## Лабораторная работа №1. Предварительная обработка речевой информации

*Цель работы:* получить практические навыки обработки речевой информации с помощью языка python. Научиться загружать данные, рассчитывать кратковременную энергию речевого сигнала, выявлять участки голосовой активности, удалять шум.

*Задание:*

1. Изучить возможности библиотек wave, soundfile по загрузке и воспроизведению аудиоданных в формате wav. Изучить примеры работы с аудиоданными в файлах dataprocessing.py, recording.py.
2. Написать программу, выполняющую
   1. загрузку речевой информации из файла формата wav;
   2. разбиение речевого сигнала на кадры с настраиваемой степенью перекрытия;
   3. расчет кратковременной энергии в каждом кадре;
   4. выявление участков голосовой активности;
   5. Построение графиков речевого сигнала и участков голосовой активности.
3. В соответствии с номером варианта загрузить зашумленные данные и выполнить шумоочистку, используя
   1. метод вычитания спектров;
   2. низкочастотный фильтр.

Сравнить результаты и оценить качество удаления шума.

1. Написать программу выполняющую запись с микрофона звуковых данных в файл формата wav с частотой дискретизации 22050 Гц, количеством каналов – 1, количеством бит на отсчет – 16.
2. Для словаря голосовых команд, выбранного из таблицы № в соответствии с вариантом, записать не менее десяти реализаций каждой голосовой команды, выполнив ручное или автоматическое удаление участков молчания с помощью ранее реализованного алгоритма выявления участков голосовой активности.
3. Для выбранных двух слов из словаря произвести сегментацию голосовых команд на фонемы и построить графики нескольких вокализованных и невокализованных сегментов.

Варианты словарей голосовых команд к лабораторной работе приведены в таблице 2.

Таблица 2

**Варианты заданий к лабораторной работе №1**

|  |  |
| --- | --- |
| **Вариант №** | **Список команд** |
| 1 | Ноль, один, два, три, четыре, пять, шесть, семь, восемь, девять, плюс |
| *2* | Вперед, назад, налево, направо, стоп, разворот, вверх, вниз, увеличить, уменьшить, сканировать, осмотреться |
| *3* | Копировать, вставить, вырезать, удалить, увеличить, уменьшить, найти, заменить, открыть, закрыть, сохранить |
| *4* | Блокнот, консоль, Word, Excel, проводник, Chrome, Firefox, календарь, putty, paint, калькулятор |
| *5* | Книги, фильмы, музыка, фотографии, документы, проекты, дистрибутивы, загрузки, игры |
| *6* | Сложить, вычесть, умножить, разделить, корень, степень, равно, логарифм, факториал, синус, косинус, тангенс |
| *7* | Карта, погода, энциклопедия, почта, новости, переводчик, электрички, расписание, напоминания, задачи |

**Контрольные вопросы**

1. Расскажите про особенности хранения аудиоданных в формате wav. Какие еще существуют форматы хранения аудиоданных? В чем их отличие от формата wav?
2. Назовите особенности речевых сигналов. Как эти особенности учитываются при обработке данных?
3. Расскажите про алгоритмы выявления участков голосовой активности. Зачем они нужны?
4. Как осуществляется удаление шума с помощью метода спектральных вычитаний? Назовите его достоинства и недостатки.
5. Можно ли удалять шум с помощью низкочастотного фильтра? Назовите другие методы удаления шума и поясните их принцип работы.
6. Расскажите про обработку речевой информации с помощью библиотек языка python. Какие функции и методы в них реализованы, в чем их особенности?
7. В чем отличие вокализованных от невокализованных участков речевого сигнала?

## Лабораторная работа №2. Выявление информативных признаков и построение системы распознавания голосовых команд

*Цель работы:* получить навыки расчета информативных признаков речевой информации, построения простой системы распознавания голосовых команд и оценки качества работы системы.

