
PDF

Выпуск 1.0.0-a9

OpenAV

мая 11, 2024

Содержание:

1	Установка и обновление	3
2	Модули	5
3	API	37
4	Диаграмма классов	63
5	Набор аудиовизуальных данных	65
6	Команда разработчиков	67
7	Ответы на часто задаваемые вопросы	73
8	Изменения версий	75
	Содержание модулей Python	77
	Алфавитный указатель	79

OpenAV - библиотека с открытым исходным кодом, предназначена для решения задач автоматического распознавания речевых команд на основе интеллектуального анализа аудиовизуальных данных.

Установка и обновление

1.1 Установка с помощью PyPi

```
pip install openav
```

1.2 Обновление с помощью PyPi

```
pip install --upgrade openav
```

1.3 Зависимости

Таблица 1: Устанавливаются автоматически

Библиотека	Рекомендуемая версия	Текущая версия
ipython	8.10.0	
colorama	0.4.6	
numpy	1.24.2	
pandas	1.5.3	
prettytable	3.6.0	
torch	1.13.1	
torchaudio	0.13.1	
torchvision	0.14.1	
vosk	0.3.44	
av	10.0.0	
filetype	1.2.0	
requests	2.28.2	
pyyaml	6.0	
streamlit	1.20.0	
watchdog	2.3.1	
pymediainfo	6.0.1	
pillow	9.5.0	
imgaug	0.4.0	
flask	2.3.3	
ffmpeg	1.4	
librosa	0.10.1	
matplotlib	3.6.3	
mediapipe	0.9.3.0	
opencv_contrib_python	4.9.0.80	
einops	0.7.0	
lion_pytorch	0.1.4	
scikit-learn	1.4.2	
tqdm	4.66.2	

2.1 Модуль записи речевых аудиовизуальных данных

Модуль предназначен для записи аудиовизуальных данных с их последующей сортировкой по заданному словарю.

Команда для запуска модуля записи данных:

```
python ./openav/api/recorder_app.py --config <путь_к_вашему_конфигурационному_файлу>.yaml
```

После выполнения команды запускается локальный сервер, на котором разворачивается приложение для записи данных. Для того чтобы начать использование графического интерфейса модуля перейдите по адресу в браузере <http://127.0.0.1:5000>

2.1.1 Конфигурационный файл

В конфигурационном файле определены следующие функции, с помощью которых выполняется поиск доступных устройств и определение их технических характеристик для записи аудиовизуальных данных.

Функция `find_my_devices (camera = True, micro = True)` определяет и возвращает название устройств, которые подключены в системе. Передаваемые параметры `camera` и `micro` по умолчанию имеют значение `True`. Если вам не требуется использование камеры или микрофона, то необходимо передать значение `False`.

Пример использования данной функции. Переменным `video` и `audio` присваиваются значения, являющиеся результатом выполнения функции.

```
video, audio = find_my_devices(True, True)
```

Значения, возвращаемые из функции `find_my_devices()` необходимы для определения параметров изображения и звука, которые поддерживает найденное в системе устройство.

Функция `get_available_parameters (device)` принимает название устройства `<str> device` и возвращает список параметров, которые поддерживает указанное устройство. В случае, если передается название вебкамеры, то возвращаемый список будет содержать все доступные параметры разрешения изображения и максимальное количество кадров, соответствующее этому разрешению.

Пример использования данной функции. Переменной `available_camera_params` присваивается список доступных параметров.

```
available_camera_params = get_available_parameters (video)
```

Функция `get_camera_params (dict, prefer = 'max')` возвращает выбранные параметры из полного списка доступных параметров вебкамеры. Обязательным принимаемым на вход аргументом является список `dict`, который был получен с помощью предыдущей функции. Аргумент `prefer = 'max'` по умолчанию установлен на получение максимально допустимых параметров устройства. Данное значение передается в формате `640x480`, либо `max` или `min`. Результатом выполнения функции является возврат значений `available_size`, `fps`.

Пример использования данной функции. Переменным `available_res` и `available_fps` присваиваются значения, полученные в ходе выполнения функции `get_camera_params`.

```
available_res, available_fps = get_camera_params(available_params, '640x480')
```

В конфигурационном файле также необходимо указать словарь, в соответствии с которым будет выполняться запись данных. Словарь имеет вид `key-phrase`, где `key` выступает в качестве порядкового номера фразы в словаре.

Пример словаря

```
dict = [
    {'key': 0, 'phrase': 'Левая'},
    {'key': 1, 'phrase': 'Правая'},
    {'key': 2, 'phrase': 'Нажать левую'},
    {'key': 3, 'phrase': 'Отпустить левую'},
    {'key': 4, 'phrase': 'Нажать правую'},
    {'key': 5, 'phrase': 'Отпустить правую'},
    {'key': 6, 'phrase': 'Нажать левую'},
    {'key': 7, 'phrase': 'Отпустить левую'},
    {'key': 8, 'phrase': 'Нажать правую'},
    {'key': 9, 'phrase': 'Отпустить правую'},
    {'key': 10, 'phrase': 'Нажать левую'},
    {'key': 11, 'phrase': 'Отпустить левую'},
    {'key': 12, 'phrase': 'Нажать правую'},
    {'key': 13, 'phrase': 'Отпустить правую'},
    {'key': 14, 'phrase': 'Нажать левую'},
    {'key': 15, 'phrase': 'Отпустить левую'},
    {'key': 16, 'phrase': 'Нажать правую'},
    {'key': 17, 'phrase': 'Отпустить правую'},
    {'key': 18, 'phrase': 'Нажать левую'},
    {'key': 19, 'phrase': 'Отпустить левую'},
    {'key': 20, 'phrase': 'Нажать правую'},
    {'key': 21, 'phrase': 'Отпустить правую'},
    {'key': 22, 'phrase': 'Нажать левую'},
    {'key': 23, 'phrase': 'Отпустить левую'},
    {'key': 24, 'phrase': 'Нажать правую'},
    {'key': 25, 'phrase': 'Отпустить правую'},
    {'key': 26, 'phrase': 'Завершить'}
]
```

2.1.2 Графический интерфейс модуля записи

Интерфейс модуля для состоит из двух частей:

- В левой части представлено изображение, получаемое с подключенной камеры и кнопка начать запись (Rec)
- В правой части располагается блок управление словарем. В этом блоке отображаются следующие элементы:
 - Цифра в круге означает порядковый номер фразы в словаре
 - Фраза из словаря
 - Кнопки переключения элементов словаря `prev` и `next`, после нажатия на которые отображаются предыдущую или следующую фразу соответственно

Нажатие на кнопку записи (Rec) начинает процесс записи данных. После нажатия на кнопку, она изменяет свой цвет, что означает об успешном запуске записи.

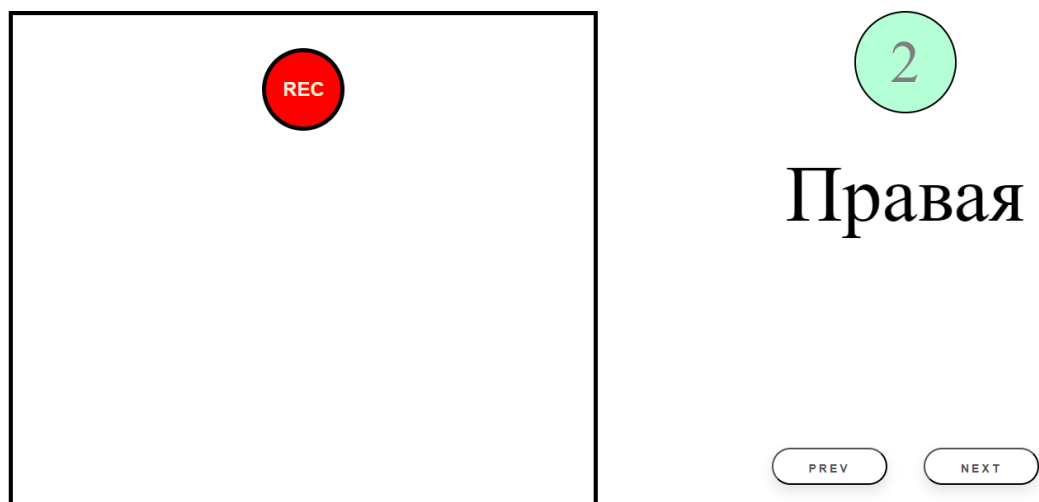


Рис. 1: Интерфейс модуля для записи данных



Рис. 2: Активное состояние кнопки записи

Повторное нажатие на кнопку записи завершает процесс. Записанный файл сохраняется по пути, указанному в конфигурационном файле. Изменить словарь словарь также можно в конфигурационном файле.

2.2 Модуль загрузки данных

Данный модуль позволяет выполнять поиск, проверку и загрузку данных. Модуль предусматривает вложенность директорий.

2.2.1 Конфигурационный файл

Конфигурационный файл включает в себя следующие настройки:

- Отображение процесса выполнения программы в терминале (командной строке)
- Работа с файловой системой

2.2.2 Отображение процесса выполнения команды в терминале

Параметр	Тип	Значение по умолчанию	Описание
hide_metadata	bool	false	Включение отображения метаданных
hide_libs_vers	bool	false	Включение отображения версий установленных библиотек в командной строке

2.2.3 Работа с файловой системой

Параметр	Тип	Значение по умолчанию	Описание
path_to_dataset	str	<путь_к_исходным_данным>	Директория, где находятся данные, которые необходимо загрузить или проверить
depth	int	1	Глубина иерархии для получения данных. Указывается количество подкаталогов в директории path_to_dataset
ext_search_files	list	["mov", "mp4", "webm", "wav"]	Список расширений файлов, которые будут обрабатываться. Указывать можно как для видео, так и для аудио

2.3 Модуль детектирования речевой активности

2.3.1 Детектирование речевой активности (Silero VAD)

Команда для запуска детектирования речевой активности в аудиовизуальном сигнале:

```
openav_vad --config <путь_к_вашему_конфигурационному_файлу>.yaml
```

Важно: Для запуска команды необходимо обязательно указать путь к конфигурационному файлу. Запускать программу необходимо из директории, где она расположена.

