумя последовательностями измерений за некоторый промежуток времени. В общем случае эти последовательности могут быть разной длины, и измерения могут производиться с разной скоростью [11].

2.3.2 Скрытая Марковская модель

Hidden Markov Model (HMM) – статистическая модель, которая служит для решения задачи классификации скрытых параметров на основе наблюдаемых. HMM представляет собой конечный автомат, в котором переходы между состояниями осуществляются с некоторой вероятностью, и задано начальное состояние, с которого начинается процесс.

Через дискретные моменты времени может осуществляться переход в новые состояния. При этом каждому скрытому состоянию с заданной вероятностью соответствует наблюдаемое состояние. Кроме того, текущее состояние автомата зависит только от конечного числа предыдущих, а закон смены состояний не меняется во времени [12].

2.3.3 Вектор квантования

Задача векторного квантования с кодовыми векторами $W = \{w_1, w_2, \dots, w_n\}$ для последовательности входных векторов $C = \{c_1, c_2, \dots, c_m\}$ ставится как задача минимизации искажения при замещении каждого вектора из соответствующим кодовым вектором.

Моделью диктора в данном методе является множество кодовых векторов, получаемое из входной последовательности векторов признаков речевого сигнала. Для построения этого множества исходная последовательность векторов признаков разбивается на L кластеров, и в качестве кодовых векторов берутся их центры.

2.3.4 Нейронная сеть

В любой системе распознавания речи всегда присутствует этап сравнения входного сигнала с имеющимися эталонами. Вне зависимости от наличия или отсутствия предварительной обработки сигнала (выделение основных признаков, преобразование в другую форму в новом параметрическом пространстве и т. д.) сигнал представляет собой вектор в установленном параметрическом пространстве, который в дальнейшем будет сравниваться с векторами, хранящимися в памяти для определения его принадлежности к определённому классу. Такая классификация образов является одной из основных задач решаемых с помощью нейронной сети,

среди которых наиболее распространены:

- Распознавание зрительных, слуховых образов; огромная область применения: от распознавания текста и целей на экране радара до систем голосового управления;
- ассоциативный поиск информации и создание ассоциативных моделей;
- формирование моделей различных нелинейных и трудно описываемых математически систем, прогнозирование развития этих систем во времени; применение на производстве; прогнозирование природных процессов, изменений курсов и т.д;
- системы управления и регулирования с предсказанием; управление роботами, другими сложными устройствами - разнообразные конечные автоматы: системы массового обслуживания и коммутации, телекоммуникационные системы;
- принятие решений и диагностика, исключаящие логический вывод; особенно в областях, где отсутствуют чёткие математические модели: в медицине, криминалистике, финансовой сфере;

Выделим характерные свойства искусственных нейронных сетей:

- Обучаемость. Одним из этапов функционирования нейронной сети является обучение, в процессе которого на ее вход поочерёдно поступают данные из обучающего набора с целью корректировки весовых коэффициентов синаптических связей для получения наиболее адекватного сигнала на выходе нейронной сети.
- Способность к обобщению. Отклик сети после обучения может быть до некоторой степени нечувствителен к небольшим изменениям входных сигналов (шуму или вариациям входных образов).
- Способность к абстрагированию. Если при обучении предъявить сети несколько искажённых вариантов входного образа, то сеть может создать на выходе идеальный образ, который не входил в обучение.

- Параллельность обработки и реализуемости нейронных сетей.
- Универсальность. Хотя почти для всех перечисленных задач существуют эффективные математические методы решения и, несмотря на то, что сети проигрывают специализированным методам; благодаря универсальности и перспективности они являются важным направлением исследования, требующим тщательного изучения[13].

Модель искусственного нейрона. Искусственные нейронные сети индуцированы биологией, так как они состоят из элементов, функциональные возможности которых аналогичны большинству элементарных функций биологического нейрона. На рисунке 11 показана обобщённая модель нейрона, используемая в качестве основного строительного блока в нейронных сетях.

