

Перв. примен.		Содержание	
Справ. №		Введение4	
		1 Техническое задание5	
		1.1 Назначение разработки и область применения5	
		1.2 Технические требования.....5	
		2 Анализ технического задания.6	
		2.1 Выбор операционной системы6	
		2.2 Выбор языка программирования9	
		2.3 Выбор среды разработки11	
		2.4 Анализ способа работы со звуковыми данными.12	
		3 Разработка структуры системы идентификации пола диктора по голосу.19	
		4 Разработка алгоритма распознавания пола диктора по голосу.....21	
		5 Разработка программных средств.....25	
		6 Тестирование системы27	
		Заключение.....30	
		Список литературы.....32	
Подпись и дата			
Инв. № дубл.			
Взам. инв. №			
Подпись и дата			
Инв. № подл.			
		ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ)	

Перв. примен.					
Справ. №					
Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					

Введение

Популярной проблемой в научном сообществе является проблема автоматического распознавания и понимания речи. Потребность в исследовании этой проблемы стимулируется практическими требованиями, связанными с созданием систем военного и коммерческого назначения.

В последнее время среди различных предприятий и государственных учреждений стала возрастать потребность в системах безопасности с использованием биометрических технологий, а это – один из ключевых факторов на мировом рынке распознавания речи.

Одной из проблем, с которыми сталкиваются при создании современных систем распознавания, является проблема идентификации пола диктора на основе записи голоса.

Большинство современных программных продуктов, направленных на решение задачи распознавания речи, ограничиваются задачами перевода речи в текст, или же выполнения алгоритмов на основе голосовых команд, и при этом не учитывают физиологические параметры речи говорящего, не выполняют так называемую "настройку на диктора", что существенно снижает устойчивость показателей данных систем.

В связи с этим возникла идея создания программной системы, которая позволит определить пол диктора на основе физиологических характеристик речи, что станет первым шагом к решению проблемы неустойчивости систем распознавания, обусловленной отсутствием привязки речи к типу голоса диктора.

Таким образом, целью данной работы является создание программной системы распознавания пола диктора по голосу. Полных функциональных аналогов данной системы нет на рынке, это обусловлено, в первую очередь, тем, что задача определения пола диктора является лишь частью более широкой задачи - точной идентификации личности диктора, и исследования особенностей его речи. Задача разрабатываемой системы заключается в том, чтобы подготовить почву для дальнейших разработок в этой области, и получить возможность применять разработанную технологию при работе над аналогичными задачами в дальнейшем.

Изм.	Лист	№ докум.	Подпись	Дата	ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ)	Лист
						4

Перв. примен.	<h1>1 Техническое задание</h1> <h2>1.1 Назначение разработки и область применения</h2> <p>Разрабатываемая программная система предназначена для реализации алгоритма определения пола диктора по голосу, основанного на методе "k ближайших соседей" и методе опорных векторов. Для работы с данной системой необходима заранее подготовленная база записей голосов в определённом формате (.wav).</p> <p>Области применения разрабатываемой системы:</p> <ul style="list-style-type: none"> – определение пола диктора на записях, выбранных пользователем из специальной базы; – самообучение системы на основе подготовленной базы тренировочных записей; <p>Разрабатываемая система предназначена для использования на стационарных и портативных компьютерах.</p>				
	Справ. №	<h2>1.2 Технические требования</h2> <p>Рассмотрим требования, предъявляемые разрабатываемой системой к ЭВМ:</p> <ul style="list-style-type: none"> – операционная система Microsoft Windows начиная с версии Vista и выше; – требования к аппаратному обеспечению определяются операционной системой; – клавиатура, мышь, дисплей; – динамики. <p>Рассмотрим, каким функционалом должна обладать разрабатываемая система определения пола диктора по голосу:</p> <ul style="list-style-type: none"> – перед началом работы система должна выполнить самообучение, основываясь на подготовленной базе тренировочных записей, для каждой из которых указан класс, определяющий принадлежность диктора, голос которого содержится в записи, к какому-либо полу. – система должна предоставить пользователю возможность выбрать трек для тестирования, и загрузить его на обработку; – произвести анализ выбранной пользователем записи, определив, к какому классу относится данная запись (в данной программной системе существуют два класса, характеризующие пол диктора: "Male" и "Female"), опираясь на информацию, полученную во время самообучения; – вывести на экран результаты анализа, и предоставить пользователю возможность выбрать новый трек для проверки. 			
Подпись и дата		Инв. № дубл.	Взам. инв. №	Подпись и дата	Инв. № подл.
<div> <div>ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ)</div> <div>Лист</div> <div>5</div> </div>					

Перв. примен.	<h2>2 Анализ технического задания.</h2>				
	<h3>2.1 Выбор операционной системы</h3>				
Справ. №	<p>Выбор операционной системы - важная задача на этапе подготовки разработки. Среди множества операционных систем, обладающих своими особенностями, влияющими на разработку, необходимо определить ту, которая подойдёт для нашей программной системы наилучшим образом. Рассмотрим наиболее популярные из всех операционных систем: Windows, Mac OS и Linux:</p>				
	<p>– ОС Windows – самая распространённая среди рядовых пользователей операционная система. Поставляется компанией Microsoft Corporation для персональных компьютеров с 1985 года. Применение данной операционной системы не ограничивается использованием на домашних компьютерах, она так же часто встречается на предприятиях, где требуется исполнение более сложных задач, и более высокая надёжность. Windows была первой операционной системой, взявшей ориентацию на графический интерфейс при управлении компьютером, что стало своего рода прорывом. Все когда-либо существовавшие версии Windows можно условно разделить на 5 семейств: расширения системы MS-DOS (в эту группу входят все версии до Windows-95), семейство Windows-9x (все версии до Windows ME), семейство Windows NT (спереходом на 32/64 бита, к этому семейству относятся все современные версии, включая Windows 10), а так же семейство Windows для смартфонов, и семейство встраиваемых ОС (Windows Embedded).</p>				
Подпись и дата	<p>– Mac OS – вторая по популярности операционная система среди пользователей домашних компьютеров, разработана компанией Apple, впервые введена на рынок в 1984 году в качестве операционной системы для машины Macintosh, и называлась первая версия просто - System Software. Своё современное название приобрела после выпуска версии 7.6. Главной отличительной чертой Mac OS от Windows является то, что Mac OS основана на системе Unix, а, следовательно, она считается частью семейства операционных систем Unix, равно как и Linux.</p>				
	<p>– Linux – общее название Unix-подобных операционных систем, основанных на ядре Linux Kernel. Его первая версия была опубликована автором в 1991 году. Важная отличительная особенность LinuxKernel- открытость кода. То есть эта операционная система создаётся и распространяется в соответствии с моделью разработки свободного и открытого программного обеспечения. Так как Linux Kernel- лишь основа операционной системы, то не существует никакой официальной комплектации ОС, и обычно Linux- системы распространяются в виде бесплатных готовых дистрибутивов, каждый из</p>				
Инв. № подл.					Лист
	<p>ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ)</p>				
Изм.	Лист	№ докум.	Подпись	Дата	6

Перв. примен.					
Справ. №					
Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					

которых имеет свой набор прикладных программ и свою специфику, направленную на удовлетворение конкретных нужд пользователя. Не смотря на то, что ОС Linux появилась на рынке позже основных конкурентов, её популярность неуклонно растёт. Это обусловлено как бесплатным распространением, так и большой гибкостью системы, позволяющей наилучшим образом использовать все возможности компьютера для решения конкретных задач пользователей.