Конфигурационный файл

Конфигурационный файл включает в себя следующие настройки:

- Отображение процесса выполнения программы в терминале (командной строке)
- Работа с файловой системой
- Параметры детектора голосовой активности (Silero VAD)
- Параметры кодирования выходного файла

Пример конфигурационного файла

Отображение процесса выполнения команды в терминале

Параметр	Тип	Значение по умолчанию	Описание
hide_metadata	bool	false	Включение отображения метаданных
hide_libs_vers	bool	false	Включение отображения версий установленных библиотек в командной строке

Работа с файловой системой

Параметр	Тип	Значение по умолчанию	Описание
path_to_save_model	str	<./models>	Директория, где будут размещаться скачанные модели, в данном случае модель для работоспособности VAD
path_to_dataset	str	<путь_к_исходным_данным>	Директория, где находятся данные, которые необходимо обработать VAD
path_to_dataset_vad	str	<./dataset_vad>	Директория, куда сохраняются фрагменты аудиовизуального сигнала после обработки VAD
dir_va_names	dict	{"video": "Video", "audio": "Audio"}	Директории для сохранения видео и аудио файлов. Названия директорий могут быть произвольными
force_reload	bool	false	Включение принудительной загрузки модели VAD из сети
clear_dirvad	bool	true	Очистка директории, в которую сохраняются фрагменты аудиовизуального сигнала
depth	int	1	Глубина иерархии для получения данных. Указывается количество подкаталогов в директории path_to_dataset
ext_search_files	list	["mov", "mp4", "webm", "wav"]	Список расширений файлов, которые будут обрабатываться. Указывать можно как для видео, так и для аудио

Параметры детектора голосовой активности (Silero VAD)

Параметр	Тип	Значение по умолчанию	Описание
sampling_rate	int	16000	Частота дискретизации. На текущий момент поддерживаются частоты: 8000 и 16000
threshold	float	0.5	Порог вероятности речи (от 0.0 до 1.0). VAD выводит вероятности речи для каждого звукового фрагмента, вероятности выше установленного значения считаются речью. Параметр необходимо настраивать индивидуально в зависимости от набора данных. Например для шумных условий параметр рекомендуется устанавливать в значение от 0.7 до 0.95. В условиях низкого уровня шума или его отсутствия, параметр лучше устанавливать на низкие значения 0.1 — 0.25
min_speech_duration_ms	float	250	Минимальная длительность речевого фрагмента. Рекомендуется устанавливать в пределах от 750 мс до 1000 мс. Настройка также является индивидуальной, которую необходимо подбирать в зависимости от набора данных
min_silence_duration_ms	float	100	Минимальная длительность тишины в выборках между отдельными речевыми фрагментами, прежде чем разделить его. Рекомендуется устанавливать в пределах 500 мс, для того, чтобы не разделялись предложения. Однако, можно настроить индивидуально, под свои условия
window_size_samples	int	1536	Количество выборок в каждом окне. Предупреждение! Модели VAD были обучены с использованием выборок 512, 1024, 1536 для частоты дискретизации 16000 и 256, 512, 768 для частоты дискретизации 8000. Настоятельно рекомендуется использовать эти значения, изменение значений может повлиять на производительность модели
speech_pad_ms	float	30	Внутренние отступы для итоговых речевых фрагментов. Рекомендуется использовать значение 250 мс — 400 мс чтобы избежать обрезания фрагментов речи. Настройка также является индивидуальной и устанавливается в соответствии с условиями пользователя

Параметры кодирования выходного файла

Параметр	Тип	Значение по умолчанию	Описание
type_encode	str	crf	Типы кодирования. Доступные варианты: ['qscale', 'crf']
crf_value	int	23	Качество кодирования (от 0 до 51. Чем ниже значение, тем лучше качество и наоборот. Стоит учитывать, что изменения качества кодирования влияет на скорость обработки
presets_crf_encode	str	medium	Скорость кодирования и сжатия. Доступные варианты: ['ultrafast', 'superfast', 'veryfast', 'faster', 'fast', 'medium', 'slow', 'slower', 'veryslow']. Изменения параметра влияет на скорость кодирования и степень сжатия
sr_input_type	str	audio	Типы файлов для распознавания речи. Доступные варианты: ['audio', 'video']

Демонстрация работы детектора речевой активности

Ниже продемонстрированы результаты работы детектора речевой активности как для входного аудиовизуального сигнала, так и для акустического.

Аудиовизуальный сигнал

Акустический сигнал

Пример видеозаписи до обработки детектором речевой активности

Видеозапись до обработки VAD

Примеры видеозаписей после обработки детектором речевой активности

Фраза 1

Фраза 2

Фраза 3

Фраза 4

Фраза 5

Пример аудиозаписи до обработки детектором речевой активности

Примеры аудиозаписей после обработки детектором речевой активности

2.3.2 Детектирование речевой активности (Vosk)

Команда для запуска детектирования речевой активности в аудиовизуальном сигнале:

```
openav_vosk_sr --config <путь_к_вашему_конфигурационному_файлу>.yaml
```

Важно: Для запуска команды необходимо обязательно указать путь к конфигурационному файлу. Запускать программу необходимо из директории, где она расположена.

Конфигурационный файл

Конфигурационный файл включает в себя следующие настройки:

- Отображение процесса выполнения программы в терминале (командной строке)
- Работа с файловой системой
- Параметры детектора голосовой активности (Vosk)
- Параметры кодирования выходного файла

Пример конфигурационного файла

Отображение процесса выполнения команды в терминале

Параметр	Тип	Значение по умолчанию	Описание
hide_metadata	bool	false	Включение отображения метаданных
hide_libs_vers	bool	false	Включение отображения версий установленных библиотек в командной строке

Работа с файловой системой

Параметр	Тип	Значение по умолчанию	Описание
path_to_save_model	str	<./models>	Директория, где будут размещаться скачанные модели, в данном случае модель для работоспособности VAD
path_to_dataset	str	<путь_к_исходным_данным>	Директория, где находятся данные, которые необходимо обработать
path_to_dataset_vosk	str	<./dataset_vosk>	Директория, куда сохраняются фрагменты аудиовизуального сигнала после обработки
dir_va_names	dict	{"video": "Video", "audio": "Audio"}	Директории для сохранения видео и аудио файлов. Названия директорий могут быть произвольными
force_reload	bool	false	Включение принудительной загрузки модели Vosk из сети
folder_name_unzip	str	<название_папки>	Название папки, в которую будет извлекаться модель Vosk
clear_dirvosk_sr	bool	true	Очистка директории, в которую сохраняются фрагменты аудиовизуального сигнала
depth	int	1	Глубина иерархии для получения данных. Указывается количество подкаталогов в директории path_to_dataset
ext_search_files	list	["mov", "mp4", "webm", "wav"]	Список расширений файлов, которые будут обрабатываться. Указывать можно как для видео, так и для аудио

Параметры детектора голосовой активности (Vosk)

Параметр	Тип	Значение по умолчанию	Описание
sampling_rate	int	16000	Частота дискретизации. На текущий момент поддерживаются частоты: 8000 и 16000
speech_left_pad_ms	float	0	Внутренний отступ до начала речевого фрагмента. Настройка поможет избавиться от лишней тишины в начале обработанного фрагмента
speech_right_pad_ms	float	300	Внутренний отступ в конце речевого фрагмента. Настройка поможет избавиться от лишней тишины в конце обработанного фрагмента, либо наоборот увеличить длительность речевого фрагмента, на случай, если фраза незначительно обрезается после обработки
lang_model	str	ru	Выбор языка, на котором необходимо обработать данные. Поддерживаемые языки: ru и en
dict_size	str	big	Размер словаря, на котором была обучена модель для распознавания. big и small. Влияет на качество работы модели, однако стоит учитывать, что малоресурсная модель будет работать быстрее, чем большая.

В текущей версии для русского языка используются модели `vosk-model-ru-0.42` и `vosk-model-small-ru-0.22`, обученные на большом и малом словарях соответственно. Более подробное описание моделей представлено по ссылке .

Параметры кодирования выходного файла

Параметр	Тип	Значение по умолчанию	Описание
type_encode	str	crf	Типы кодирования. Доступные варианты: ['qscale', 'crf']
crf_value	int	23	Качество кодирования (от 0 до 51. Чем ниже значение, тем лучше качество и наоборот. Стоит учитывать, что изменения качества кодирования влияет на скорость обработки
presets_crf_encode	str	medium	Скорость кодирования и сжатия. Доступные варианты: ['ultrafast', 'superfast', 'veryfast', 'faster', 'fast', 'medium', 'slow', 'slower', 'veryslow']. Изменения параметра влияет на скорость кодирования и степень сжатия
sr_input_type	str	audio	Типы файлов для распознавания речи. Доступные варианты: ['audio', 'video']

Демонстрация работы детектора речевой активности

Ниже продемонстрированы результаты работы детектора речевой активности как для входного аудио-визуального сигнала, так и для акустического.

Аудиовизуальный сигнал

Акустический сигнал

Пример видеозаписи до обработки детектором речевой активности

Видеозапись до обработки VAD

Примеры видеозаписей после обработки детектором речевой активности

Фраза 1

Фраза 2

Фраза 3

Фраза 4

Фраза 5

Пример аудиозаписи до обработки детектором речевой активности

Примеры аудиозаписей после обработки детектором речевой активности

2.4 Модуль предобработки речевых аудиоданных

Модуль выполняет предобработку речевых аудиоданных, в данном случае извлекается спектрограмма из исходной аудиодорожки. Команда для запуска предобработки речевых аудиоданных:

```
openav_preprocess_audio --config <путь_к_вашему_конфигурационному_файлу>.yaml
```

Важно: Для запуска команды необходимо обязательно указать путь к конфигурационному файлу. Запускать программу необходимо из директории, где она расположена.