*Задание:*

1. Выполнить расчет информативных признаков для реализаций голосовых команд, записанных в лабораторной работе №1.
2. Построить графики рассчитанных информативных признаков:
   1. Для сравнения реализаций одинаковых голосовых команд
   2. Для сравнения реализаций разных голосовых команд
   3. Для сравнения вокализованных и невокализованных участков голосовых команд, найденных в п.6 лабораторной работы №1
3. Изучить работу алгоритмов нескольких методов распознавания (метод динамической трансформации шкалы времени, многослойный персептрон) на простых данных.
4. Сформировать обучающие и тестовые выборки
5. Применить разработанные алгоритмы для распознавания обучающих и тестовых выборок голосовых команд, оценить качество распознавания (быстродействие и точность), проанализировать полученные результаты.
6. Повторить п.5. при использовании различных значений степеней перекрытия кадров, разных окнах взвешивания, при разном числе информативных признаков, разной структуре нейронной сети. Полученные результаты свести в единую таблицу.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | *Параметры предварительной обработки данных* | | | *Точность* | | *Быстродействие в миллисекундах* | |
| № | Степень перекрытия | Окно | Кол-во инф. признаков | DTW | NN | DTW | NN |
| 1 |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |
| … |  |  |  |  |  |  |  |
| 20 |  |  |  |  |  |  |  |

1. Построить графики полученных в предыдущем пункте результатов и сделать выводы о качестве распознавания голосовых команд при использовании различных параметров предварительной обработки данных и методах распознавания.

**Контрольные вопросы**

1. Приведите примеры информативных признаков речевых сигналов. Как они рассчитываются?
2. Как рассчитываются мел-частотные кепстральные коэффициенты?
3. Как рассчитываются форманты?
4. Объясните назначение и принцип работы метода динамической трансформации шкалы времени. В чем его недостатки?
5. Как осуществляется распознавание голосовых команд с помощью многослойного персептрона?
6. Как формируются обучающие и тестовый выборки для решения задачи распознавания голосовых команд?
7. Как оценивается качество распознавания голосовых команд?
8. Как влияют параметры предварительной обработки данных на качество распознавания?

## Лабораторная работа №3. Распознавание автомобильных номеров на основе применения контурного анализа

*Цель работы:* получить практические навыки обработки и анализа изображений, решив задачу классификации контуров символов автомобильных номеров.

*Задание:*

1. Изучить функциональные возможности библиотек opencv для предварительной обработки изображений.
2. Написать программу, осуществляющую:
   1. загрузку и визуализацию цветного изображения эталонных символов автомобильных регистрационных знаков;
   2. сглаживание изображения;
   3. бинаризацию изображения;
   4. нахождение и визуализацию контуров символов;
3. Написать функции, осуществляющие:
   1. кодирование контуров в последовательность комплексных чисел;
   2. нахождение скалярного произведения контуров;
   3. нахождение длины контура;
   4. расчет взаимной корреляционной функции двух контуров;
   5. эквализацию контура;
4. Написать программу выполняющую
5. загрузку тестового изображения автомобиля с регистрационными знаками;
6. обнаружение и выделение контуров символов автомобильного номера;
7. распознавание(классификацию) найденных символов автомобильного номера;
8. Проанализировать полученные результаты, сделать выводы о качестве обнаружения, распознавания номеров и временных затрат на всех этапах обработки изображений.

**Контрольные вопросы**

1. Этапы предварительной обработки изображений
2. Поясните суть и этапы контурного анализа
3. Как осуществляется поиск контура?
4. Как кодируются контуры?
5. Как осуществляется сравнение контуров?
6. Поясните принцип работы медианного фильтра. В каких ситуациях его применение будет эффективным, а в каких нет?

## Лабораторная работа №4. Распознавание лиц на фотографиях

*Цель работы:* получить практические навыки решения задачи распознавания лиц на фотографиях. Научиться обнаруживать лица на фотографиях, оценивать качество обнаружения, выявлять характерные черты и проводить классификацию.

*Задание:*

1. Изучить функциональные возможности библиотеки opencv для обнаружения объектов на изображениях.
2. Сформировать выборки из коллекций фотографий с конференций и выставок.
3. Выполнить поиск лиц на фотографиях с помощью настроенного каскадного классификатора использующего признаки Хаара, оценить качество обнаружения лиц.
4. Написать программу, которая с помощью детектора на основе признаков Хаара сформирует набор изображений лиц. Сформировать обучающие и тестовые выборки для распознавания лиц.
5. Написать программу, выполняющую
6. Создание сверточной нейронной сети;
7. Загрузку обучающих и тестовых выборок;
8. Обучение сверточной нейронной сети на обучающей выборке;
9. Оценку качества распознавания лиц на тестовой выборке, построение графиков изменения ошибки и точности классификации в процессе обучения.
10. Согласно указаниям преподавателя сформировать обучающую и тестовую выборку для обучения каскадного классификатора использующего признаки Хаара, провести его обучение и оценить качество обнаружения заданных объектов на тестовой выборке.