Теперь рассмотрим основные различия операционных систем Windows, MacOS и Linux, а так же их преимущества и недостатки:

1) Защищённость.

Если рассматривать критерий защищённости системы от внешних воздействий и нарушений работы, то данному критерию операционная система Windowsуступает иLinux,и MacOS. Причина этого - популярность операционной системы Windows, следствием которой стало порождение большого количества вредоносных программ, направленных на нарушение работы данной системы.

2) Стабильность работы.

По этому критерию наилучшей системой считается операционная система Linux. Сбои, нарушающие работу, происходят на ней гораздо реже, чем на Windows или MacOS.

3) Оборудование, необходимое для работы с системой.

Системой, проигрывающей по данному критерию, бесспорно, является Mac OS. ОС Windows и ОС Linux можно установить практически на любое оборудование, предоставленное различными производителями (Hewlett Packard,Acer,Dell, и т.д.), в то время как для установки Mac OS подойдёт только оборудование от фирмы Apple.

4) Необходимое программное обеспечение.

Благодаря наибольшей популярности операционной системы Windows она имеет наибольшую библиотеку прикладного программного обеспечения, по сравнению с Linux и Mac OS, следовательно, большинство программ написаны для Windows, и далеко не все из них имеют аналоги, предназначенные для установки на другие операционные системы.

5) Разработка и применение программного обеспечения.

Большинство программ для операционной системы Windows написаны на таких языках программирования как C++, C#, Java и Visual Basic. Некоторые из этих языков программирования работают на операционных системах Mac OS и Linux, но новые и наиболее распространенные из них работают только в операционной системе Windows, например Visual Basic и C #.

Изм.	Лист	№ докум.	Подпись	Дата	ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ)	Лист
						7

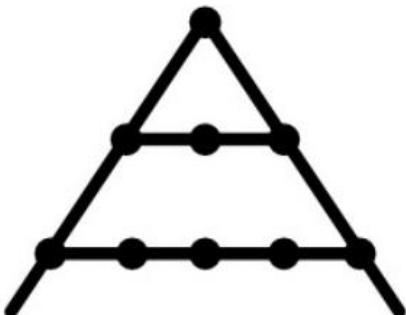
Перв. примен.	<p>Для разработки программной системы определения пола диктора по голосу была выбрана ОС Windows. Это обусловлено тем, что, несмотря на относительную незащищённость, данная операционная система обладает наибольшим выбором программного обеспечения для разработки, а в нашем случае это очень важно, так как наиболее известные языки программирования не подходят для работы с нашим проектом. В целом ОС Windows полностью соответствует необходимым технологическим и функциональным требованиям для разработки данного проекта.</p>					
Справ. №						
Подпись и дата						
Инв. № дубл.						
Взам. инв. №						
Подпись и дата						
Инв. № подл.						
					ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ)	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

Перв. примен.	<h2 style="text-align: center;">2.2 Выбор языка программирования</h2> <p>Одним из важнейших этапов разработки программного обеспечения является выбор языка программирования. Для того, чтобы определиться с тем, на каком языке будет написана наша программная система, предлагаем рассмотреть особенности и направленность нескольких популярных языков программирования: C++, Java и R.</p> <p>Рассмотрим их более подробно:</p> <ul style="list-style-type: none"> – C++ – объектно-ориентированное расширение языка C, созданное Бьярном Страуструпом в 1980 году. Строго типизированный компилируемый язык программирования общего назначения, поддерживающий такие типы программирования, как: структурное, объектно-ориентированное и обобщённое. – Java –объектно-ориентированный язык программирования. Был создан компанией Sun в начале 1990-х годов на основе языка C++. Главная особенность Java - то, что компилируется этот язык не в машинный код, а в платформно-независимый байт-код, который может выполняться с помощью интерпретатора - виртуальной Java-машины, что делает этот язык частично платформно-независимым. Был призван упростить разработку приложений на C++, путём исключения из него низкоуровневых возможностей. Изначально разрабатывался для бытовой электроники, но впоследствии стал использоваться для написания апплетов, приложений и серверного программного обеспечения. – R – это система, состоящая из языка программирования и среды для статистических вычислений и графического анализа. Является одним из самых популярных средств для анализа и визуализации данных с открытым кодом, предназначенным для ОС Windows, Mac OS и Linux. Основными особенностями системы R являются: то, что в ней реализовано большинство известных на данный момент способов анализа данных; имеется возможность визуализации сложных данных; имеется возможность наиболее просто импортировать данные из самых разных источников (файлов, СУБД, специализированных хранилищ, и т.д.); имеет реализацию новых сложных статистических процедур, более нигде не реализованных. <p>Для разработки данной программной системы был выбран именно язык R, и решающим фактором в выборе стала ориентированность этого языка, и разработанной для него среды вычислений на реализацию анализа данных и разработку программ, реализующих новые статистические методы.</p> <p>Для языка R реализованы следующие библиотеки, использование которых пригодиться нам в разработке программной системы:</p>															
	Справ. №															
Подпись и дата		Инв. № дубл.	Взам. инв. №	Подпись и дата	Инв. № подл.											
	<div style="display: flex; justify-content: space-between; align-items: center;"> <div> <table border="1" style="width: 100%;"> <tr> <td>Изм.</td> <td>Лист</td> <td>№ докум.</td> <td>Подпись</td> <td>Дата</td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </table> </div> <div style="text-align: center; flex-grow: 1;"> ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ) </div> <div style="text-align: right;"> <table border="1" style="width: 100%;"> <tr> <td>Лист</td> </tr> <tr> <td style="text-align: center;">9</td> </tr> </table> </div> </div>					Изм.	Лист	№ докум.	Подпись	Дата						Лист
Изм.	Лист	№ докум.	Подпись	Дата												
Лист																
9																

Перв. примен.																	
Справ. №																	
<p>– audio – интерфейсы для аудио-устройств (в основном семплерных) для обеспечения записи и воспроизведения звука. Встроенные устройства включают в себя Windows, MM, Mac OS X, AudioUnits и PortAudio.</p> <p>– seewave – содержит функции для анализа, обработки, отображения, редактирования и синтеза временных волн (в частности, звуковых). Этот пакет выполняет временной анализ (осциллограмм и конверты), анализ спектрального состава, добротности резонанса, энтропии, кросс-корреляции и автокорреляции пересечения нулевого уровня, доминирующей частоты, аналитического сигнала, частоты когерентности, 2D и 3D спектрограмм и т.д.</p> <p>– e1071 – функции для выполнения латентного анализа, преобразования Фурье короткого времени, нечеткой кластеризации, машины опорных векторов, алгоритма расчета кратчайшего пути, и т.д.</p> <p>– kknnp – функции для реализации взвешенного метода "k ближайших соседей" для классификации, регрессии и кластеринга.</p>																	
Подпись и дата																	
Инв. № дубл.																	
Взам. инв. №																	
Подпись и дата																	
Инв. № подл.																	
<table border="1"> <tr> <td>Изм.</td> <td>Лист</td> <td>№ докум.</td> <td>Подпись</td> <td>Дата</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>					Изм.	Лист	№ докум.	Подпись	Дата						<table border="1"> <tr> <td>Лист</td> </tr> <tr> <td>10</td> </tr> </table>	Лист	10
Изм.	Лист	№ докум.	Подпись	Дата													
Лист																	
10																	
ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ)																	