2.4.1 Конфигурационный файл

Конфигурационный файл включает в себя следующие настройки:

- Отображение процесса выполнения программы в терминале (командной строке)
- Работа с файловой системой
- Параметры предобработки речевых аудиоданных

Пример конфигурационного файла

2.4.2 Отображение процесса выполнения команды в терминале

Параметр	Тип	Значение по умолчанию	Описание
hide_metadata	bool	false	Включение отображения метаданных
hide_libs_vers	bool	false	Включение отображения версий установленных библиотек в командной строке

2.4.3 Работа с файловой системой

Параметр	Тип	Значение по умолчанию	Описание
path_to_dataset	str	<путь_к_исходным_данным>	Директория, где находятся данные, которые необходимо загрузить или проверить
path_to_dataset_audio	str	<путь_к_конечным_данным>	Директория, в которую будут сохраняться аудиоданные после предобработки
depth	int	1	Глубина иерархии для получения данных. Указывается количество подкаталогов в директории path_to_dataset
ext_search_files	list	["mov", "mp4", "webm", "wav"]	Список расширений файлов, которые будут обрабатываться. Указывать можно как для видео, так и для аудио
clear_dir_audio	bool	true	Предварительная очистка директории с предобработанными аудиоданными
save_raw_data	bool	true	Сохранение данных в формате numpy

2.4.4 Параметры предобработки речевых аудиоданных

Параметр	Тип	Значение по умолчанию	Описание
sampling_rate	int	16000	Частота дискретизации аудиосигнала. Доступные значения 16000; 22050; 44100; 48000
n_fft	int	2048	Размер параметра FFT, создает $n_fft // 2 + 1$ бин. Допустимы значения в пределе от 256 до 2048
hop_length	int	512	Длина перехода между окнами STFT. Допустимы значения в пределе от 64 до 512
n_mels	int	128	Количество фильтроблоков mel. Допустимы значения в пределе от 20 до 512
power	float	2.0	Показатель степени магнитудной спектрограммы. Должен быть либо 1.0, либо 2.0
center	bool	true	Включение установки отступов с обеих сторон относительно центральной части аудиодорожки
pad_mode	str	reflect	Управление отступами, применяется когда значение параметра center = True. Доступные значения constant, reflect, replicate, circular. По умолчанию reflect
norm	str	slaney	Нормализация площади. Отношение треугольных мел-весов к ширине мел-зоны
dpi	int	600	Качество изображения спектрограммы. Доступные значения 72; 96; 150; 300; 600; 1200
color_gradients	str	magma	Выбор цветовой схемы итогового изображения спектрограммы. Доступные значения viridis, plasma, inferno, magma, cividis

2.5 Модуль предобработки речевых видеоданных

Модуль выполняет предобработку речевых видеоданных, в данном случае извлекаются области губ из исходных видеок кадров. Команда для запуска предобработки речевых видеоданных:

```
openav_preprocess_video --config <путь_к_вашему_конфигурационному_файлу>.yaml
```

Важно: Для запуска команды необходимо обязательно указать путь к конфигурационному файлу. Запускать программу необходимо из директории, где она расположена.

2.5.1 Конфигурационный файл

Конфигурационный файл включает в себя следующие настройки:

- Отображение процесса выполнения программы в терминале (командной строке)
- Работа с файловой системой
- Параметры предобработки речевых видеоданных

Пример конфигурационного файла

2.5.2 Отображение процесса выполнения команды в терминале

Параметр	Тип	Значение по умолчанию	Описание
hide_metadata	bool	false	Включение отображения метаданных
hide_libs_vers	bool	false	Включение отображения версий установленных библиотек в командной строке

2.5.3 Работа с файловой системой

Параметр	Тип	Значение по умолчанию	Описание
path_to_dataset	str	<путь_к_исходным_данным>	Директория, где находятся данные, которые необходимо загрузить или проверить
path_to_dataset_video	str	<путь_к_конечным_данным>	Директория, в которую будут сохраняться данные после предобработки
depth	int	2	Глубина иерархии для получения данных. Указывается количество подкаталогов в директории path_to_dataset
ext_search_files	list	["mov", "mp4", "webm"]	Список расширений файлов, которые будут обрабатываться.
clear_dir_video	bool	true	Предварительная очистка директории с предобработанными видеоданными
save_raw_data	bool	true	Сохранение данных в формате numpy

2.5.4 Параметры предобработки речевых видеоданных

Параметр	Тип	Значение по умолчанию	Описание
dpi	int	600	Качество изображения. Доступные значения 72; 96; 150; 300; 600; 1200
resize	bool	true	Разрешить изменять полученное изображение. Этот параметр активирует следующие параметры: <code>width</code> , <code>height</code> , <code>resize_resample</code> . При значении <code>false</code> изменение изображения заблокировано
resize_resample	str	nearest	Фильтры применяемые для масштабирования конечного изображения. Доступные значения <code>nearest</code> , <code>bilinear</code> , <code>lanczos</code>
width	int	112	Ширина изображения извлеченной области губ
height	int	112	Высота изображения извлеченной области губ
color_mode	str	rgb	Цветовая гамма конечного изображения. Доступные значения: <code>gray</code> - изображение в градациях серого; <code>rgb</code> - изображение в цветном формате

2.6 Модуль аугментации данных

Данный модуль позволяет генерировать дополнительные данные с помощью различных параметров аугментации. Команда для запуска аугментации данных:

```
python ./openav/api/augmentation.py --config <путь_к_вашему_конфигурационному_файлу>.yaml
```

Важно: Для запуска команды необходимо обязательно указать путь к конфигурационному файлу. Запускать программу необходимо из директории, где она расположена.

2.6.1 Конфигурационный файл

Конфигурационный файл включает в себя следующие настройки:

- Отображение процесса выполнения программы в терминале (командной строке)
- Работа с файловой системой
- Параметры аугментации данных

Пример конфигурационного файла

2.6.2 Отображение процесса выполнения команды в терминале

Параметр	Тип	Значение по умолчанию	Описание
hide_metadata	bool	false	Включение отображения метаданных
hide_libs_vers	bool	false	Включение отображения версий установленных библиотек в командной строке

2.6.3 Работа с файловой системой

Параметр	Тип	Значение по умолчанию	Описание
path_to_input_directory	str	<путь_к_исходным_данным>	Директория, где находятся данные, которые необходимо аугментировать
path_to_output_directory	str	<путь_к_аугментированным_данным>	Директория, куда сохраняются аугментированные данные
clear_diraug	bool	true	Очистка директории, в которую сохраняются аугментированные данные
depth	int	1	Глубина иерархии для получения данных. Указывается количество подкаталогов в директории <code>path_to_input_directory</code>
ext_search_files	list	["jpg", "png"]	Список расширений файлов, которые будут обрабатываться

2.6.4 Параметры аугментации данных

Параметр	Тип	Значение по умолчанию	Описание
crop_px_min	int	0	Минимальное количество пикселей для обрезки изображения с каждой стороны. Диапазон значений от 0 и 1000000
crop_px_max	int	16	Максимальное количество пикселей для обрезки изображения с каждой стороны. Диапазон значений от 0 и 1000000
crop_percent_min	float	0	Минимальный процент для обрезки изображения с каждой стороны. Диапазон значений от 0 и 1.0
crop_percent_max	float	0.5	Максимальный процент для обрезки изображения с каждой стороны. Диапазон значений от 0 и 1.0
flip_lr_probability	float	0.5	Значение коэффициента вероятности отражения по вертикальной оси. Диапазон значений от 0 и 1.0
flip_ud_probability	float	0.5	Значение коэффициента вероятности отражения по горизонтальной оси. Диапазон значений от 0 и 1.0
blur_min	float	0	Минимальное значение коэффициента размытия изображения. Диапазон значений от 0 и 3.0
blur_max	float	1	Максимальное значение коэффициента размытия изображения. Диапазон значений от 0 и 3.0
scale_x_min	float	0.5	Минимальное значение масштабирования по оси X. Диапазон значений от 0 и 10.0
scale_x_max	float	2	Максимальное значение масштабирования по оси X. Диапазон значений от 0 и 10.0
scale_y_min	float	0.5	Минимальное значение масштабирования по оси Y. Диапазон значений от 0 и 10.0
scale_y_max	float	2	Максимальное значение масштабирования по оси X. Диапазон значений от 0 и 10.0
rotate_min	int	-45	Минимальное значение угла поворота изображения. Диапазон значений от -90 и 90
rotate_max	int	45	Максимальное значение угла поворота изображения. Диапазон значений от -90 и 90
contrast_min	float	0	Минимальное значение коэффициента контрастности. Диапазон значений от -10.0 и 10.0
contrast_max	float	1.0	Максимальное значение коэффициента контрастности. Диапазон значений от -10.0 и 10.0
alpha	float	0.7	Значения коэффициента MixUp. Диапазон значений от 0 и 1.0
count	int	5	Количество применений процесса аугментации к изображению

2.6.5 Примеры

Ниже приведен ряд изображений, сгенерированных с помощью модуля аугментации данных.