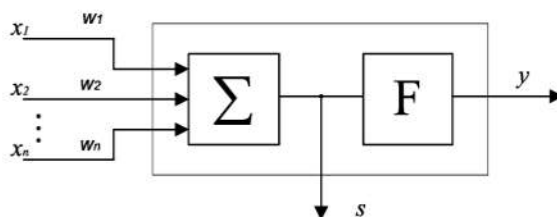


Рисунок 11 – Искусственный нейрон с активационной функцией

На входы нейрона подаётся множество сигналов, каждый из которых умножается на вес, и затем произведения складываются. Результат суммирования s (2.1) служит аргументом функции активации. Значение функции активации соответствует отклику нейрона y (2.2) на произвольную комбинацию входных воздействий. Другими словами, посредством активации нейрона осуществляется трансформация множества входных воздействий в выходной сигнал с желаемыми характеристиками.

$$s = \sum_{i=1}^p w_i x_i + w_0, \quad (2.1)$$

$$y = f(s), \quad (2.2)$$

где w_i - вес синапса, ($i=0,1,2...p$);

w_0 - значение смещения;

s - результат суммирования;

x_i - компонента входного вектора (входной сигнал), ($i=1,2,...p$);

y - выходной сигнал нейрона;

p - число входов нейрона;

f - нелинейное преобразование (функция активации).

В общем случае входной сигнал, весовые коэффициенты и значения смещения могут принимать действительные значения. Выход y определяется видом функции активации и может быть как действительным, так и целым. Во многих практических задачах входы, веса и смещения могут принимать лишь некоторые фиксированные значения. Синаптические связи с положительными весами называют возбуждающими, с отрицательными весами - тормозящими. Таким образом, нейрон полностью описывается своими весами w_i и передаточной функцией $f(x)$. Вместе с правилами корректировки весовых коэффициентов на входе нейрона, правилами обучения, отличительной особенностью многих нейронных структур является выбор функции активации f . Активационная функция может быть обычной линейной функцией (2.3).

$$y=k(s), \quad (2.3)$$

где k – постоянная пороговой функции;

$$y = \begin{cases} 0, & s \leq T \\ 1, & s > T \end{cases}, \quad (2.4)$$

где T – некоторая постоянная пороговая величина.

Если функция активации сужает диапазон изменения величины s (2.1) так, что при любых значениях s значения y (2.2) принадлежат некоторому конечному интервалу, то f называется “сжимающей” функцией. В качестве “сжимающей” функции часто используется логистическая или “сигмоидальная” (S-образная) функция, показанная на рисунке 12.

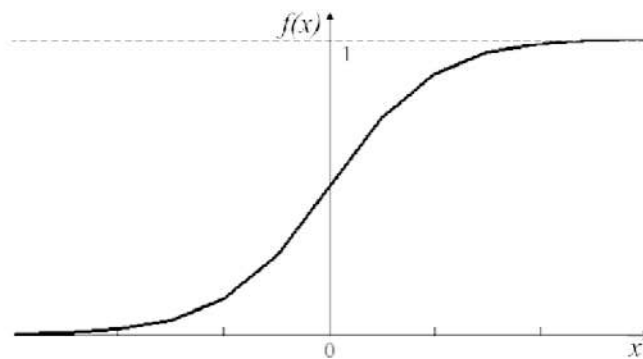


Рисунок 12 - Сигмоидальная функция

Эта функция задаётся формулой (2.5).

$$f(x) = \frac{1}{1 + \exp(-x)}. \quad (2.5)$$

Другой широко используемой активационной функцией является гиперболический тангенс (2.6).

$$f(x) = th(x). \tag{2.6}$$

График функции гиперболического тангенса представлен на рисунке 13.

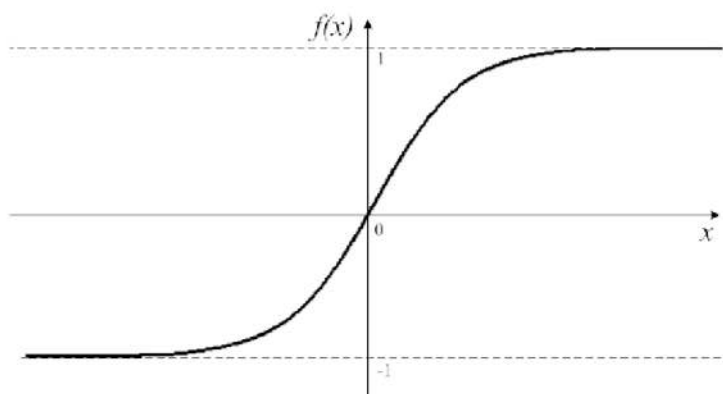


Рисунок 13 – функция гиперболического тангенса

Хотя один нейрон и способен выполнять простейшие процедуры распознавания, сила нейронных вычислений проистекает от соединений нейронов в сетях.

Нейронная сеть представляет собой структуру нейронов, соединённых между собой и характеризуется их внутренними свойствами, индивидуальной топологией (архитектурой), а также правилами обучения для получения желаемого выходного сигнала.