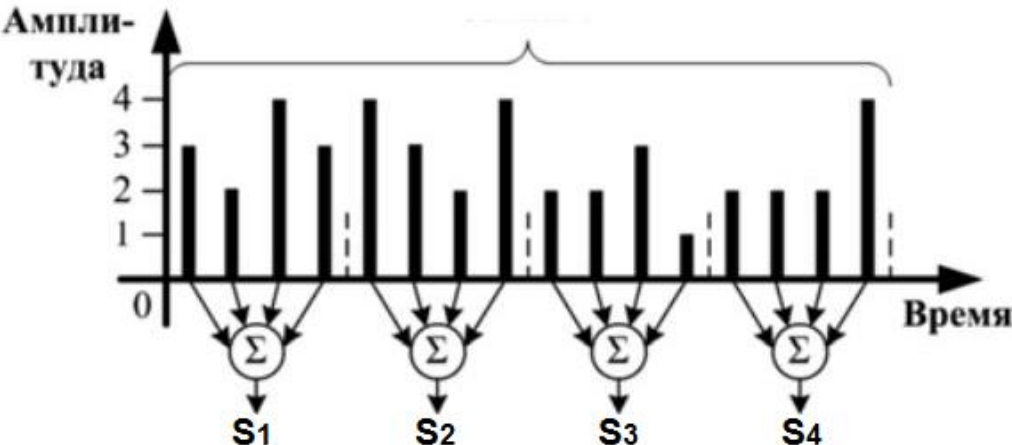
Перв. примен.		<h3>2.3 Выбор среды разработки</h3> <p>Современный выбор сред разработки для языка R достаточно обширен. В него входят как непосредственно среда выполнения, поставляющаяся вместе с языком (в консольном и графическом вариантах), так и средства, поставляющиеся сторонними разработчиками (по сути, большинство из них является графическими интерфейсами к R). Самые заметные из них - RCommander (платформо-независимый интерфейс), RKWard, JGR (Java GUI), SciViews-K (среда для языков с динамической типизацией), Rattle (среда для выявления скрытых закономерностей или взаимосвязей между переменными в больших массивах необработанных данных), PMG и RPMG (максимально простые кросс-платформенные GUI), Rweb (набор скриптов для работы с R через браузер), Gnumeric (табличный процессор), RevolutionREnterprise (высокопроизводительная версия R, ориентированная на многопоточные вычисления и работу с концепцией "big data").</p> <p>Среди всего этого многообразия для разработки данной программной системы нами была выбрана интегрированная среда разработки R Studio. Она сочетает интуитивный пользовательский интерфейс с мощной консолью R, и её основным преимуществом перед остальными интегрированными средами разработки является её бесплатность.</p> <p>R Studio позволяет не только работать с возможностями консоли R, но и создавать для программной системы пользовательский интерфейс с помощью библиотеки RGtk2.</p>					
	Справ. №						
<div>Подпись и дата</div> <div>Инв. № дубл.</div> <div>Взам. инв. №</div> <div>Подпись и дата</div> <div>Инв. № подл.</div>							
Изм.	Лист	№ докум.	Подпись	Дата	ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ)		Лист
							11

Перв. примен.		<h2 style="text-align: center;">2.4 Анализ способа работы со звуковыми данными.</h2> <p>Для реализации алгоритма, необходимого для корректной работы данной программной системы, нам необходимо определить способ обработки звуковых данных.</p> <p>Для большинства существующих методов обработки цифровых данных можно выявить следующие проблемы [2]:</p> <ul style="list-style-type: none"> – проблема формирования исходного описания сигнала: данная проблема связана с тем, что существующие модели и методы распознавания адаптированы к конкретному классу прикладных задач и требуют априорного знания свойств анализируемых сигналов; – проблема формирования системы признаков, связанная с выбором конечного множества признаков, обеспечивающих однозначность решения задачи классификации на этапе распознавания и отвечающая требованиям необходимости и достаточности. Этап выбора системы признаков необходим для сокращения размерности входного описания. Учитывая, что задача сокращения размерности – оптимизационная задача, то для её решения необходимо использование критерия информативности. Отсутствие модели априорной неопределённости и модели её раскрытия породило большое количество методов в выборе критерия информативности, что, в свою очередь, порождает большое число возможных вариантов признаков; – проблема принятия решений в условиях априорной неопределённости. Этап принятия решения заключается в сравнении с имеющимся эталоном признакового описания анализируемого вибросигнала. Предполагается, что эталону соответствует компактное множество точек в системе признаков. Однако помехи, структурные изменения одного и того же представителя класса приводят к перекрытию классов. Поэтому проблема принятия решения замыкается на проблемы формирования системы признаков, позволяющей сформировать эталон, имеющий компактное представление. Использование методов теории активного восприятия позволяет решить описанные проблемы. <p>Основной алгоритм, основанный на применении теории активного восприятия включает в себя три этапа:</p> <ul style="list-style-type: none"> –этап предварительной обработки сигнала (формирование исходного описания сигнала); – формирование системы признаков сигнала; –этап классификации сигнала на основе системы признаков. <p>Предварительная обработка сигнала - первый шаг в реализации выбранного алгоритма. Целью данного этапа является представление сигнала в удобной форме для</p>				
	Справ. №					
Подпись и дата		<div style="display: flex; justify-content: space-between; align-items: center;"> <div> <div>Изм.</div> <div>Лист</div> </div> <div> <div>№ докум.</div> <div>Подпись</div> <div>Дата</div> </div> <div style="border: 1px solid black; padding: 5px;"> ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ) </div> <div> <div>Лист</div> <div>12</div> </div> </div>				
	Инв. № дубл.					
Взам. инв. №						
Подпись и дата						
Инв. № подл.						

Перв. примен.	<p>последующего анализа, а именно - формирование его спектрального представления с помощью U-преобразования.</p> <p>Первым этапом U-преобразования является фильтрация сигнала. Она выполняется с целью компенсации помех с помощью уравнения Фредгольма первого рода:</p> $g(x) = \int_{t \in T} K(x, t) f(t) dt, \quad (1)$ <p>где $g(x)$ – наблюдаемая функция, $f(t)$ – неизвестная функция (априори неизвестное изменение во времени амплитуды сигнала на входе), $K(x, t)$ – ядро преобразования.</p> <p>Следующий этап U-преобразования - Q-преобразование, призванное избавить нас от некорректности, возникающей в результате преобразования Фредгольма. Для этого в уравнении Фредгольма в качестве ядра $K(x, t)$ используем единичную весовую матрицу, и приведём его к следующему виду:</p> $g(x) = \int_{t \in T} f(t) dt, \quad (2)$ <p>Теперь уравнение (2) является корректным, так как из него исключена операция дифференцирования. Теперь, чтобы получить информацию о сигнале $f(t)$ в целом, выполним операцию интегрирования:</p> $m(\bar{T}) = \int_{t \in \bar{T}} f(t) dt, \quad (3)$ <p>Основываясь на результате интегрирования $m(\bar{T})$ мы можем сказать, содержит ли исследуемый сигнал какую-либо информацию: если $m(\bar{T})$ отлично от нуля, то мы можем утверждать, что сигнал полезен, в случае же, когда $m(\bar{T})$ равен нулю, дальнейший анализ сигнала не имеет смысла.</p> <p>Особенностью Q-преобразования является то, что за одно измерение оно позволяет получить один бит информации об анализируемом сигнале, и выявить его структурный элемент. Таким образом, последовательное применение Q-преобразования к сигналу позволяет сгенерировать пирамидальную структуру, основанную на операциях дихотомии:</p> <div style="text-align: center; margin: 20px 0;"> <div style="display: flex; justify-content: space-between; align-items: center;"> <div>Уровень 0</div> <div>$m_{00}(T)$</div> </div> <div style="display: flex; justify-content: center; align-items: center; margin: 10px 0;">  </div> <div style="display: flex; justify-content: space-between; align-items: center;"> <div>Уровень 1</div> <div>$m_{1j}(T_j)$</div> </div> <div style="display: flex; justify-content: space-between; align-items: center;"> <div>Уровень 2</div> <div>$m_{2j}(T_j)$</div> </div> </div> <p>Рисунок 1 - пирамидальная структура отсчётов.</p>				
	Справ. №				

Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					

					ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ)	Лист
						13
Изм.	Лист	№ докум.	Подпись	Дата		

Перв. примен.	<p>Мы видим, что в рамках теории активного восприятия входящий сигнал рассматривается как системное преобразование. Для обнаружения его структурных элементов используется операция интегрирования, а для обнаружения связей между этими элементами - пространственное дифференцирование. Вместе эти два преобразования и называются U-преобразованием:</p> $U = d\int \tag{4}$ <p>Теперь можно приступить к следующему этапу нашего алгоритма - формированию системы признаков сигнала. Делается это следующим образом: всю полученную массу измерений (отсчётов) разобьём на 16 частей, и сложим амплитуды отсчётов, относящиеся к каждой части:</p>  <p>Рисунок 2 - сложение амплитуд отсчётов сигнала</p> <p>Затем к получившемуся массиву сумм S применим базисные одномерные фильтры-покрытия, позволяющие получить набор спектральных коэффициентов. Исследования показали, что для целей анализа звукового сигнала необходимо использовать не менее 16 фильтров:</p>				
Справ. №					
Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					

					ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ)	Лист 14
Изм.	Лист	№ докум.	Подпись	Дата		

Перв. примен.					
Справ. №					
Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					

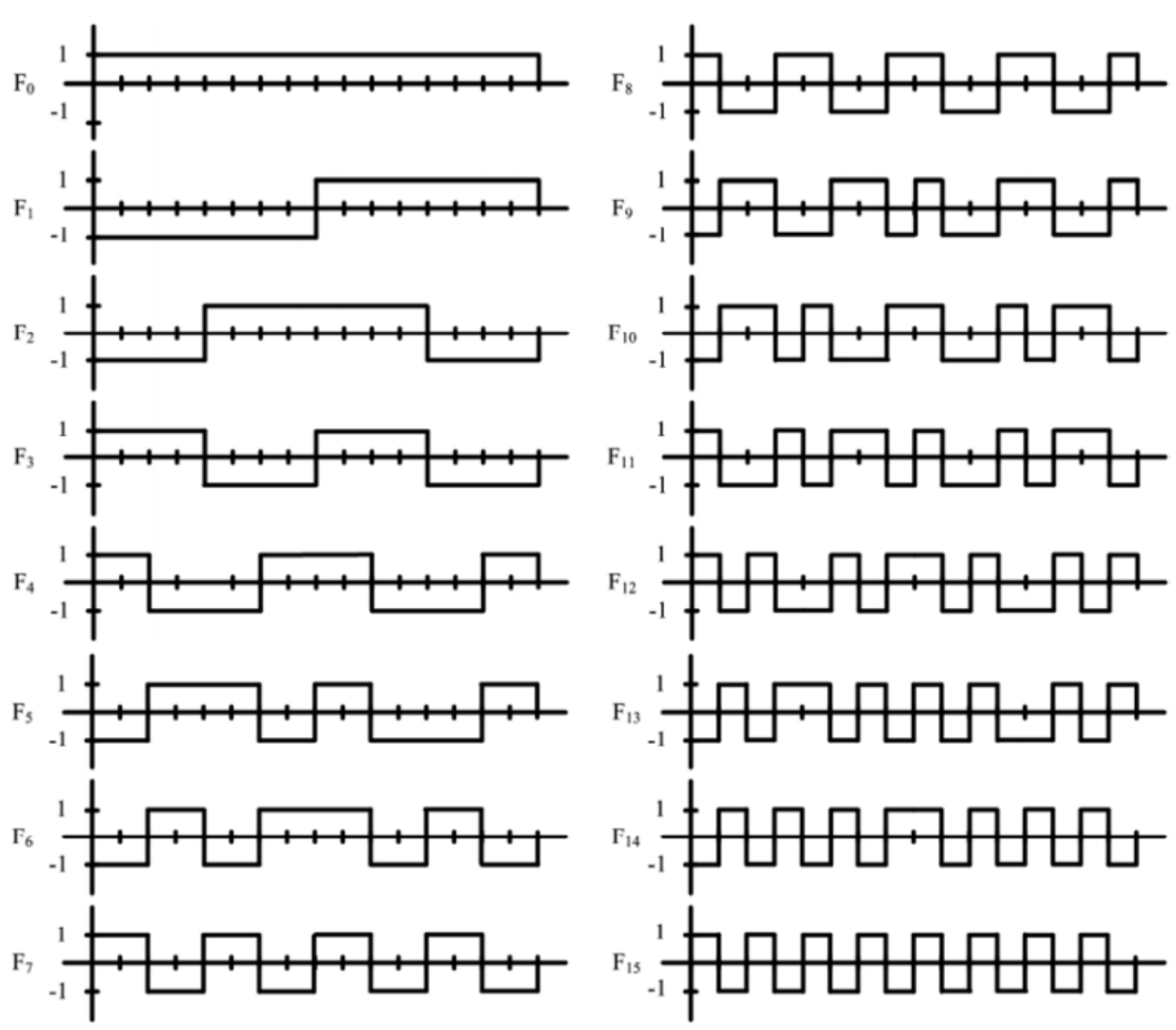


Рисунок 3 - фильтры-покрытия

Фильтры применяются следующим образом: выполняется операция сложения над суммами отсчётов, причём те суммы, которые попали в положительную область фильтра, берутся с положительным знаком, а те, которые попали в отрицательную область - с отрицательным. Таким образом, для каждого фильтра мы получаем некоторое число, называемое спектральным коэффициентом. С этими-то числами и будет производиться дальнейшая работа. Для начала формируется огрублённое представление спектрального представления сигнала для приведения значений спектральных коэффициентов сигнала к конечному, заранее заданному набору амплитуд. Для выполнения этого алгоритма задаётся степень огрубления - k . После получения огрублённого представления сигнала необходимо определить так называемые ключевые сегменты - участки сигнала, на которых максимален отклик небольшого количества операторов. Поиск ключевых сегментов используется для того, чтобы можно было уменьшить количество

						ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ)	Лист
Изм.	Лист	№ докум.	Подпись	Дата			15

Перв. примен.					
Справ. №					
Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					

SVM основан на максимальных пределах линейных дискриминантов, и напоминает модель с вероятностным подходом, с тем исключением, что здесь не учитываются зависимости между атрибутами объектов модели. В отличие от традиционных подходов к работе с нейронными сетями, модели в которых сильно страдают от зашумления данных вследствие используемых алгоритмов оптимизации для выбора параметров и статистических показателей, svm больше направлен на принципы минимизации структурных рисков (SRM), минимизирующих верхнюю границу ожидаемого риска, а не эмпирических (ERM), минимизирующих ошибки обучающих данных, что позволяет этому алгоритму получать многообещающие эмпирические показатели.