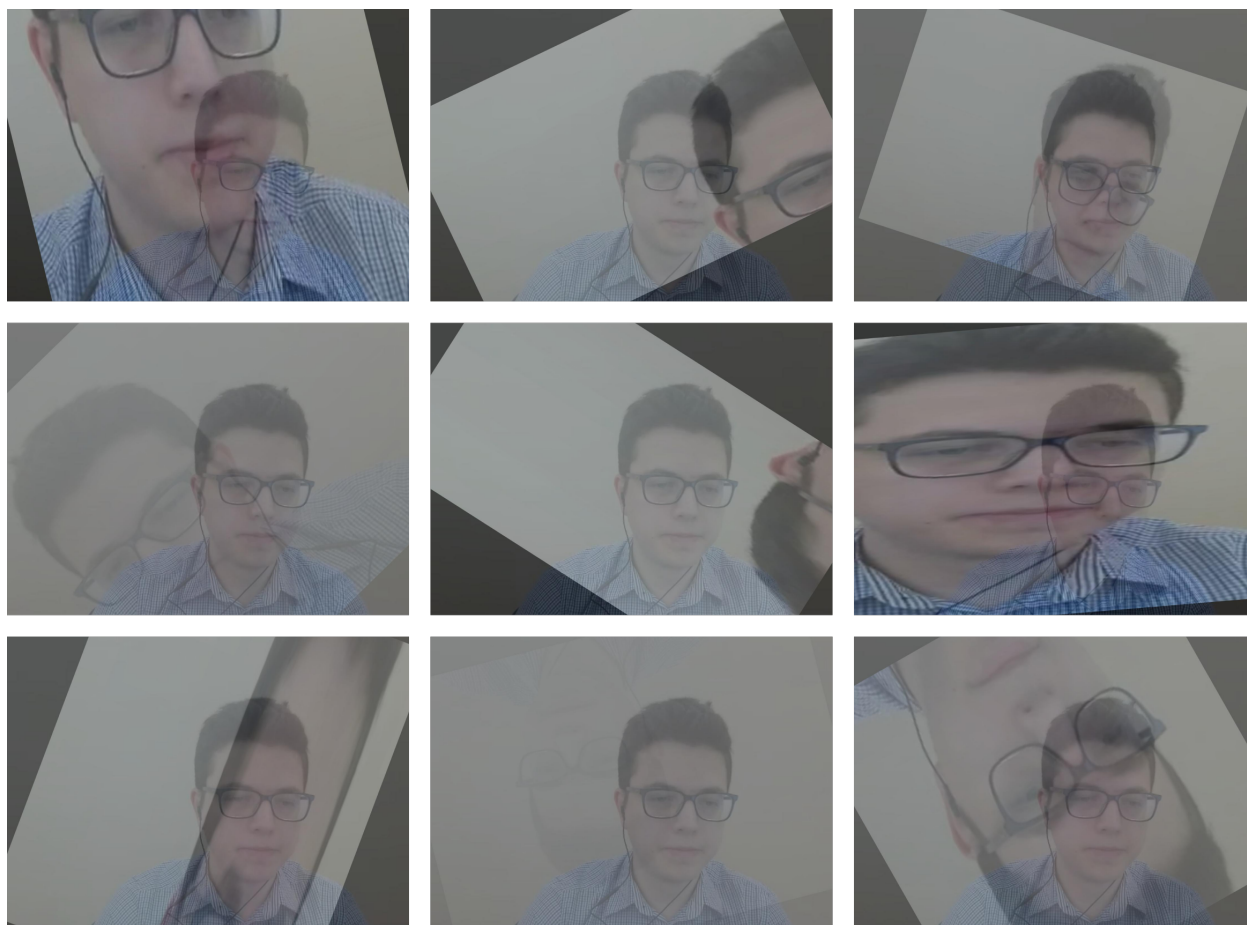


Рис. 3:

2.7 Модуль обучения нейросетевых моделей

2.7.1 Обучение акустических нейросетевых моделей

Команда для запуска процесса обучения акустических нейросетевых моделей:

```
python ./openav/api/train_audio.py --config <путь_к_вашему_конфигурационному_файлу>.yaml
```

Важно: Для запуска команды необходимо обязательно указать путь к конфигурационному файлу. Запускать программу необходимо из директории, где она расположена.

Конфигурационный файл

Конфигурационный файл включает в себя следующие настройки:

- Отображение процесса выполнения программы в терминале (командной строке)
- Работа с файловой системой
- Параметры настройки процесса обучения нейросетевых моделей

Пример конфигурационного файла

Важно: Для обучения на видеоадаптере потребуется вручную установить библиотеку для машинного обучения PyTorch с использованием GPU.

Инструкция для обучения нейросетевых моделей с использованием GPU

Отображение процесса выполнения команды в терминале

Параметр	Тип	Значение по умолчанию	Описание
hide_metadata	bool	false	Включение отображения метаданных
hide_libs_vers	bool	false	Включение отображения версий установленных библиотек в командной строке

Работа с файловой системой

Параметр	Тип	Значение по умолчанию	Описание
path_to_dataset	str	<путь_к_набору_данных>	Директория, где размещается подготовленный набор данных для обучения нейросетевых моделей

Параметры процесса обучения акустических нейросетевых моделей

Параметр	Тип	Значение по умолчанию	Описание
len_audio	int	ваше значение	Количество аудиофайлов
size_spec	int	width: 224 height: 224	Размер входного изображения спектрограммы в px
padding_spec	bool	True	Добавление отступов на изображениях спектрограмм
seed	int	42	Параметр для инициализации случайных процессов, который обеспечивает воспроизводимость результатов и одинаковые начальные условия
batch_size	int	8	Размер батча. Общее число тренировочных объектов, представленных в одном батче. Устанавливается в зависимости от возможности вашего оборудования
channels_spec	int	1	Количество каналов изображения. 1- одноканальное изображение (в серых тонах), 3 - трёхканальное изображение (RGB)
lr	float	0.0001	Коэффициент скорости обучения. Чем меньше значение, тем дольше будет идти обучение модели. Однако, стоит помнить, что может наступить переобучение модели. Данный коэффициент подбирается эмпирическим путем
epoch	int	100	Количество эпох обучения модели. Проход одной эпохи значит, что весь датасет прошел через нейронную сеть в прямом и обратном направлении только один раз. Параметр <code>epoch_stop</code> позволяет избежать переобучения модели
epoch_stop	int	5	Количество эпох, в течении которых модель не прогрессирует в обучении. Т.е. если по прошествии, например, 5 эпох модель не показывала прирост <code>accuracy</code> (либо ваша метрика), то в данном случае процесс обучения останавливается и сохраняется модель на эпохе с наибольшей точностью. Данный параметр позволяет предотвратить переобучение модели, а также снизить длительность обучения

2.7.2 Обучение визуальных нейросетевых моделей

Команда для запуска процесса обучения визуальных нейросетевых моделей:

```
python ./openav/api/train_video.py --config <путь_к_вашему_конфигурационному_файлу>.yaml
```

Важно: Для запуска команды необходимо обязательно указать путь к конфигурационному файлу. Запускать программу необходимо из директории, где она расположена.

Конфигурационный файл

Конфигурационный файл включает в себя следующие настройки:

- Отображение процесса выполнения программы в терминале (командной строке)
- Работа с файловой системой
- Параметры настройки процесса обучения нейросетевых моделей

Пример конфигурационного файла

Важно: Для обучения на видеоадаптере потребуется вручную установить библиотеку для машинного обучения PyTorch с использованием GPU.

Инструкция для обучения нейросетевых моделей с использованием GPU

Отображение процесса выполнения команды в терминале

Параметр	Тип	Значение по умолчанию	Описание
hide_metadata	bool	false	Включение отображения метаданных
hide_libs_vers	bool	false	Включение отображения версий установленных библиотек в командной строке

Работа с файловой системой

Параметр	Тип	Значение по умолчанию	Описание
path_to_dataset	str	<путь_к_набору_данных>	Директория, где размещается подготовленный набор данных для обучения нейросетевых моделей

Параметры процесса обучения визуальных нейросетевых моделей

Параметр	Тип	Значение по умолчанию	Описание
len_video	int	ваше значение	Количество видеофайлов
size_lips	int	width: 112 height: 112	Размер входного изображения области губ в px
padding_lips	bool	True	Добавление отступов на изображениях губ
seed	int	42	Параметр для инициализации случайных процессов, который обеспечивает воспроизводимость результатов и одинаковые начальные условия
batch_size	int	8	Размер батча. Общее число тренировочных объектов, представленных в одном батче. Устанавливается в зависимости от возможности вашего оборудования
channels_lips	int	1	Количество каналов изображения. 1- одноканальное изображение (в серых тонах), 3 - трёхканальное изображение (RGB)
lr	float	0.0001	Коэффициент скорости обучения. Чем меньше значение, тем дольше будет идти обучение модели. Однако, стоит помнить, что может наступить переобучение модели. Данный коэффициент подбирается эмпирическим путем
epoch	int	100	Количество эпох обучения модели. Проход одной эпохи значит, что весь датасет прошел через нейронную сеть в прямом и обратном направлении только один раз. Параметр <code>epoch_stop</code> позволяет избежать переобучения модели
epoch_stop	int	5	Количество эпох, в течении которых модель не прогрессирует в обучении. Т.е. если по проставии, например, 5 эпох модель не показывала прирост <code>accuracy</code> (либо ваша метрика), то в данном случае процесс обучения останавливается и сохраняется модель на эпохе с наибольшей точностью. Данный параметр позволяет предотвратить переобучение модели, а также снизить длительность обучения

2.8 Модуль распознавания речи

2.8.1 Модуль распознавания акустической речи

Команда для запуска модуля распознавания акустической речи:

```
python ./openav/api/test_audio.py --config <путь_к_вашему_конфигурационному_файлу>.yaml
```

Важно: Для запуска команды необходимо обязательно указать путь к конфигурационному файлу.

Запускать программу необходимо из директории, где она расположена.

Конфигурационный файл

Конфигурационный файл включает в себя следующие настройки:

- Отображение процесса выполнения программы в терминале (командной строке)
- Работа с файловой системой
- Параметры процесса распознавания акустической речи

Пример конфигурационного файла

Отображение процесса выполнения команды в терминале

Параметр	Тип	Значение по умолчанию	Описание
hide_metadata	bool	false	Включение отображения метаданных
hide_libs_vers	bool	false	Включение отображения версий установленных библиотек в командной строке

Работа с файловой системой

Параметр	Тип	Значение по умолчанию	Описание
path_to_dataset	str	<путь_к_набору_данных>	Директория, где размещается подготовленный набор данных для тестирования обученных нейросетевых моделей
path_to_model	str	<путь_к_моделям>	Директория, где размещаются обученные нейросетевые модели
depth	int	3	Глубина иерархии для получения данных. Указывается количество подкаталогов в директории path_to_dataset
ext_search_files	list	["wav", "aac"]	Список расширений файлов, которые будут обрабатываться

Параметры процесса распознавания акустической речи

Параметр	Тип	Значение по умолчанию	Описание
size_spec	int	width: 224 height: 224	Размер входного изображения спектрограммы в px
channels_spec	int	1	Количество каналов изображения. 1- одноканальное изображение (в серых тонах), 3 - трёхканальное изображение (RGB)
metric	str	accuracy	Метрика, в соответствии с которой будет вывод результатов тестирования обученных нейросетевых моделей

2.8.2 Модуль распознавания визуальной речи

Команда для запуска модуля распознавания визуальной речи:

```
python ./openav/api/test_video.py --config <путь_к_вашему_конфигурационному_файлу>.yaml
```

Важно: Для запуска команды необходимо обязательно указать путь к конфигурационному файлу. Запускать программу необходимо из директории, где она расположена.