Сеть обучается, чтобы для некоторого множества входов давать желаемое множество выходов. Каждое такое входное (или выходное) множество рассматривается как вектор. Обучение осуществляется путём последовательного предъявления входных векторов с одновременной подстройкой весов в соответствии с определённой процедурой. В процессе обучения веса сети постепенно становятся такими, чтобы каждый входной вектор вырабатывал выходной вектор.

Существуют три алгоритма обучения: "с учителем", "без учителя"

(самообучение) и смешанное. В первом случае нейронная сеть располагает правильными ответами (выходами сети) на каждый входной пример. Веса настраиваются так, чтобы сеть производила ответы как можно более близкие к известным. Обучение без учителя не требует знания правильных ответов на каждый пример обучающей выборки. В этом случае раскрывается внутренняя структура данных или корреляции между образцами в системе данных, что позволяет распределить образцы по категориям. При смешанном обучении часть весов определяется посредством обучения с учителем, в то время как остальная получается с помощью самообучения.

Рассмотрим ряд основных архитектур нейронных сетей, успешно применяемых для решения задачи классификации, одна из постановок которой представлена в данном дипломном проекте[13].

Персептрон. Наиболее простой нейросетевой архитектурой является однослойный персептрон. Он состоит из одного слоя искусственных нейронов, соединённых с помощью весовых коэффициентов со множеством входов (рисунок 14). Слой - это совокупность нейронов с единым входным сигналом. Элемент Σ умножает каждый вход x на вес w и суммирует взвешенные входы. Если эта сумма больше заданного порогового значения, выход равен единице, в противном случае – нулю.

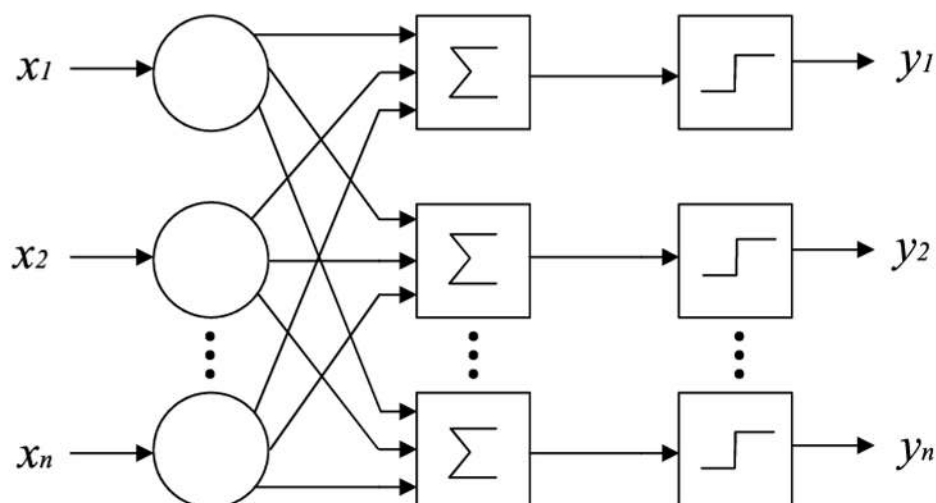


Рисунок 14 – Персептрон

Персептрон обучают, подавая множество образов по одному на его вход и подстраивая веса до тех пор, пока для всех образов не будет достигнут требуемый выход.

Применение такой топологии оправдано только для задач, обладающих высокой линейностью. Например, можно построить нейронную сеть, разбивающую точки (0,0) и (1,1) на два класса для двумерного сигнала, но невозможно решить задачу по разбиению точек (0,0), (1,1) – в первый класс, и (0,1), (1,0) – во второй. Это широко известный пример неспособности простого персептрона решить задачу «исключающее или».

Имеется обширный класс функций, не реализуемых однослойной сетью. Вероятность того, что случайно выбранная функция окажется линейно разделимой, весьма мала даже для умеренного числа переменных. По этой причине однослойные персептроны на практике ограничены простыми задачами[13].

Многослойный персептрон. Серьезное ограничение однослойными сетями можно преодолеть, добавив дополнительные слои. Например, двухслойные сети можно получить каскадным соединением двух

однослойных сетей. Они способны выполнять более общие классификации. Такая сеть может моделировать функцию практически любой степени сложности, причём число слоёв и число элементов в каждом слое определяют сложность функции.

На рисунке 15 изображена двухслойная сеть, которая может обучаться с помощью процедуры обратного распространения.