Классификатор, как правило, включает в себя тренировочные и тестовые наборы, состоящие из экземпляров данных. Каждый экземпляр в наборе содержит одно целевое значение (класс, в нашем случае этот класс определяет пол диктора), и несколько атрибутов (признаков). Цель классификатора заключается в создании модели, способной прогнозировать целевые значения экземпляров данных в наборе тестирования, для которых известны только атрибуты. В общем и целом задачу классификации можно рассматривать как проблему двух классов, имеющую цель в том, чтобы отделить один класс от другого с помощью функции, индуцированной из доступных примеров. Основная задача здесь - создать классификатор, который будет хорошо обобщать данные. Идеей svm является поиск оптимальной гиперплоскости, то есть такого линейного классификатора, который обеспечит максимальную дистанцию между собой и ближайшими примерами объектов из каждого класса (пример представлен на рисунке ниже):

Рисунок 4 - пример разделяющих линейных классификаторов

KNN (k-NearestNeighbors)

KNN- очень простой алгоритм обучения на основе объектов. Не смотря на свою простоту, данный алгоритм может предложить довольно хорошую производительность на

Изм.	Лист	№ докум.	Подпись	Дата	ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ)	Лист
						17

Перв. примен.					
Справ. №					
Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					

некоторых задачах. Особенность данного алгоритма в том, что здесь отсутствует явная модель обучения. Несмотря на то, что для knn тоже требуется набор тренировочных объектов, используется он исключительно для того, чтобы заполнить образец пространства поиска объектами с уже известными классами. Ни одна другая модель фактического обучения не начинает выполняться уже на стадии этой фазы, поэтому knn известен так же как "алгоритм ленивого обучения".

В данном алгоритме могут быть использованы различные показатели расстояния между объектами, в зависимости от характера исследуемых данных. Для непрерывных переменных, например, характерно использование Евклидова расстояния. Для каких-то конкретных проблем, например, проблемы классификации текста, используются так же специализированные показатели.

Принцип работы алгоритма следующий: когда для оценки предоставляется экземпляр с неизвестным классом, алгоритм вычисляет некоторое заданное количество (k, обычно нечётное) ближайших соседей этого объекта (объектов, классы которых уже известны), и тогда его класс определяется тем, какой класс наиболее многочислен среди этих соседей. Обычно используется голосование большинством голосов, но в тех случаях, когда классы набирают равное количество голосов, может быть использовано взвешенное голосование, в котором учитывается так же расстояние до исследуемого объекта.

Основными преимуществами knn являются:

- простая реализация;
- прочность касательно пространства поиска;
- небольшое количество параметров для настройки оптимального классификатора: только выбор типа расстояний и параметра k;

Теперь об основных недостатках: дорогое тестирование каждого нового объекта, так как приходится вычислять для него расстояния до всех известных, а их может быть очень много; чувствительность к шумным или незначимым атрибутам; чувствительность к разбалансированным наборам данных, в которых большинство объектов принадлежат к одному или нескольким классам, и поэтому доминируют в большинстве окружений.

Изм.	Лист	№ докум.	Подпись	Дата	ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ)	Лист
						18

Перв. примен.					
Справ. №					
Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					

3 Разработка структуры системы идентификации пола диктора по голосу.

Нашу программную систему можно представить в виде следующих модулей:

Модуль обучения

Модуль тестирования

Модуль реализации классификатора

Графический модуль

Рисунок 5 - Структура программной системы

Рассмотрим эти модули подробнее:

- Модуль обучения - предназначен для создания и обработки массива аудиозаписей, которые будут служить программной системе в качестве набора записей для обучения классификатора. В этом модуле происходят следующие действия:
 - считывание определённых участков тренировочных аудиозаписей;
 - выполнение U -преобразования для каждого участка;
 - определение для каждого участка системы признаков;
 - создание массива признаков всех тренировочных записей;
 - дополнение массива столбцом с заранее определённым признаком класса;
 - преобразование массива в фрейм (формат, пригодный для работы классификатора).
- Модуль тестирования - предназначен для обработки конкретной аудиозаписи, выбранной пользователем, которая будет служить программной системе в качестве тестовой записи. В этом модуле происходят следующие действия:

						Лист
					ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ)	19
Изм.	Лист	№ докум.	Подпись	Дата		

Перв. примен.					
Справ. №					
Подпись и дата					
Инв. № дубл.					
Взам. инв. №					
Подпись и дата					
Инв. № подл.					

- считывание определённых участков аудиозаписи;
- выполнение U -преобразования для каждого участка;
- определение для каждого участка системы признаков;
- создание массива признаков тестовой записи;
- преобразование массива в фрейм.

Стоит заметить, что, помимо модуля тестирования, созданного для работы с одной тестовой записью, в программе так же целесообразно будет реализовать модуль для работы с массивом тестовых записей. Это необходимо для упрощения разработки и тестирования.

- Модуль реализации классификатора - предназначен для подготовки и реализации алгоритма классификации. В этом модуле происходит следующее:
 - создание так называемой модели для классификатора. Для создания модели необходимо применить функцию классификатора, используя в ней подготовленные в тренировочном модуле данные, а так же набор определённых параметров, необходимых конкретному классификатору, имеющих оптимальные значения, выявленные в процессе тестирования;
 - обработка тестовой записи (массива записей) с применением созданной на предыдущем шаге модели с помощью функций классификатора;
 - вывод результатов классифицирования тестовой записи в консоль.
- Графический модуль - модуль, предназначенный для интеракции с пользователем программы. В графическом модуле выполняются следующие действия:
 - пользователю демонстрируется окно программы, на котором присутствует кнопка для выбора аудиозаписи, которую пользователь хочет протестировать, и область вывода результатов;
 - при вводе пути к файлу, и нажатии кнопки ввода, программа начинает выполняться;
 - по завершении классификации файла на экран выводится результат выполнения программы;
 - пользователю предлагается ввести новый файл, чтобы снова выполнить программу;

Изм.	Лист	№ докум.	Подпись	Дата	ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ)	Лист
						20

Перв. примен.		4 Разработка алгоритма распознавания пола диктора по голосу				
Справ. №		Рассмотрим блок-схему работы модуля подготовки тренировочных данных для классификаторов:				
Подпись и дата		Рисунок 6 - блок-схема работы модуля подготовки тренировочных данных				
Инв. № дубл.		Принцип работы модуля подготовки данных следующий:				
Взам. инв. №		Специальная функция считывает отрывки аудиозаписей некоторой определённой				
Подпись и дата		длины (данная программная система работает с аудиозаписями формата .wav) из				
Инв. № подл.		заданных директорий. В файловой системе тренировочные данные располагаются				
		следующим образом: существует две папки, в каждой из которых располагаются				
		аудиозаписи одного класса (в одной - класса “Male”, аудиозаписи мужских голосов, в				
		другой - класса “Female”, женские голоса). Аудиозаписи считываются подряд, сначала из				
		одной папки, затем из другой, что позволяет удобно расположить их в массиве. Над				
		каждым считанным отрывком производятся следующие операции: сначала он поступает				
		на вход функции, выполняющей фильтрацию сигнала, а затем - U-преобразование и				
		расчёт спектральных коэффициентов, затем над этим отрывком производятся операция				
						Лист
						21
		Изм.	Лист	№ докум.	Подпись	Дата