Конфигурационный файл

Конфигурационный файл включает в себя следующие настройки:

- Отображение процесса выполнения программы в терминале (командной строке)
- Работа с файловой системой
- Параметры процесса распознавания визуальной речи

Пример конфигурационного файла

Отображение процесса выполнения команды в терминале

Параметр	Тип	Значение по умолчанию	Описание
hide_metadata	bool	false	Включение отображения метаданных
hide_libs_vers	bool	false	Включение отображения версий установленных библиотек в командной строке

Работа с файловой системой

Параметр	Тип	Значение по умолчанию	Описание
path_to_dataset	str	<путь_к_набору_данных>	Директория, где размещается подготовленный набор данных для тестирования обученных нейросетевых моделей
path_to_model	str	<путь_к_моделям>	Директория, где размещаются обученные нейросетевые модели
depth	int	3	Глубина иерархии для получения данных. Указывается количество подкаталогов в директории path_to_dataset
ext_search_files	list	["mov", "mp4", "webm"]	Список расширений файлов, которые будут обрабатываться

Параметры процесса распознавания визуальной речи

Параметр	Тип	Значение по умолчанию	Описание
size_lips	int	width: 112 height: 112	Размер входного изображения области губ в px
channels_lips	int	1	Количество каналов изображения. 1- одноканальное изображение (в серых тонах), 3 - трёхканальное изображение (RGB)
metric	str	accuracy	Метрика, в соответствии с которой будет вывод результатов тестирования обученных нейросетевых моделей

2.9 Модуль объединения модальностей

2.9.1 Модуль объединения модальностей

Команда для запуска модуля объединения модальностей:

```
openav_train_audiovisual --config <путь_к_вашему_конфигурационному_файлу>.yaml
```

Важно: Для запуска команды необходимо обязательно указать путь к конфигурационному файлу. Запускать программу необходимо из директории, где она расположена.

Конфигурационный файл

Конфигурационный файл включает в себя следующие настройки:

- Отображение процесса выполнения программы в терминале (командной строке)
- Работа с файловой системой
- Параметры процесса обучения аудиовизуальных нейросетевых моделей

Пример конфигурационного файла

Важно: Для обучения на видеоадапторе потребуется вручную установить библиотеку для машинного обучения PyTorch с использованием GPU.

Инструкция для обучения нейросетевых моделей с использованием GPU

Отображение процесса выполнения команды в терминале

Параметр	Тип	Значение по умолчанию	Описание
hide_metadata	bool	false	Включение отображения метаданных
hide_libs_vers	bool	false	Включение отображения версий установленных библиотек в командной строке

Работа с файловой системой

Параметр	Тип	Значение по умолчанию	Описание
path_to_dataset	str	<путь_к_набору_данных>	Директория, где размещается подготовленный набор данных для обучения нейросетевых моделей
subfolders	str	train: "train" val: "val" test: "test"	Директории с обучающей, тестовой и валидационной выборками
path_to_model_fa	str	<путь_к_весам_аудио_м>	Путь к предобученной модели, которая используется для извлечения акустических признаков. Предобученную модель можно скачать по ссылке
path_to_model_fv	str	<путь_к_весам_видео_м>	Путь к предобученной модели, которая используется для извлечения визуальных признаков. Предобученную модель можно скачать по ссылке
path_to_save_models	str	<путь_сохранения_обуч>	Путь, по которому будут храниться обученные нейросетевые модели

Параметры процесса обучения аудиовизуальных нейросетевых моделей

Параметр	Тип	Значение по умолчанию	Описание
n_classes	int	26	Количество классов для задачи классификации. Соответствуют количеству фраз из базы данных
classes	list	["1_Левая", "2_Правая", ..., "26_Калибровка"]	Список названий классов, которые представлены в базе данных
seed	int	42	Параметр, задающий начальное значение генератора псевдослучайных чисел в PyTorch. Установка фиксированного seed обеспечивает воспроизводимость результатов между разными запусками программы. При одинаковом seed все случайные операции, такие как инициализация весов нейронной сети, перемешивание данных и др., будут давать одинаковые результаты.
max_segment	int	2	Гиперпараметр, определяющий максимальное количество перекрывающихся сегментов, на которые разбиваются длинные последовательности аудио и видео данных перед подачей их на вход нейронной сети для обучения или вывода
epochs	int	150	Количество эпох обучения модели. Проход одной эпохи значит, что весь датасет прошел через нейронную сеть в прямом и обратном направлении только один раз. Параметр patience позволяет избежать переобучения модели
patience	int	15	Количество эпох, в течении которых модель не прогрессирует в обучении. Т.е. если по прошествии, например, 5 эпох модель не показывала прирост accuracy (либо ваша метрика), то в данном случае процесс обучения останавливается и сохраняется модель на эпохе с наибольшей точностью. Данный параметр позволяет предотвратить переобучение модели, а также снизить длительность обучения
batch_size	int	2	Размер батча. Общее число тренировочных объектов, представленных в одном батче. Устанавливается в зависимости от возможности вашего оборудования
learning_rate	float	0.0001	Коэффициент скорости обучения. Чем меньше значение, тем дольше будет идти обучение модели. Однако, стоит помнить, что может наступить переобучение модели. Данный коэффициент подбирается эмпирическим путем
weight_decay	float	0.0	Параметр, используемый для регуляризации весов нейронной сети путем добавления L2-регуляризации к функции потерь во время обучения. Позволяет предотвратить переобучение и подбирается экспериментальным путем
hidden_units	int	256	Количество скрытых единиц (hidden units) в декодере нейронной сети. Этот параметр определяет размерность внутреннего представления данных в декодере, что влияет на емкость (expressive capacity) и способность декодера извлекать и обобщать сложные зависимости в данных. Подбирается эмпирическим путем
hidden_features	int	128	Количество скрытых признаков (hidden features)

2.9.2 Модуль распознавания аудиовизуальной речи

Команда для запуска модуля распознавания аудиовизуальной речи:

```
openav_test_audiovisual --config <путь_к_вашему_конфигурационному_файлу>.yaml
```

Важно: Для запуска команды необходимо обязательно указать путь к конфигурационному файлу. Запускать программу необходимо из директории, где она расположена.

Конфигурационный файл

Конфигурационный файл включает в себя следующие настройки:

- Отображение процесса выполнения программы в терминале (командной строке)
- Работа с файловой системой
- Параметры процесса распознавания аудиовизуальных речи

Пример конфигурационного файла

Отображение процесса выполнения команды в терминале

Параметр	Тип	Значение по умолчанию	Описание
hide_metadata	bool	false	Включение отображения метаданных
hide_libs_vers	bool	false	Включение отображения версий установленных библиотек в командной строке

Работа с файловой системой

Параметр	Тип	Значение по умолчанию	Описание
path_to_dataset	str	<путь_к_набору_данных>	Директория, где размещается подготовленный набор данных для обучения нейросетевых моделей
subfolders	str	test: "test"	Директория с тестовой выборкой
path_to_model	str	<путь_к_обученной_модели>	Путь, по которому находится обученная нейросетевая модель
path_to_save_confusion	str	<путь_сохранения_матрицы>	Путь, по которому будут храниться матрицы спутывания

Параметры процесса распознавания аудиовизуальной речи

Параметр	Тип	Значение по умолчанию	Описание
n_classes	int	26	Количество классов для задачи классификации. Соответствуют количеству фраз из базы данных
classes	list	["1_Левая", "2_Правая", ... "26_Калибровка"]	Список названий классов, которые представлены в базе данных
max_segment	int	2	Гиперпараметр, определяющий максимальное количество перекрывающихся сегментов, на которые разбиваются длинные последовательности аудио и видео данных перед подачей их на вход нейронной сети для обучения или вывода
hidden_units	int	256	Количество скрытых единиц (hidden units) в декодере нейронной сети. Этот параметр определяет размерность внутреннего представления данных в декодере, что влияет на емкость (expressive capacity) и способность декодера извлекать и обобщать сложные зависимости в данных
hidden_features	int	128	Количество скрытых признаков (hidden features) или временных шагов, используемых в нейросетевых моделях. Этот гиперпараметр связан с тем, как модель обрабатывает последовательные данные, такие как аудио и видео. Увеличение этого параметра позволяет модели обрабатывать более длинные входные последовательности, но также увеличивает вычислительную сложность и требования к памяти, уменьшение может ускорить обучение, но при этом модель будет видеть только более короткие временные зависимости
input_dim	int	512	Размерность входных векторов признаков для аудио и видео данных, подаваемых в модель. Например, параметр input_dim=512 указывает, что на вход нейронной сети будут подаваться векторы признаков размера 512 для каждого временного шага последовательности. Размер входного вектора обычно является результатом предварительной обработки и извлечения низкоуровневых признаков из исходных аудио/видео данных с помощью отдельных моделей или методов обработки сигналов.
shape_audio	int	channels: "1" n_mels: "64" samples: "306"	Здесь определяется размерность входных тензоров аудиоданных, ожидаемые моделью. Это форма задается тремя числами channels - количество каналов в аудиоданных, чаще всего аудио является моно, поэтому здесь канал равен 1. При стерео он равен 2. n_mels - количество мел-частотных кепстральных коэффициентов (MFCC), используемых для представления аудиоданных. samples - количество временных выборок или окон, входящих в одно аудио представление. Являются фиксированными параметрами
shape_video	int	frames: "29" channels: "3" width: "88" height: "88"	Здесь определяется размерность входных тензоров видеоданных, ожидаемые моделью. frames - это количество видеок кадров в одном сегменте. channels - это количество цветочных каналов (R, G, B)