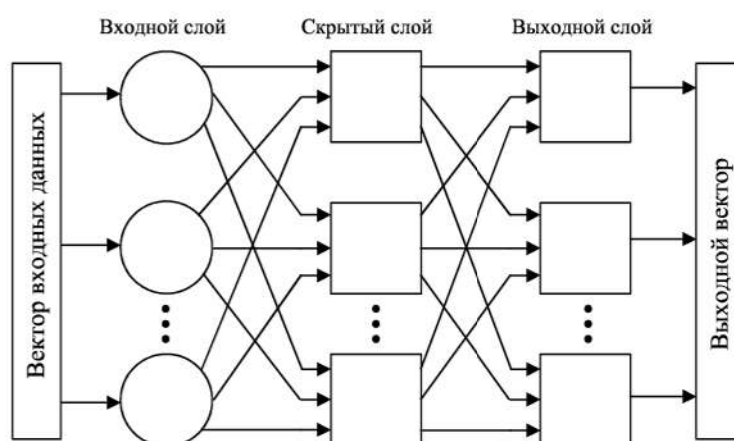


Рисунок 15 – Двухслойный персептрон

Нейроны одного и того же слоя друг с другом не связаны, и каждый нейрон связан со всеми нейронами последующего слоя (кроме последнего слоя - его выходы являются выходами сети в целом). Первый слой нейронов (соединённый с входами) служит лишь в качестве распределительных точек, суммирования входов здесь не производится. Входной сигнал просто проходит через них к весам на их выходах. Получив входные сигналы, нейрон следующего слоя суммирует их с соответствующими весами, затем применяет к этой сумме передаточную функцию и передаёт результат на один из входов нейрона второго слоя, который в свою очередь, суммирует полученные от первого слоя сигналы с некоторыми весами и т.д. Прямое функционирование сети описывается следующими соотношениями:

$$y^0 = x, \quad (2.7)$$

$$y_j^k = f\left(\sum_{i=1}^{n(k-1)} w_{ij}^k y_i^{k-1}\right), j = 1 : n(k), \quad (2.8)$$

где x - входной сигнал;

y_j^k - значение j -го выхода нейрона k -го слоя;

w_{ij}^k - вес связи от i -го нейрона $(k-1)$ -го слоя к j -му нейрону k -го слоя;

f - функция активации;

$n(k)$ - число нейронов в k -м слое.

В качестве активационной функцией в сетях обратного распространения обычно используется сигмоидальная функция. Многослойные нейронные сети обладают большей представляющей мощностью, чем однослойные, только в случае присутствия нелинейности. Сжимающая функция обеспечивает требуемую нелинейность [13].

Для обучения многослойных нейронных сетей применяется алгоритм обратного распространения ошибки. Если при прямом функционировании входной сигнал распространяется по сети от входного слоя к выходному, то при подстройке весов ошибка сети распространяется от выходного слоя к входному.

2.3.5 Вывод

На данный момент существует немного методов классификации речевого сигнала, позволяющих решать задачу идентификации диктора по голосу. Многие из них ещё находятся в стадии разработки и закрыты от общего обзора, поэтому решено разработать свой модуль идентификации диктора. Для решения задачи классификации рационально использовать нейронную сеть с архитектурой трёхслойного персептрона. К её достоинствам можно отнести сравнительную простоту анализа и достаточно высокую эффективность классификации. Благодаря использованию непрерывной функции возбуждения такие сети способны к обобщению обучающей выборки.

3 Программная реализация

Средой для разработки программы был выбран программный пакет **MatLab 12**, т.к. в состав пакета входят готовые библиотеки для работы со звуковыми сигналами. Также в состав **MatLab** входит среда **Guide Builder** для создания приложений с графическим интерфейсом пользователя. Для моделирования структуры нейронной сети был выбран пакет **Neural Networks. Wavelet Toolbox**, входящий в состав **MatLab** является очень удобным инструментом для изучения и проведения вейвлет-преобразований. В стандартный пакет аудио поддержки системы **Matlab** включены функции, позволяющие произвести запись звукового сигнала.

3.1 Описание работы программы

Перед началом работы пользователь должен добавить БД и обучить нейронную сеть. Все эти операции осуществляются через графический интерфейс. Также есть возможность управления записью/воспроизведением звука, открытия и сохранения звуковых файлов с помощью диалоговых окон, а также графического отображения речевого сигнала.