**Контрольные вопросы**

1. Методы обнаружения лиц на фотографиях. Как оценивается качество обнаружения?
2. Методы распознавания лиц
3. Как решается задача распознавания лиц на фотографиях с помощью сверточной нейронной сети?
4. Для решения каких задач может использоваться сверточная нейронная сеть? Каковы её достоинства и недостатки?
5. Как настраивается каскадный классификатор, использующий признаки Хаара?

## Методические указания

Лабораторные работы выполняются с использованием языка python версии не ниже 3.6. Для написания и отладки программ рекомендуется использовать среду PyCharm. В процессе выполнения лабораторной работы необходимо фиксировать в текстовом редакторе полученные результаты: код программы, графики, выбранные числовые значения параметров

Для выполнения *лабораторной работы №1* потребуются функциональные возможности следующих библиотек языка python:

* *wave* (<https://docs.python.org/3.6/library/wave.html>) – предоставляет функции загрузки и сохранения аудиоданных в формате wav
* *winsound* (<https://docs.python.org/3.6/library/winsound.html>) – предоставляет функции воспроизведения аудиоданных в операционной системе Windows
* *pyaudio* (<http://people.csail.mit.edu/hubert/pyaudio/#docs>) – предоставляет возможности записи аудиоданных
* *NumPy (*[*http://www.numpy.org*](http://www.numpy.org)*) –* позволяет работать с многомерными массивами и высокоуровневыми математическими функциями.
* *matplotlib* и ее модуль *pyplot (*[*https://matplotlib.org/users/pyplot\_tutorial*](https://matplotlib.org/users/pyplot_tutorial)*)* – содержит богатые возможности для построения графиков
* *scipy* и её модуль *signal* (<https://docs.scipy.org/doc/scipy/reference/signal.html#module-scipy.signal>) – содержит множество функций для обработки и фильтрации сигналов

Подключение необходимых библиотек и модулей выполняется следующим фрагментом программы:

import wave

import winsound

import matplotlib.pyplot as plt

import numpy as np

import pyaudio as py

from scipy.signal import butter, filtfilt, welch

Для извлечения из речевого сигнала информативных для распознавания признаков и других алгоритмов обработки речевые сигналы разбивают на кадры небольшой длительности (20-30 мс), в пределах которых сигнал можно считать условно стационарным. Кадры рекомендуется брать с перекрытием в 50-80%.

Перед построением низкочастотного фильтра необходимо оценить спектральную плотность мощности речевого сигнала, построить его график и выбрать частоту среза. Построение оценки спектральной плотности мощности и её графика можно выполнить, например, с помощью следующего кода:

Fig1 = plt.figure()

F\_welch, S\_welch = welch(y, fs=Fs, nperseq=480, noverlap = 240, detrend=None,scaling=’density’, window=’hann’)

plt.plot(F\_welch, S\_welch)

plt.show()

Запись каждой голосовой команды в пункте 11 рекомендуется выполнять в отдельный файл, записывая сразу несколько реализаций внутри файла. Затем файлы с голосовыми командами следует разбить на отдельные файлы с помощью реализованного алгоритма выявления участков голосовой активности, предварительно настроив порог на участках молчания в начале файла.

В *лабораторной работе № 2* потребуются функции из библиотек:

* librosa (<http://librosa.github.io/librosa/>), позволяющей выполнять расчет мел-частотных кепстральных коэффициентов, числа нулей интенсивности, сравнивать последовательности отсчетов разной длины с помощью алгоритма динамической трансформации шкалы времени;
* scikit-learn (<https://scikit-learn.org/stable/>), собравшей в себе множество методов и алгоритмов машинного обучения;
* tensorflow (<https://www.tensorflow.org/overview>), предназначенной для решения задач построения и обучения нейронных сетей