Пример построения матрицы спутывания

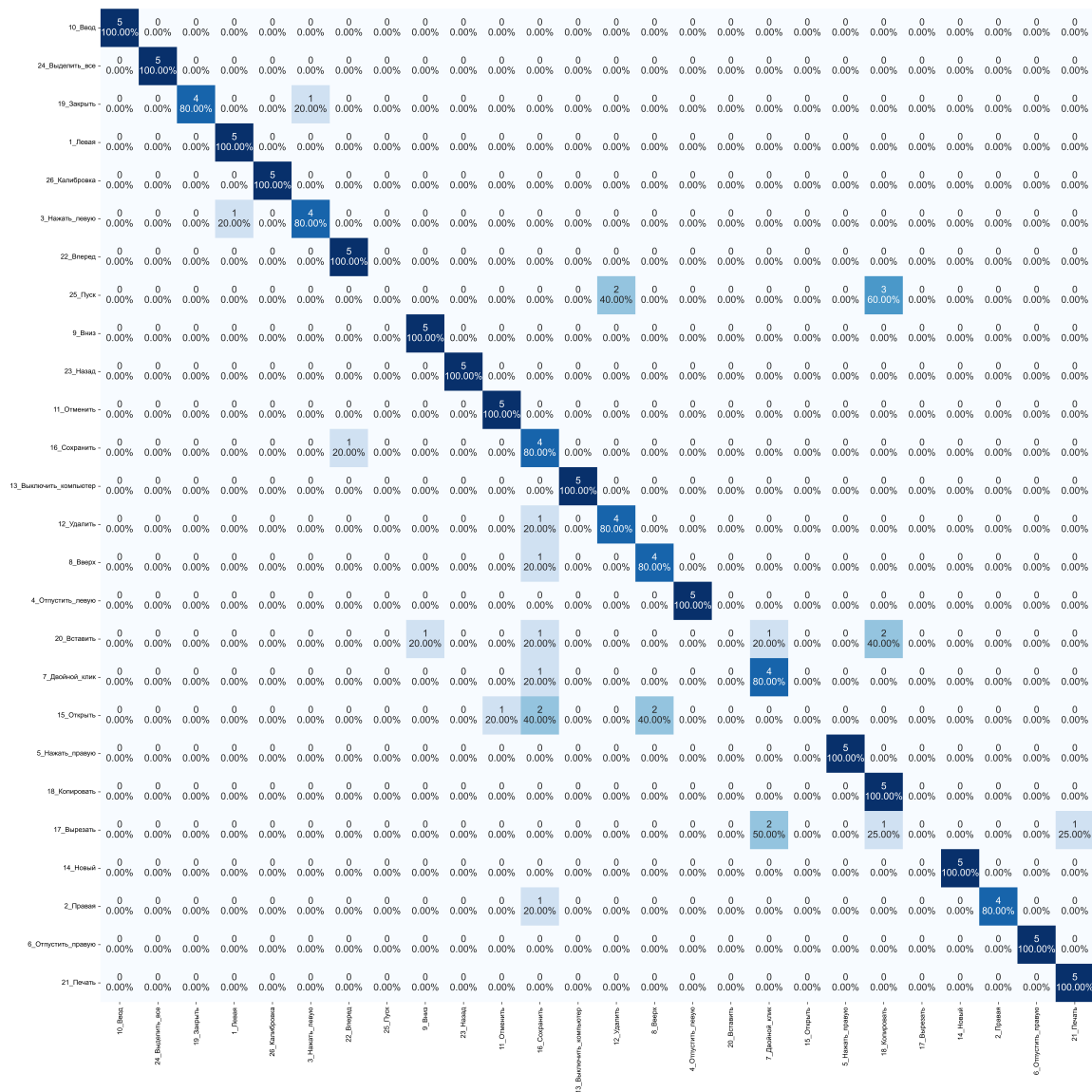


Рис. 4:

3.1 Работа с Shell

```
class openav.modules.trml.shell.Shell
```

Базовые классы: `object`

Класс для работы с Shell

```
static add_line()
```

Добавление линии во весь экран

```
static clear()
```

Очистка консоли

3.2 Пользовательские исключения

```
exception openav.modules.core.exceptions.BlurError
```

Базовые классы: *CustomException*

Указан неверный диапазон значений размытия

```
exception openav.modules.core.exceptions.ContrastError
```

Базовые классы: *CustomException*

Указан неверный диапазон значений контрастности

```
exception openav.modules.core.exceptions.CropPXEError
```

Базовые классы: *CustomException*

Указан неверный диапазон обрезки в пикселях

`exception openav.modules.core.exceptions.CropPercentsError`

Базовые классы: *CustomException*

Указан неверный диапазон обрезки в процентах

`exception openav.modules.core.exceptions.CustomException`

Базовые классы: *Exception*

Класс для всех пользовательских исключений

`exception openav.modules.core.exceptions.FlipLRProbabilityError`

Базовые классы: *CustomException*

Указано неверное значение вероятности отражения по вертикальной оси

`exception openav.modules.core.exceptions.FlipUDPProbabilityError`

Базовые классы: *CustomException*

Указано неверное значение вероятности отражения по горизонтальной оси

`exception openav.modules.core.exceptions.InvalidContentLength`

Базовые классы: *CustomException*

Не определен размер файла для загрузки

`exception openav.modules.core.exceptions.IsNestedCatalogsNotFoundError`

Базовые классы: *CustomException*

Вложенные директории, где хранятся данные не найдены

`exception openav.modules.core.exceptions.IsNestedDirectoryANotFoundError`

Базовые классы: *CustomException*

Вложенная директория, для аудиофрагментов не найдена

`exception openav.modules.core.exceptions.IsNestedDirectoryVNotFoundError`

Базовые классы: *CustomException*

Вложенная директория, для видеофрагментов не найдена

`exception openav.modules.core.exceptions.MixUpAlphaError`

Базовые классы: *CustomException*

Указан неверный коэффициент для MixUp-аугментации

`exception openav.modules.core.exceptions.PresetCFREncodeVideoError`

Базовые классы: *CustomException*

Указан неподдерживаемый параметр обеспечивающий определенную скорость кодирования и сжатия видео

`exception openav.modules.core.exceptions.RotateError`

Базовые классы: *CustomException*

Указан неверный диапазон значений угла наклона

`exception openav.modules.core.exceptions.SRInputTypeError`

Базовые классы: *CustomException*

Указан неподдерживаемый тип файла для распознавания речи

exception openav.modules.core.exceptions.SamplingRateError

Базовые классы: *CustomException*

Указана неподдерживаемая частота дискретизации речевого сигнала

exception openav.modules.core.exceptions.ScaleError

Базовые классы: *CustomException*

Указан неверный диапазон значений масштабирования

exception openav.modules.core.exceptions.TypeEncodeVideoError

Базовые классы: *CustomException*

Указан неподдерживаемый тип кодирования видео

exception openav.modules.core.exceptions.TypeMessagesError

Базовые классы: *CustomException*

Указан неподдерживаемый тип сообщения

exception openav.modules.core.exceptions.WindowSizeSamplesError

Базовые классы: *CustomException*

Указано неподдерживаемое количество выборок в каждом окне

3.3 Логирование

class openav.modules.core.logging.Logging(*path_to_logs: str = <property object>*)

Базовые классы: object

Класс для логирования

Параметры

path_to_logs (str) – Путь к директории для сохранения LOG файлов

property *check_create_logger: bool*

Получение создания регистратора и обработчика для логирования

Результат

True если регистратор и обработчик созданы, в обратном случае **False**

Тип результата

bool

property *logger_gui: bool*

Получение/установка логирования GUI

Параметры

(bool) – **True** если запущен GUI, в обратном случае **False**

Результат

True если запущен GUI, в обратном случае **False**

Тип результата

bool

property *path_to_logs: str*

Получение/установка пути к директории для сохранения LOG файлов

Параметры
(*str*) – Путь

Результат
Путь

Тип результата
str

`static re_inv_chars(path: str) → str`

Удаление недопустимых символов из пути

Параметры
path (*str*) – Путь

Результат
Путь

Тип результата
str

3.4 Определение языка

`class openav.modules.core.language.Language(path_to_logs: str = <property object>, lang: str = <property object>)`

Базовые классы: *Logging*

Класс для интернационализации (I18N) и локализации (L10N)

Параметры

- *path_to_logs* (*str*) – Смотреть *path_to_logs*
- *lang* (*str*) – Язык

`property lang: str`

Получение/установка текущего языка

Параметры
(*str*) – Язык, доступные варианты:

- "ru" - Русский язык (по умолчанию)
- "en" - Английский язык

Результат
Язык

Тип результата
str

`property locales: List[str]`

Получение поддерживаемых языков

Результат
Список поддерживаемых языков

Тип результата
List[str]

property path_to_locales: str

Получение директории с языковыми пакетами

Результат

Директория с языковыми пакетами

Тип результата

str

3.5 Сообщения

```
class openav.modules.core.messages.Messages(path_to_logs: str = <property object>, lang: str =
<property object>)
```

Базовые классы: *Language*

Класс для сообщений

Параметры

- path_to_logs (*str*) – Смотреть *path_to_logs*
- lang (*str*) – Смотреть *lang*

3.6 Настройки

```
class openav.modules.core.settings.Settings(path_to_logs: str = <property object>, lang: str =
<property object>)
```

Базовые классы: *Messages*

Класс для настроек

Параметры

- path_to_logs (*str*) – Смотреть *path_to_logs*
- lang (*str*) – Смотреть *lang*

property chunk_size: int

Получение/установка размера загрузки файла из сети за 1 шаг

Параметры

(int) – Размер загрузки файла из сети за 1 шаг

Результат

Размер загрузки файла из сети за 1 шаг

Тип результата

int

property clear_line: str

Получение очистки линии в терминале

Результат

Очистка линии в терминале

Тип результата

str

`property color_blue: str`

Получение синего цвета текста в терминале

Результат

Цвет текста в терминале

Тип результата

str

`property color_green: str`

Получение зеленого цвета текста в терминале

Результат

Цвет текста в терминале

Тип результата

str

`property color_red: str`

Получение красного цвета текста в терминале

Результат

Цвет текста в терминале

Тип результата

str

`property dir_va_names: List[str]`

Получение/установка списка с названиями директорий для видео и аудио

Параметры

(List[str]) – Список с директориями

Результат

Список с директориями

Тип результата

List[str]

`property ext_search_files: List[str]`

Получение/установка списка с расширениями искомых файлов

Параметры

(List[str]) – Список с расширениями искомых файлов

Результат

Список с расширениями искомых файлов

Тип результата

List[str]

`property ignore_dirs: List[str]`

Получение/установка списка с директориями не входящими в выборку

Параметры

(List[str]) – Список с директориями

Результат

Список с директориями

Тип результата

List[str]

property path_to_dataset: str

Получение/установка пути к директории набора данных

Параметры

(str) – Путь

Результат

Путь

Тип результата

str

property path_to_dataset_vad: str

Получение/установка пути к директории набора данных состоящего из фрагментов аудио-визуального сигнала (VAD)

Параметры

(str) – Путь

Результат

Путь

Тип результата

str

property path_to_dataset_vosk_sr: str

Получение/установка пути к директории набора данных состоящего из фрагментов аудио-визуального сигнала (VOSK)

Параметры

(str) – Путь

Результат

Путь

Тип результата

str

property path_to_save_models: str

Получение/установка пути к директории для сохранения моделей

Параметры

(str) – Путь

Результат

Путь

Тип результата

str

property text_bold: str

Получение жирного начертания текста в терминале

Результат

Жирное начертание текста в терминале

Тип результата

str

property text_end: str

Получение сброса оформления текста в терминале

Результат

Сброс оформления текста в терминале

Тип результата

str

3.7 Ядро

```
class openav.modules.core.core.CoreMessages(path_to_logs: str = <property object>, lang: str = <property object>)
```