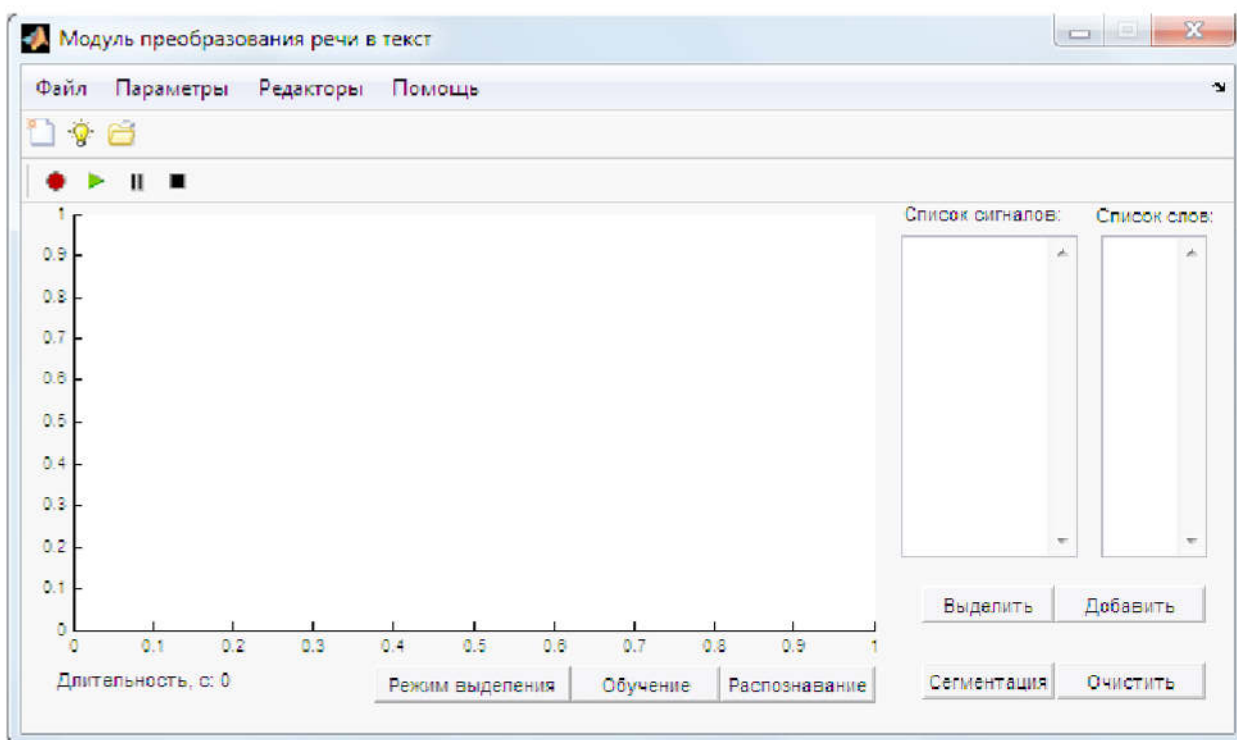


Рисунок 16 – Пользовательский интерфейс программы

На панели управления располагаются кнопки «Play», «Record», «Pause» и «Stop». При помощи которых можно записать и прослушать записанный сигнал. В графической области отображается записанный аудиосигнал или загруженный файл. При нажатии на кнопку «Режим выделения» графическая область поделится на сектора, для более удобного выделения. Выделить нужный фрагмент из потока аудиосигнала, можно выделив нужный отрезок на графической области и нажать на кнопку «Добавить». Данный сегмент отобразится в списке входных сигналов.

Обучающая выборка формируется с помощью команды «Добавить», путём добавления нужных слов эталонов из «Списка сигналов» в «Список слов» с указанием содержания в диалоговом окне. При этом выполняется предварительная обработка, и рассчитываются признаки выбранных слов.

Сформированную обучающую выборку признаков слов можно

сохранить на диск для последующего использования. Сохранение и загрузка выборки выполняется с помощью диалоговых окон, вызываемых через соответствующие команды в меню «Файл» (рисунок 17).

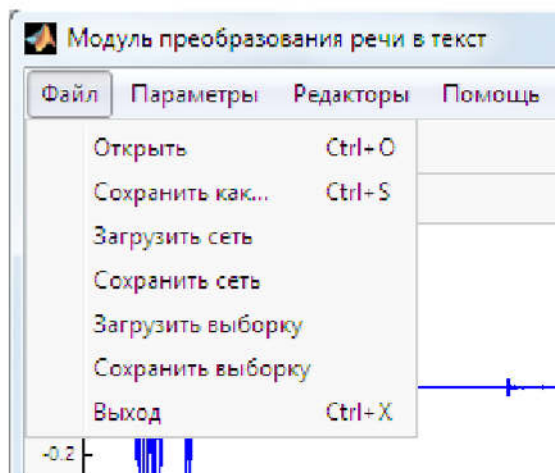


Рисунок 17 – Команды в меню «Файл»

Функция «Обучение» запускает процедуру обучения нейронной сети на сформированном обучающем множестве с заданными параметрами.