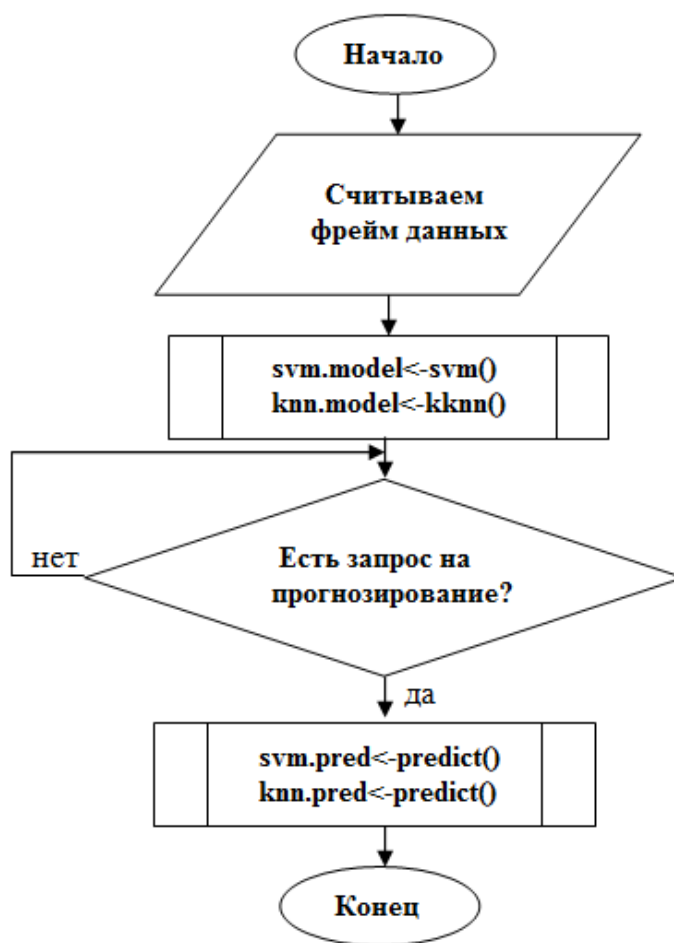



Рисунок 7 - блок-схема работы модуля классификации данных

Затем классификатор производит расчёты, и выводит результат в виде массива, который программа представляет в виде таблицы, анализирует, и выводит таблицу следующего типа:

Результаты классификации		
Тестовые данные	Класс 1	Класс 2
Класс 1	А	В
Класс 2	С	Д

Таблица 1 - результирующая таблица классификатора

Данные в этой таблице располагаются следующим образом: А - характеризует, что классификатор верно определил, что исследуемые данные принадлежат к классу 1, В - характеризует ошибку, когда данным класса 1 классификатор присвоил значение класса 2, С - характеризует, что классификатор верно определил, что исследуемые данные принадлежат к классу 2, Д - характеризует ошибку, когда данным класса 2 классификатор присвоил значение класса 1.

Перв. примен.		<p>Для идеально выполненной классификации таблица должна принимать вид, когда все значения, кроме тех, которые лежат на основной диагонали А-D, стремятся к нулю, а диагональные значения максимальны.</p> <p>Теперь рассмотрим блок-схему работы программы с точки зрения пользователя (интерфейс программной системы):</p>  <pre> graph TD Start([Начало]) --> Prep[Подготовка массива данных для обучения] Prep --> Create[Создание модели классификатора] Create --> Open[/>>"Открыть файл"/] Open --> Select{Файл выбран?} Select -- нет --> Open Select -- да --> Forecast[Прогнозирование] Forecast --> Output[/>>"Вероятно, пол диктора на этой записи:"/] Output --> End([Конец]) </pre>				
Справ. №						
Подпись и дата						
Инв. № дубл.						
Взам. инв. №						
Подпись и дата						
Инв. № подл.		<p>Рисунок 8 - блок-схема работы программной системы по интеракции с пользователем.</p> <p>Как видно из приведённой выше блок-схемы, в модуле интеракции с пользователем происходит следующее: в начале работы программа выполняет весь блок подготовки тренировочных данных, создаёт модель классификации, и затем ожидает от пользователя ввода файлового пути для аудиозаписи, которую пользователь хочет протестировать. При вводе этого пути, и нажатии кнопки ввода, программа выполняет блок подготовки тестовых данных, а затем - прогнозирование на их основе. Полученный результат обрабатывается, и выводится на экран. После этого программа снова входит в режим ожидания дальнейших действий пользователя.</p>				
Изм.	Лист	№ докум.	Подпись	Дата	ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ)	Лист
						24

Перв. примен.	<h2>5 Разработка программных средств</h2> <p>Рассмотрим основные переменные, использованные в нашей программной системе:</p> <ol style="list-style-type: none"> 1) <code>sources <- c("Male", "Female")</code> – массив имён классов; 2) <code>nameNums[]</code> – массив имён тренировочных аудиозаписей; 3) <code>classesMatrix[]</code> - массив, состоящий из одного столбца, наполовину заполненный значениями класса "Male", наполовину - "Female". 4) <code>frame[]</code> –фрейм подготовленных и обработанных данных для использования классификатором (включающий в себя столбец значений из <code>classesMatrix[]</code>); 5) <code>features = 840</code> – количество признаков, определяемых функцией <code>getFeatures()</code> 6) <code>trainsetNum = 40</code> - количество обучающих аудиозаписей 6) <code>testsetNum = 32</code> - количество аудиозаписей, предназначенных для тестирования и отладки программы. <p>Метод <code>getUTrans()</code> предназначен для выполнения <i>U</i>-преобразования. Для его корректной работы необходимо на вход задать анализируемый сигнал <i>f</i>, а так же длину сегментов <code>slen</code> и величину сдвига <code>shft</code> для выполнения преобразования:</p> <pre>getUTrans <- function(f, slen, shft, oper, flt)</pre> <p>Возвращает данный метод массив спектральных значений анализируемого сигнала.</p> <p>Метод <code>getFeatures()</code> предназначен для формирования массива признаков анализируемого сегмента. На вход данный метод получает массив <code>udec</code> (результат <i>U</i>-преобразования, возвращаемый функцией <code>getUTrans()</code>).</p> <pre>getFeatures = function(featname, udec, fullGrp, clsGrp, oper)</pre> <p>Возвращает эта функция набор признаков для анализируемого сегмента (840 признаков, в нашем случае)</p> <p>Метод <code>getSummarrayArray()</code> предназначен подготовки массива обучающих данных. На вход метод принимает массив классов, и базовый путь до директории, в которой располагаются тренировочные записи.</p> <pre>getSummaryArray = function(sources, pathToDir)</pre> <p>Возвращает эта функция массив <code>summaryArray[]</code>. Он составляется следующим образом: сначала выполняется считывание отрывков для анализа, затем над ними поочерёдно проводятся операции <code>getUTrans()</code> и <code>getFeatures()</code>, полученные результаты записываются построчно в искомый массив.</p> <p>Метод <code>setModels()</code> предназначен для создания моделей классификаторов <code>svm</code> и <code>knn</code>.</p> <pre>setModels = function(trainset)</pre>																	
	Справ. №																	
Подпись и дата		Инв. № дубл.	Взам. инв. №	Подпись и дата	Инв. № подл.													
<table border="1"> <tr> <td>Изм.</td> <td>Лист</td> <td>№ докум.</td> <td>Подпись</td> <td>Дата</td> <td rowspan="2"> ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ) </td> <td>Лист</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td>25</td> </tr> </table>						Изм.	Лист	№ докум.	Подпись	Дата	ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ)	Лист						25
Изм.	Лист	№ докум.	Подпись	Дата	ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ)	Лист												
						25												