Базовые классы: *Settings*

Класс для сообщений

Параметры

- path_to_logs (str) – Смотреть *path_to_logs*
- lang (str) – Смотреть *lang*

```
class openav.modules.core.core.Core(path_to_logs: str = <property object>, lang: str = <property object>)
```

Базовые классы: *CoreMessages*

Класс-ядро модулей

Параметры

- path_to_logs (str) – Смотреть *path_to_logs*
- lang (str) – Смотреть *lang*

```
build_args(description: str, conv_to_dict: bool = True, out: bool = True) → Dict[str, Any]
```

Построение аргументов командной строки

Параметры

- description (str) – Описание парсера командной строки
- conv_to_dict (bool) – Преобразование списка аргументов командной строки в словарь
- out (bool) – Печатать процесс выполнения

Результат

Словарь со списком аргументов командной строки

Тип результата

Dict[str, Any]

```
clear_shell(cls: bool = True, out: bool = True) → bool
```

Очистка консоли

Параметры

- cls (bool) – Очистка консоли
- out (bool) – Печатать процесс выполнения

Результат

True если консоль очищена, в обратном случае **False**

Тип результата

bool

property df_pkgs: DataFrame

Получение DataFrame с версиями установленных библиотек

Результат**DataFrame** с версиями установленных библиотек**Тип результата**

pd.DataFrame

inv_args(class_name: str, build_name: str, out: bool = True) → None

Сообщение об указании неверных типов аргументов

Параметры

- class_name (str) – Имя класса
- build_name (str) – Имя метода/функции
- out (bool) – Печатать процесс выполнения

Результат

None

Тип результата

None

property is_notebook: bool

Получение результата определения запуска библиотеки в Jupyter или аналогах

Результат**True** если библиотека запущена в Jupyter или аналогах, в обратном случае **False****Тип результата**

bool

libs_vers(out: bool = True) → bool

Получение и отображение версий установленных библиотек

Параметры

out (bool) – Отображение

Результат**True** если версии установленных библиотек отображены, в обратном случае **False****Тип результата**

bool

message_error(message: str, space: int = 0, start: bool = False, end: bool = True, out: bool = True) → None

Сообщение об ошибке

Параметры

- message (str) – Сообщение
- space (int) – Количество пробелов в начале текста
- start (bool) – Начинать сообщение переходом на новую строку
- end (bool) – Заканчивать сообщение переходом на новую строку

- `out (bool)` – Отображение

Результат

None

Тип результата

None

`message_info(message: str, space: int = 0, start: bool = False, end: bool = True, out: bool = True)`
→ None

Информационное сообщение

Параметры

- `message (str)` – Сообщение
- `space (int)` – Количество пробелов в начале текста
- `start (bool)` – Начинать сообщение переходом на новую строку
- `end (bool)` – Заканчивать сообщение переходом на новую строку
- `out (bool)` – Отображение

Результат

None

Тип результата

None

`message_line(message: str, type_message: str = 'info', out: bool = True) → str`

Информационное сообщение (в виде одной строки)

Параметры

- `message (str)` – Сообщение
- `type_message (str)` – Тип сообщения
- `out (bool)` – Отображение

Результат

Информационное сообщение (в виде одной строки)

Тип результата

str

`message_metadata_info(out: bool = True) → None`

Информация об библиотеке

Параметры

- `out (bool)` – Отображение

Результат

None

Тип результата

None

`message_progressbar(message: str = '', space: int = 0, close: bool = False, out: bool = True) → str`

Информационный индикатор выполнения

Параметры

- `message (str)` – Сообщение

- `space (int)` – Количество пробелов в начале текста
- `close (bool)` – Закрыть информационный индикатор
- `out (bool)` – Отображение

Результат

None

Тип результата

str

`message_true(message: str, space: int = 0, start: bool = False, end: bool = True, out: bool = True)`
 → None

Сообщение с положительной информацией

Параметры

- `message (str)` – Сообщение
- `space (int)` – Количество пробелов в начале текста
- `start (bool)` – Начинать сообщение переходом на новую строку
- `end (bool)` – Заканчивать сообщение переходом на новую строку
- `out (bool)` – Отображение

Результат

None

Тип результата

None

3.8 Запись базы данных

`openav.modules.dataset_recording.app.download_processed_video()`

Выгрузка файла записи

GET /download_processed_video

Скачивает файл с записью в формате webm

`openav.modules.dataset_recording.app.download_timing_data()`

Выгрузка файла временных отметок записи

GET /download_processed_video

Скачивает файл временных отметок записи в формате txt

`openav.modules.dataset_recording.app.get_questions()`

Получение списка вопросов

GET /get_questions

Возвращает список вопросов в json формате

`openav.modules.dataset_recording.app.index()`

Отображение основной страницы записи

GET /

Отображает статику страницы записи. Является точкой входа в приложение

`openav.modules.dataset_recording.app.read_questions_from_csv()`

`openav.modules.dataset_recording.app.store_timing_data()`

Сохранение файла временных отметок записи

POST `/store_timing_data`

Сохраняет файл временных отметок записи, который передается в виде json в теле запроса

`openav.modules.dataset_recording.app.upload()`

Сохранение файла записи

POST `/upload`

Сохраняет файл записи, который передается в виде файла видео в теле запроса

3.9 Обработка архивов

```
class openav.modules.file_manager.unzip.UnzipMessages(path_to_logs: str = <property object>,
                                                       lang: str = <property object>)
```

Базовые классы: *Core*

Класс для сообщений

Параметры

- `path_to_logs (str)` – Смотреть *path_to_logs*
- `lang (str)` – Смотреть *lang*

```
class openav.modules.file_manager.unzip.Unzip(path_to_logs: str = <property object>, lang: str =
                                              <property object>)
```

Базовые классы: *UnzipMessages*

Класс для обработки архивов

Параметры

- `path_to_logs (str)` – Смотреть *path_to_logs*
- `lang (str)` – Смотреть *lang*

property `path_to_unzip: str`

Получение директории для разархивирования

Результат

Директория для разархивирования

Тип результата

`str`

```
unzip(path_to_zipfile: str, new_name: str / None = None, force_reload: bool = True, out: bool =
      True) → bool
```

Разархивирование архива

Параметры

- `path_to_zipfile (str)` – Полный путь до архива
- `new_name (str)` – Имя директории для разархивирования
- `force_reload (bool)` – Принудительное разархивирование

- `out (bool)` – Отображение

Результат

True если разархивирование прошло успешно, в обратном случае **False**

Тип результата

`bool`

3.10 Работа с файлами

```
class openav.modules.file_manager.file_manager.FileManagerMessages(path_to_logs: str =
    <property object>, lang:
    str = <property object>)
```

Базовые классы: *Unzip*

Класс для сообщений

Параметры

- `path_to_logs (str)` – Смотреть *path_to_logs*
- `lang (str)` – Смотреть *lang*

```
class openav.modules.file_manager.file_manager.FileManager(path_to_logs: str = <property
    object>, lang: str = <property
    object>)
```

Базовые классы: *FileManagerMessages*

Класс для работы с файлами

Параметры

- `path_to_logs (str)` – Смотреть *path_to_logs*
- `lang (str)` – Смотреть *lang*

```
clear_folder(path_to_folder: str, out: bool = True) → bool
```

Очистка директории

Параметры

- `path_to_folder (str)` – Путь к директории
- `out (bool)` – Печатать процесс выполнения

Результат

True если директория очищена, в обратном случае **False**

Тип результата

`bool`

```
create_folder(path_to_folder: str, out: bool = True) → bool
```

Создание директории

Параметры

- `path_to_folder (str)` – Путь к директории
- `out (bool)` – Печатать процесс выполнения

Результат

True если директория создана, в обратном случае **False**

Тип результата

bool

`get_paths(path: Iterable, depth: int = 1, out: bool = True) → List[str | None]`

Получение поддиректорий

Параметры

- `path (Iterable)` – Путь к директории
- `depth (int)` – Глубина иерархии для извлечения поддиректорий
- `out (bool)` – Отображение

Результат

Список с поддиректориями

Тип результата

List[Optional[str]]

`static re_inv_chars(path: str) → str`

Удаление недопустимых символов из пути

Параметры`path (str)` – Путь**Результат**

Путь

Тип результата

str

`search_file(path_to_file: str, ext: str, create: bool = False, out: bool = True) → bool`

Поиск файла

Параметры

- `path_to_file (str)` – Путь к файлу
- `ext (str)` – Расширение файла
- `create (bool)` – Создание файла в случае его отсутствия
- `out (bool)` – Печатать процесс выполнения

Результат**True** если файл найден, в обратном случае **False****Тип результата**

bool

`search_files(path_to_folder: str, exts: List[str], sort: bool = True, out: bool = True) → List[str | None]`

Поиск файлов в указанной директории

Параметры

- `path_to_folder (str)` – Путь к директории с файлами
- `exts (List[str])` – Расширения файлов
- `sort (bool)` – Сортировать файлы
- `out (bool)` – Печатать процесс выполнения

Результат

список с найденными файлами

Тип результата

List[Optional[str]]

3.11 Загрузка файлов

```
class openav.modules.file_manager.download.DownloadMessages(path_to_logs: str = <property
                                                           object>, lang: str = <property
                                                           object>)
```

Базовые классы: *FileManager*

Класс для сообщений

Параметры

- path_to_logs (*str*) – Смотреть *path_to_logs*
- lang (*str*) – Смотреть *lang*

```
class openav.modules.file_manager.download.Download(path_to_logs: str = <property object>, lang:
                                                    str = <property object>)
```

Базовые классы: *DownloadMessages*

Класс для загрузки файлов

Параметры

- path_to_logs (*str*) – Смотреть *path_to_logs*
- lang (*str*) – Смотреть *lang*

```
download_file_from_url(url: str, force_reload: bool = True, out: bool = True) → int
```

Загрузка файла из URL

Параметры

- url (*str*) – Полный путь к файлу
- force_reload (*bool*) – Принудительная загрузка файла из сети
- out (*bool*) – Отображение

Результат

Код статуса ответа:

- 200 - Файл загружен
- 400 - Ошибка при проверке аргументов
- 404 - Не удалось скачать файл