Обучение будет направлено на отношение характерных признаков голоса к одному из трёх классов:

- Первый класс - детский голос
- Второй класс - мужской голос
- Третий класс - женский голос

В дальнейшем можно будет увеличить количество классов для более лучшего результата.

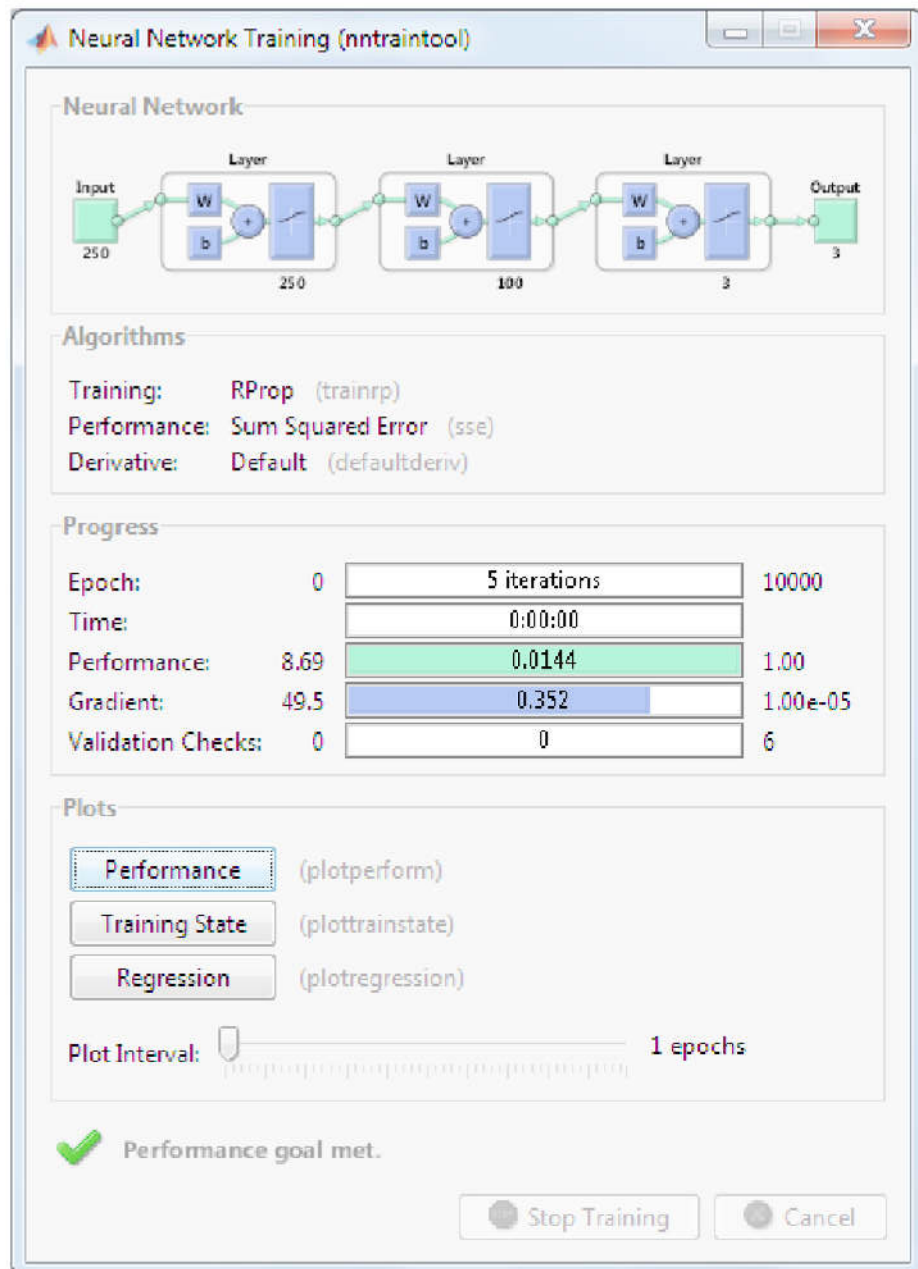


Рисунок 18 – Обучение нейронной сети

В «Neural Network» показано, что нейронная сеть состоит из трёх слоёв (трёхслойный персептрон). Первый слой состоит из 250 нейронов, второй слой из 100 нейронов, 3 слой это количество классов.

Предоставляется возможность сохранить значения весов связей нейронной сети, обученной под конкретного диктора, для дальнейшего

использования при распознавании. Импорт и загрузка сети осуществляется через меню «Файл» (рисунок 17).