Перв. примен.	<p>На вход данный метод получает фрейм тренировочных данных, устанавливает модели для knn и svm (для knn устанавливается так же значение kkn.dist - функции расстояний), и ничего не возвращает.</p> <p>Метод <i>getClassSVM()</i> предназначен для выполнения анализа единичной тестовой записи с использованием svm-модели.</p> <pre>getClassSVM = function(fp)</pre> <p>На вход данный метод получает путь к тестовой записи, а возвращает предсказанное имя класса.</p> <p>Метод <i>getClassKnn()</i> предназначен для выполнения анализа единичной тестовой записи с использованием knn-модели.</p> <pre>getClassKnn = function(fp)</pre> <p>Данный метод работает аналогично предыдущему, с той разницей, что он использует другой классификатор.</p>					
	Справ. №					
Подпись и дата	Инв. № дубл.	Взам. инв. №	Подпись и дата	Инв. № подл.		
					<div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> </div> <div> <div>Изм.</div> <div>Лист</div> <div>№ докум.</div> <div>Подпись</div> <div>Дата</div> </div> <div> <div>ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ)</div> <div>Лист</div> <div>26</div> </div>	

Перв. примен.	<h2>6 Тестирование системы</h2> <p>Тестирование нашей программной системы содержит в себе два этапа. Первый этап - тестирование при разработке моделей классификаторов, необходимое для того, чтобы наилучшим образом подобрать параметры модели классификатора, и получить оптимальный результат его работы, а так же выяснить, какой из классификаторов лучше справляется с задачей. Второй этап - тестирование программной системы в готовом виде, с точки зрения пользователя.</p> <p>На первом этапе тестирование проводится непосредственно в момент отладки программы. Основная работа выполняется с использованием функций <code>setModels()</code>, <code>getClassSVM()</code>, <code>getClassKnn()</code>. В функции <code>setModels()</code>, в которой происходит создание моделей для двух классификаторов, необходимо выполнить подбор оптимальных параметров, а затем выполнить классификацию на массиве тестовых записей, чтобы посмотреть примерное соотношение верных и неверных случаев классификации.</p> <p>Для модели <code>svm</code> параметры, предназначенные для изменения, следующие:</p> <p><code>cost</code> - цена нарушения ограничений (соответствует константе <code>C</code> в используемом нами методе <code>C</code>-классификации)</p> <p><code>gamma</code> - параметр, определяющий, насколько мало влияние одного тренировочного объекта на результат классифицирования. Параметр <code>gamma</code> можно рассматривать как обратную величину радиуса влияния объектов, отобранных в качестве модели опорных векторов.</p> <p>Тестирование различных значений параметров <code>cost</code> и <code>gamma</code> дало следующие результаты: оптимальной величиной для <code>cost</code> оказалось значение 5000, для <code>gamma</code> - 0.0005.</p> <p>В случае применения этих параметров для формирования модели при проведении тестирования на массиве из 32 тестовых записей результат следующий: для класса "Female" было угадано 13 из 16 записей, для класса "Male" - 16 из 16. В общем получается, что количество угаданных записей - 29 из 32, а это 90.6%, очень хороший результат.</p> <p>Для модели <code>knn</code> параметры, предназначенные для изменения, такие:</p> <p><code>k</code> - параметр, определяющий число объектов с известным классом, которые будут участвовать в "голосовании".</p> <p><code>dist.meth</code> - параметр, определяющий тип расстояний в модели. Может принимать значения: "Euclidean", "Manhattan", "Minkowski", или "Hamming", но только для тех случаев, когда используются категориальные переменные, а не непрерывные.</p>				
	Справ. №				
Подпись и дата	Инв. № дубл.	Взам. инв. №			
			Подпись и дата		
Инв. № подл.					
Изм.	Лист	№ докум.	Подпись	Дата	<div> <div>ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ)</div> <div>Лист 27</div> </div>

Перв. примен.					
Справ. №					
<div>Подпись и дата</div> <div>Инв. № дубл.</div> <div>Взам. инв. №</div> <div>Подпись и дата</div> <div>Инв. № подл.</div>					
<p>Тестирование различных значений параметров k и <code>dist.meth</code> дало следующие результаты: оптимальной величиной для k оказалось значение 7, для <code>dist.meth</code> было выбрано значение "Euclidean".</p> <p>В случае применения этих параметров для формирования модели при проведении тестирования на массиве из 32 тестовых записей результат следующий: для класса "Female" было угадано 12 из 16 записей, для класса "Male" - 13 из 16. В общем получается, что количество угаданных записей - 26 из 32, а это 81.25%, результат значительно хуже, чем у svm.</p> <p>Для сравнения, распознавание пола диктора с помощью метода Парзена даёт результат до 97%, распознавание пола на основе решения обратной задачи относительно динамики площади голосовой щели и модели одномерного потока через голосовую щель даёт результат до 94,7% точности для распознавания мужского голоса, и до 97,6% - для распознавания женского; использование метода на основе моделирования акустических параметров голоса гауссовыми смесями даёт точность до 91%. Таким образом, видно, что разработанная программная система при использовании для классификации метода опорных векторов даёт нормальный результат по сравнению с другими системами, особенно с учётом использования относительно небольшого количества обучающих записей.</p> <p>Перейдём ко второму этапу тестирования:</p> <p>Интерфейс разработанной системы представлен на рисунке ниже:</p>					
					Лист
<div> <div>Изм.</div> <div>Лист</div> <div>№ докум.</div> <div>Подпись</div> <div>Дата</div> </div>					28

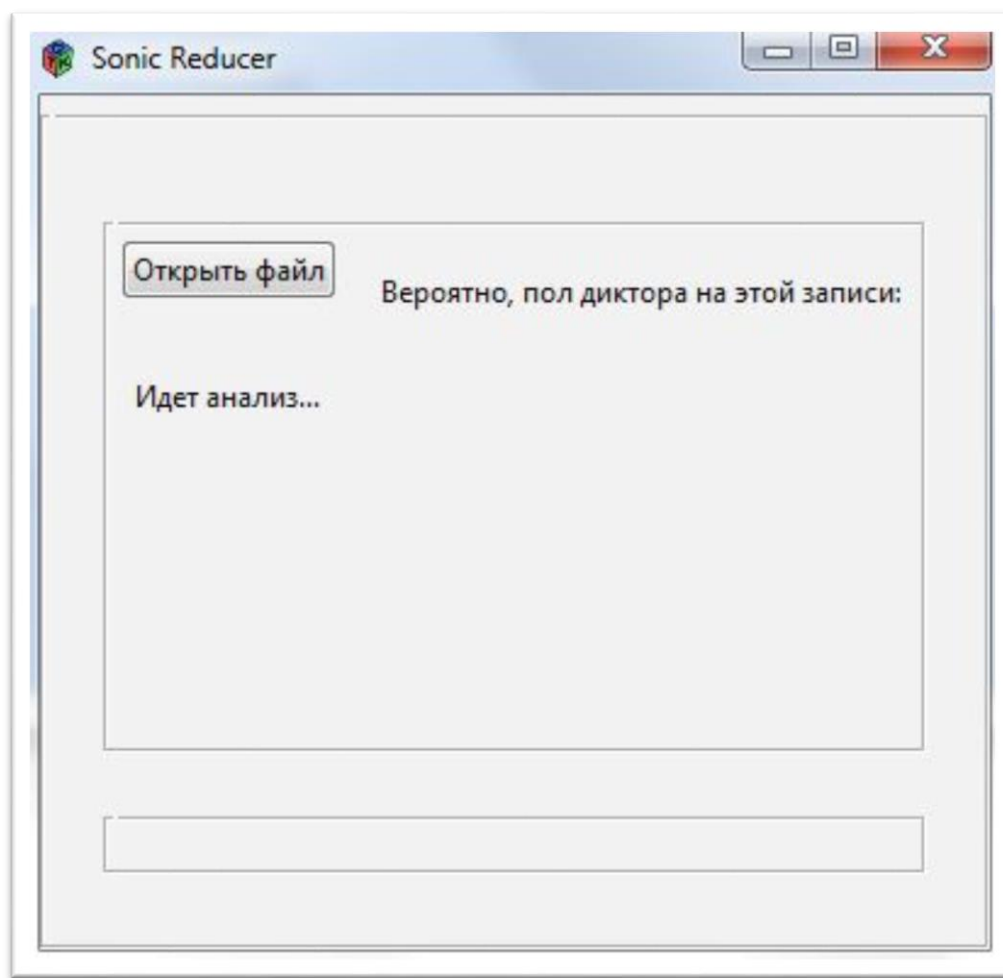


Рисунок 9 – Интерфейс системы

Рассмотрим порядок, в котором пользователь будет производить работу с системой. Для получения результата классификации необходимо:

- 1) открыть приложение;
- 2) нажать кнопку «Открыть файл»;
- 3) выбрать файл нужного формата ;
- 4) нажать "Ok";
- 5) проверить результат классификации, отображённый в окне программы;
- 6) выбрать другой файл для тестирования или закрыть программу.

Перв. примен.																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
---------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

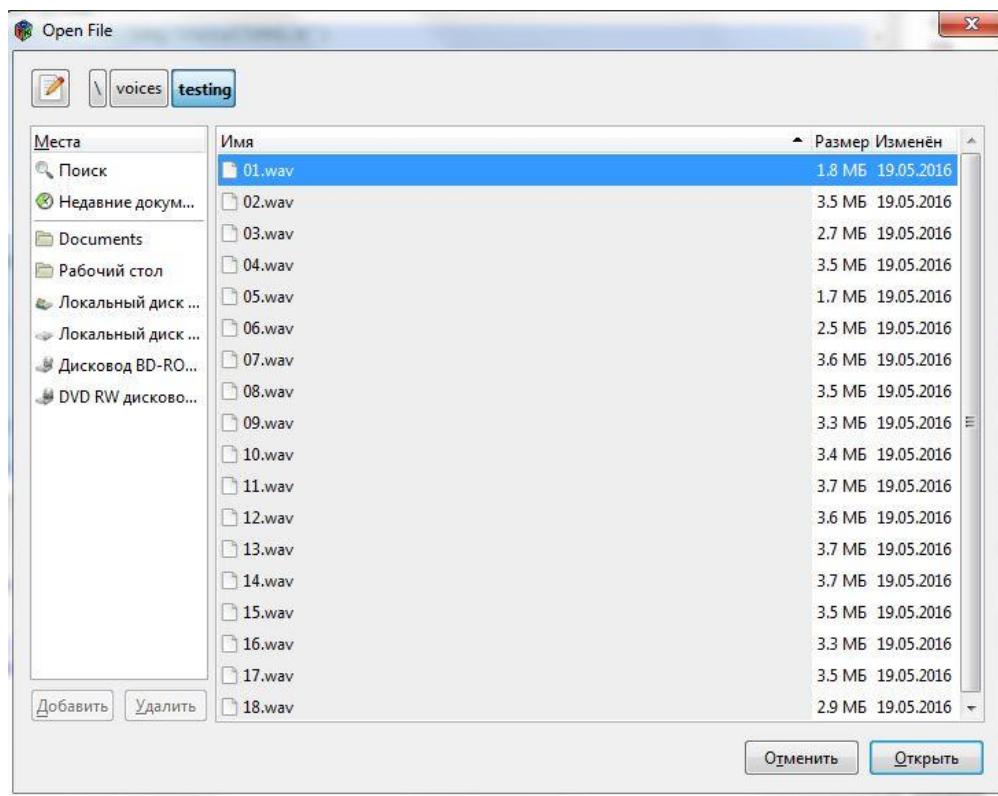


Рисунок 10 – Выбор файла для тестирования

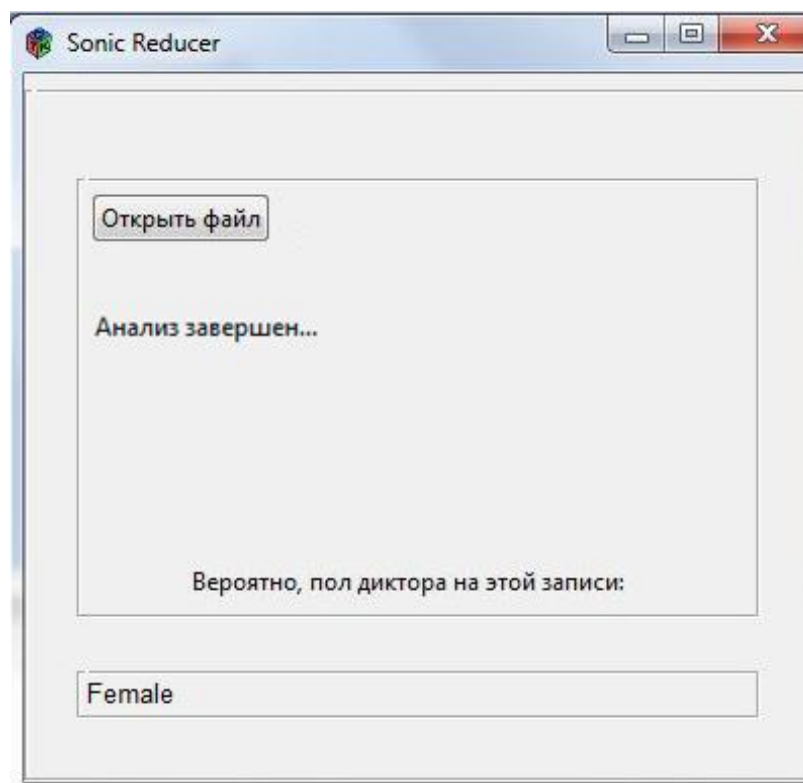


Рисунок 11 – Результат работы системы

Как видно из рисунка, приведённого выше, система работает корректно.

Перв. примен.	<p align="center">Список литературы</p> <p>1. "Информационный подход к описанию звукового сигнала" - Гай В.Е. -Труды МФТИ, Том 6, № 2, 2014, С. 167-173.</p> <p>2. "Оценка эмоционального состояния человека по голосу с позиций теории активного восприятия" - В. Е. Гай, В. А. Утробин, П. А. Родионов, М. О. Дербасов - Системы управления и информационные технологии, №1.1(59), 2015, С. 118-122.</p> <p>3. Машина опорных векторов - http://www.machinelearning.ru/wiki/index.php?title=SVM</p> <p>4. Метод ближайших соседей - http://www.machinelearning.ru/wiki/index.php?title=KNN</p> <p>5. "Статистический анализ и визуализация данных с помощью R" - Мاستицкий С.Э., Шитиков В.К. - "ДМК Пресс", Москва, 2015 г.</p> <p>6. "R в действии. Анализ и визуализация данных на языке R." - Кабаков Р. - "ДМК Пресс", Москва, 2013 г.</p> <p>7. "Beginning R: The Statistical Programming Language" –Gardener M. – 2012 г.</p>				
Подпись и дата	Инв. № дубл.	Взам. инв. №	Подпись и дата	Инв. № подл.	<div> <div>ВКР-НГТУ-09.03.01-(12 В-2)-002-2016(ПЗ)</div> <div> <div>Изм.</div> <div>Лист</div> <div>№ докум.</div> <div>Подпись</div> <div>Дата</div> </div> </div>
					<div>Лист</div> <div>32</div>