Включение необходимых для выполнения задания модулей и библиотек выполняется следующим фрагментом программы:

import librosa.core as core

import librosa.feature as ftr

import librosa.sequence as seq

import soundfile as sf

import numpy as np

import matplotlib.pyplot as plt

import os

import random as rnd

from sklearn.preprocessing import LabelBinarizer

from sklearn.model\_selection import train\_test\_split

from tensorflow.keras import layers

from tensorflow.keras import Sequential

from tensorflow.keras.layers import Conv1D, Dense, Dropout

import tensorflow as tf

В качестве информативных признаков рекомендуется воспользоваться мел-частотными кепстральными коэффициентами. Для их расчета необходимо последовательно открывать файлы с записанными в предыдущей лабораторной работе реализациями голосовых команд и с помощью функции mfcc модуля librosa.feature выполнять их расчет и сохранение в массив.

Изучение алгоритмов метода динамической трансформации шкалы времени и обучения многослойного персептрона рекомендуется выполнять на простых примерах. Например, сравнить два произвольных вектора разной размерности методом динамической трансформации шкалы времени и обучить многослойный персептрон решать задачу исключающего ИЛИ (XOR).

Обучающие и тестовые выборки для распознавания голосовых команд необходимо сформировать из массива рассчитанных мел-частотных кепстральных коэффициентов случайным образом, обеспечив соотношение объемов обучающей и тестовой выборок 4:1.

Пункты 5 и 6 лабораторной работы №2 следует выполнять, проводя эксперименты с различными степенями перекрытия кадров (40%, 50%,70%,80%,90%), с разными окнами взвешивания (окна хэмминга, ханна, бартлета), с разным числом информативных признаков в кадре (9,13,18) и разными архитектурами многослойного персептрона (менять число внутренних слоев и нейронов в них, использовать сигмоидальную функцию активации). В ходе экспериментов необходимо фиксировать затраченное время на распознавание тестовой выборки и точность.

*Лабораторная работа № 3* выполняется с использованием библиотеки алгоритмов компьютерного зрения и обработки изображений *opencv* (<https://docs.opencv.org/master/>) импортируемая в программу с помощью инструкции

import cv2 as cv

Перед выполнением основных пунктов задания необходимо изучить функциональные возможности библиотеки *opencv*, обратив внимание на следующий список функций:

*imread, imshow, waitkey, destroyAllWindows, blur, cvtColor, inRange, findContours, contourArea, boundingRect, drawContours, putText, morphologyEx, createTrackBar, namedWindow, getTrackBarPos*.

Программа в примере labwork3\_example.py поможет выбрать пороговые значения для бинаризации изображений.

Пункты 3 и 4 задания выполняются в соответствии с теоретическим материалом, посвященным контурному анализу.

*Лабораторная работа № 4* выполняется с использованием библиотек NumPy, Scikit-learn, tensorflow, opencv.

Для изучения возможностей библиотеки *opencv* по обнаружению объектов на изображениях необходимо ознакомиться с документацией к классу CascadeClassifier и примером labwork4\_example.py, в котором осуществляется поиск лиц и глаз на фотографии с использованием обученного каскадного классификатора на основе признаков Хаара.

В пункте 2 и 3 задания необходимо из коллекции фотографий с конференций и выставок отобрать 50 фотографий с участниками и без, выполнить поиск лиц с помощью каскадного классификатора и оценить количество ложных и положительных срабатываний детектора. Для выполнения пункта 4 подготовить не менее 300 фотографий с участниками конференций и выставок.

В пункте 4 задания необходимо написать программу, выполняющую поиск лиц на фотографиях и сохранение найденной прямоугольной области в отдельный файл. При сохранении изображений лиц в отдельные файлы нужно обеспечить их приведение к единой размерности. Среди созданных изображений необходимо найти изображения лиц наиболее часто встречающихся участников и сгруппировать их в отдельные папки. Далее с помощью программы необходимо загрузить все изображения в единый массив и выполнить его разбиение на обучающую и тестовую выборку с соотношением объемов – 4:1.

В п.5 необходимо написать программу создания и обучения сверточной нейронной сети для распознавания лиц сформированных в п.4. выборок. Количество нейронов в выходном слое нейронной сети должно соответствовать количеству классов в обучающей выборке, а количество нейронов на входе определяется размером изображений. Количество внутренних слоев и нейронов задается студентом самостоятельно. Обученная сверточная нейронная сеть должна обеспечивать точность распознавания лиц не менее 75%.