Для распознавания необходимо записать аудиосигнал или открыть заранее записанный файл. При нажатии кнопки «Распознать» начнётся процесс распознавания. Загружается уже обученная нейронная сеть, совершается Вейвлет-преобразование с аудиосигналом и так же отправляется на нейронную сеть. Там сравниваются характерные признаки и в текстовом окне на главной форме отображается диктор (рисунок 19), признаки которого наиболее близки к признакам аудиосигнала.

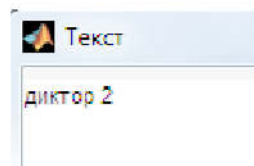


Рисунок 19 – распознанный диктор

3.2 Оценка качества работы модуля идентификации

Модуль голосовой идентификации диктора прошёл тестирование, для наблюдения различий было проведено сравнение с работой системы без модуля голосовой идентификации диктора(таблица 1).

Таблица 1 – Оценка качества работы модуля

Задействованное количество дикторов	Система без модуля идентификации диктора	Система с модулем идентификации диктора
2	9 из 10	10 из 10
3	7 из 10	9 из 10
4	7 из 10	8 из 10
5	6 из 10	8 из 10

Было проведено 4 теста с разным количеством задействованных дикторов, относящихся к разным классам.

- Первый класс – мужской голос.
- Второй класс – женский голос.
- Третий класс – детский голос.

Каждый диктор произнёс по пять слов.

1. Первый тест был проведён с двумя дикторами из 1-ого и 2-ого класса.
2. Второй тест был проведён с тремя дикторами, одного из 1-ого и двух из 2-ого класса.
3. Третий тест был проведён с четырьмя дикторами, одного из 1-ого, двух из 2-ого и 1-ого из третьего классов.
4. Четвёртый тест был проведён с пятью дикторами, двух из 1-ого, двух из 2-ого и 1-ого из третьего классов.

3.3 Вывод

Данный модуль предназначен для улучшения работы системы распознавания речи. С помощью данного модуля система сможет классифицировать дикторов по их характерным признакам голоса, что поможет повысить дикторонезависимость системы и увеличит процент надёжности распознавания диктора. Тесты показали, что надёжность распознавания диктора с модулем голосовой идентификации выросла 15-20%.

ЗАКЛЮЧЕНИЕ

Были проанализированы существующие подходы и выбран алгоритм классификации пользователей по их голосовым характеристикам. Разработан метод выделения признаков речевого сигнала, позволяющий проводить идентификацию дикторов. Была выполнена программная реализация модуля голосовой идентификации диктора с использованием среды **Matlab**. Определено повышение качества распознавания речи за счёт выбора оптимального классификатора, обученного на речевом материале, с схожими с диктором голосовыми характеристиками. В результате экспериментальных исследований разработанного модуля, отмечено повышение дикторнезависимости системы и увеличение надёжности распознавания на 15-20%.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Центр речевых технологий [Электронный ресурс] – режим доступа: <http://www.speechpro.ru/>.
2. Матвеев Ю.Н. Технологии биометрической идентификации личности по голосу и другим модальностям // Вестник МГТУ им. Н.Э. Баумана. Электронное научно-техническое издание. 2012. № 3(3) [Электронный ресурс] – режим доступа: <http://vestnik.bmstu.ru/catalog/it/biometric/91.html/>
3. ОБЗОР ОСНОВНЫХ МЕТОДОВ РАСПОЗНАВАНИЯ ДИКТОРОВЕ.А. Первушин [Электронный ресурс] – режим доступа: <http://cyberleninka.ru/article/n/obzor-osnovnyh-metodov-raspoznavaniya-diktorov.pdf>.
4. Martin A., Przybocki M. The NIST 1999 Speaker Recognition Evaluation - An Overview // Digital Signal Processing. 2000. V. 10
5. Коваль С.Л. Комплексная методика идентификации дикторов по голосу и речи // Информатизация и информационная безопасность правоохранительных органов: труды XX международной научной конференции. М.: Академия управления МВД России, 2011. С. 364-370.
6. Platt J.C. Fast Training of Support Vector Machines using Sequential Minimal Optimization // Advances in Kernel Methods / Ed. by B. Scholkopf, C.C. Burges, A.J. Smola. MIT Press, 1999. P. 185–208.
7. Википедия [Электронный ресурс] – режим доступа: ru.wikipedia.org
8. Дьяконов, В. MATLAB: Учебный курс. - СПб.: Питер, 2001. - 560 с.
9. Распознавание речи. Часть 1. Классификация систем распознавания речи [Электронный ресурс] – режим доступа: <https://geektimes.ru/post/64572/>
10. Идентификация-диктора-по-голосу-текст [Электронный ресурс] – режим доступа: <http://seminar.at.ispras.ru/wp-content/uploads/2012/07/>

11. Ing-Jr Ding, Chih-Ta Yen, Yen-Ming Hsu. Developments of Machine Learning Schemes for Dynamic Time-Wrapping-Based Speech Recognition // Mathematical Problems in Engineering. 2013.
12. Daniel Ramage. Hidden Markov Models Fundamentals // CS229 Section Notes. 2007.
13. Система исследования речевых компонентов В.С. Шерхонов
14. [Электронный ресурс] – режим доступа: <http://www.stel.ru/services/uslugi-po-napravleniyu-rechevye-tehnologii/460/>

ПРИЛОЖЕНИЕ А

Листинг wav_param.m

```
function [P,segnames] = wav_param(Y,phonames,levelwavelet,typewavelet)

xds=Y'

Fram=800;
Nfr=1; Pt=[];
cca=0;
xds=mapminmax(xds);
for ff=1:fix(length(xds)/Fram)
    Nxds=xds(Nfr:Nfr+Fram);

    [c,l]=wavedec(Nxds,levelwavelet,typewavelet);
    DCELL=detcoef(c,l,'cells');

    for kk=1:10
        NRG(kk)=sum(abs(DCELL{kk}))';
    end

    Pt(:,ff)=[NRG]';
    Nfr=Nfr+Fram+1;
end

cca=cca+1;
if cca>1
    P=[P,Pt];
else
    P=Pt;
end;

segnames=0;
```

Листинг net_train.m

```
function [net] = net_train(WDS,S1,net_goal)
WDS=getappdata(gcf,'WDS')

alphab='';
count=0; cc=0;

ff=size(WDS,2);
WDS2=struct();

for ii=1:ff
    if ii>1
        WDS2(ii-1).word=WDS(ii).word;
        WDS2(ii-1).fts=WDS(ii).fts;
    end
end;

WDS=WDS2;
ff=ff-1;

for ii=1:ff
    cc=WDS(ii).word;
    have=0;

    for jj=1:size(alphab,1)

        ll= alphab(jj,:);
        if strcmp(alphab(jj,:),cc);
            have=1 ;
        end;
    end;
    if (have==0)
        count=count+1;
        alphab=strvcat(alphab,cc);
    end;
end;
count

V=zeros(count,count);

for ii=1:count
    V(ii,ii)=1;
end;

T=zeros(count,length(2));

for ii=1:size(WDS,2)
    for jj=1:count

        if (strncmp(WDS(ii).word,alphab(jj,:),size(alphab(jj,:),1)))

            T(jj,ii)=1;
        end;
    end;
end;
```



```

end;

Pm=250;
P=zeros(250,ff);
temp=0; dm=0; ztemp=0;

for ii=1:size(WDS,2)
nframes=size(WDS(ii).fts);
nframes=nframes(2)*10;
temp=reshape(WDS(ii).fts,1,nframes);
temp=temp(:);
    dm=Pm-size(temp,1);
    temp=[temp;zeros(dm,1)]

    P(:,ii)=temp;
end
P

pr=minmax(P);

S1= 250;
S2= 100;
S3=count;

net = newff( pr , [S1 S2 S3 ] ,{'logsig' 'logsig' 'logsig'},'trainrp');

net.performFcn = 'sse';
net.trainParam.goal = net_goal;
net.trainParam.show = 20;
net.trainParam.epochs = 10000;
net.trainParam.mc = 0.95;

net = init(net);
[net,tr] = train(net,P,T);
net.userdata=alphab

```

Листинг recognize.m

```
function [phonems] = recognize(masW, fs, net, diction, levelwavelet, typewavelet)

cnt=1;
Fram=800;
Nfr=1;
alphabet=net.userdata;
P=[]; Pm=250;
xds=masW';
Ln=length(masW');

xds=mapminmax(xds);
    while Nfr<=(Ln-Fram)

        Nxds=xds(Nfr:Nfr+Fram);
        [c, l]=wavedec(Nxds, levelwavelet, typewavelet);

        DCELL2=detcoef(c, l, 'cells');
        Nfr=Nfr+Fram+1;

        for kk=1:10
            NRG2(kk, 1) = sum(abs(DCELL2{kk}))';

        end
        NRG2;
        P=[P;NRG2];

    end

    dm=Pm-size(P, 1);
    P=[P;zeros(dm, 1)]

    Q(:, cnt)=sim(net, P)
    [M, ind] = max(Q(:, cnt)');

    plot(P)

    outword=alphabet(ind, :)

H=text_out;
data = guidata(gcf);
set(data.edit1, 'string', outword);

end
```