Тип результата

int

3.12 Работа с JSON

```
class openav.modules.file_manager.json_manager.JsonMessages(path_to_logs: str = <property
                                                             object>, lang: str = <property
                                                             object>)
```

Базовые классы: *Download*

Класс для сообщений

Параметры

- `path_to_logs (str)` – Смотреть *path_to_logs*
- `lang (str)` – Смотреть *lang*

```
class openav.modules.file_manager.json_manager.Json(path_to_logs: str = <property object>, lang:
                                                    str = <property object>)
```

Базовые классы: *JsonMessages*

Класс для работы с JSON

Параметры

- `path_to_logs (str)` – Смотреть *path_to_logs*
- `lang (str)` – Смотреть *lang*

```
load_json(path_to_file: str, create: bool = False, out: bool = True) → Dict[str, bool | int | float | str]
```

Загрузка JSON файла

Параметры

- `path_to_file (str)` – Путь к файлу JSON
- `create (bool)` – Создание файла JSON в случае его отсутствия
- `out (bool)` – Печатать процесс выполнения

Результат

Словарь из json файла

Тип результата

`Dict[str, Union[str, bool, int, float]]`

```
load_json_resources(module: module, path_to_file: str, out: bool = True) → Dict[str, bool | int |
                                                                    float | str]
```

Загрузка JSON файла из ресурсов модуля

Параметры

- `module (ModuleType)` – Модуль
- `path_to_file (str)` – Путь к файлу JSON
- `out (bool)` – Печатать процесс выполнения

Результат

Словарь из json файла

Тип результата

`Dict[str, Union[str, bool, int, float]]`

3.13 Работа с YAML

```
class openav.modules.file_manager.yaml_manager.YamlMessages(path_to_logs: str = <property
                                                             object>, lang: str = <property
                                                             object>)
```

Базовые классы: *Json*

Класс для сообщений

Параметры

- `path_to_logs (str)` – Смотреть *path_to_logs*
- `lang (str)` – Смотреть *lang*

```
class openav.modules.file_manager.yaml_manager.Yaml(path_to_logs: str = <property object>, lang:
                                                    str = <property object>)
```

Базовые классы: *YamlMessages*

Класс для работы с YAML

Параметры

- `path_to_logs (str)` – Смотреть *path_to_logs*
- `lang (str)` – Смотреть *lang*

```
load_yaml(path_to_file: str, create: bool = False, out: bool = True) → Dict[str, bool | int | float | str]
```

Загрузка YAML файла

Параметры

- `path_to_file (str)` – Путь к файлу YAML
- `create (bool)` – Создание файла YAML в случае его отсутствия
- `out (bool)` – Печатать процесс выполнения

Результат

Словарь из yaml файла

Тип результата

`Dict[str, Union[str, bool, int, float]]`

```
load_yaml_resources(module: module, path_to_file: str, out: bool = True) → Dict[str, bool | int |
                                                                    float | str]
```

Загрузка YAML файла из ресурсов модуля

Параметры

- `module (ModuleType)` – Модуль
- `path_to_file (str)` – Путь к файлу YAML
- `out (bool)` – Печатать процесс выполнения

Результат

Словарь из yaml файла

Тип результата

`Dict[str, Union[str, bool, int, float]]`

3.14 Аудио

```
class openav.modules.lab.audio.AudioMessages(path_to_logs: str = <property object>, lang: str =
                                             <property object>)
```

Базовые классы: *Yaml*

Класс для сообщений

Параметры

- `path_to_logs (str)` – Смотреть *path_to_logs*
- `lang (str)` – Смотреть *lang*

```
class openav.modules.lab.audio.Audio(path_to_logs: str = <property object>, lang: str = <property
object>)
```

Базовые классы: *AudioMessages*

Класс для обработки аудиомодальности

Параметры

- `path_to_logs (str)` – Смотреть *path_to_logs*
- `lang (str)` – Смотреть *lang*

```
augmentation(depth: int = 1, crop_px_min: int = 0, crop_px_max: int = 0, crop_percent_min:
float = 0, crop_percent_max: float = 0, flip_lr_probability: float = 0,
flip_ud_probability: float = 0, blur_min: float = 0, blur_max: float = 0,
scale_x_min: float = 0, scale_x_max: float = 0, scale_y_min: float = 0,
scale_y_max: float = 0, rotate_min: int = -90, rotate_max: int = -90,
contrast_min: float = -10.0, contrast_max: float = -10.0, alpha: float = 0, count: int
= 1, clear_diraug: bool = False, out: bool = True) → bool
```

Аугментация аудиовизуальных сигналов

Параметры

- `depth (int)` – Глубина иерархии для получения данных
- `crop_px_min (int)` – Обрезка в пикселях мин
- `crop_px_max (int)` – Обрезка в пикселях макс
- `crop_percent_min (float)` – Обрезка в процентах мин
- `crop_percent_max (float)` – Обрезка в процентах макс
- `flip_lr_probability (float)` – Вероятность отражения по вертикали
- `flip_ud_probability (float)` – Вероятность отражения по горизонтали
- `blur_min (float)` – Размытие мин
- `blur_max (float)` – Размытие макс
- `scale_x_min (float)` – Масштабирование X мин
- `scale_x_max (float)` – Масштабирование X макс
- `scale_y_min (float)` – Масштабирование Y мин
- `scale_y_max (float)` – Масштабирование Y макс
- `rotate_min (int)` – Поворот мин

- `rotate_max (int)` – Поворот макс
- `contrast_min (float)` – Контраст мин
- `contrast_max (float)` – Контраст макс
- `alpha (float)` – Альфа для MixUp
- `count (int)` – Количество применений аугментации
- `clear_diraug (bool)` – Очистка директории для сохранения аугментированных аудиовизуальных сигналов
- `out (bool)` – Отображение

Результат

True если аугментация аудиовизуальных сигналов произведено, в обратном случае **False**

Тип результата

`bool`

Добавлено в версии 0.1.0.

Изменено в версии 0.1.1.

Не рекомендуется, начиная с версии 0.1.0.

```
preprocess_audio(depth: int = 1, sample_rate: int = 16000, n_fft: int = 2048, hop_length: int =
                    512, n_mels: int = 128, power: float = 2.0, pad_mode: str = 'reflect', norm: str
                    = 'slaney', center: bool = True, dpi: int = 1200, color_gradients: str = 'magma',
                    save_raw_data: bool = True, clear_dir_audio: bool = False, out: bool = True)
→ bool
```

Предобработка речевых аудиоданных

Параметры

- `depth (int)` – Глубина иерархии для получения данных
- `sample_rate (int)` – Частота дискретизации
- `n_fft (int)` – Размер параметра FFT
- `hop_length (int)` – Длина перехода между окнами STFT
- `n_mels (int)` – Количество фильтроблоков mel
- `power (float)` – Показатель степени магнитудной спектрограммы
- `pad_mode (str)` – Управление оступами
- `norm (str)` – Коэффициенты треугольных mel-фильтров делятся на ширину соответствующих mel-полос
- `center (bool)` – Отступы с обеих сторон относительно центра аудиодорожки
- `dpi (int)` – DPI
- `color_gradients (str)` – Градиент для спектрограммы
- `save_raw_data (bool)` – Сохранение сырых данных мел-спектрограммы в формате .npy
- `clear_dir_audio (bool)` – Очистка директории для сохранения аудиоданных после предобработки
- `out (bool)` –

Результат

True если предобработка речевых аудиоданных произведено, в обратном случае
False

Тип результата

bool

```
vad(depth: int = 1, type_encode: str = 'crf', crf_value: int = 23, presets_crf_encode: str =  
'medium', sr_input_type: str = 'audio', sampling_rate: int = 16000, threshold: float = 0.56,  
min_speech_duration_ms: int = 250, min_silence_duration_ms: int = 50,  
window_size_samples: int = 1536, speech_pad_ms: int = 150, force_reload: bool = True,  
clear_dirvad: bool = False, out: bool = True) → bool
```

VAD (Voice Activity Detector) или (детектирование голосовой активности)

Параметры

- **depth** (*int*) – Глубина иерархии для получения данных
- **type_encode** (*str*) – Тип кодирования
- **crf_value** (*int*) – Качество кодирования (от **0** до **51**)
- **presets_crf_encode** (*str*) – Скорость кодирования и сжатия
- **sr_input_type** (*str*) – Тип файлов для распознавания речи
- **sampling_rate** (*int*) – Частота дискретизации (**8000** или **16000**)
- **threshold** (*float*) – Порог вероятности речи (от **0.0** до **1.0**)
- **min_speech_duration_ms** (*int*) – Минимальная длительность речевого фрагмента в миллисекундах
- **min_silence_duration_ms** (*int*) – Минимальная длительность тишины в выборках между отдельными речевыми фрагментами
- **window_size_samples** (*int*) – Количество выборок в каждом окне (**512**, **1024**, **1536** для частоты дискретизации **16000** или **256**, **512**, **768** для частоты дискретизации **8000**)
- **speech_pad_ms** (*int*) – Внутренние отступы для итоговых речевых фрагментов
- **force_reload** (*bool*) – Принудительная загрузка модели из сети
- **clear_dirvad** (*bool*) – Очистка директории для сохранения фрагментов аудиовизуального сигнала
- **out** (*bool*) – Отображение

Результат

True если детектирование голосовой активности произведено, в обратном случае
False

Тип результата

bool

Добавлено в версии 0.1.0.

Изменено в версии 0.1.1.

Не рекомендуется, начиная с версии 0.1.0.

`vosk(new_name: str / None = None, force_reload: bool = True, out: bool = True) → bool`

Загрузка и активация модели Vosk для детектирования голосовой активности и распознавания речи

Параметры

- `new_name (str)` – Имя директории для разархивирования
- `force_reload (bool)` – Принудительная загрузка модели из сети
- `out (bool)` –

Результат

True если модель Vosk загружена и активирована, в обратном случае **False**

Тип результата

`bool`

property `vosk_dict_language_sr: str`

Получение/установка размера словаря для распознавания речи

Параметры

`(str)` – Размер словаря

Результат

Размер словаря

Тип результата

`str`

property `vosk_language_sr: str`

Получение/установка языка для распознавания речи

Параметры

`(str)` – Язык

Результат

Язык

Тип результата

`str`

`vosk_sr(depth: int = 1, type_encode: str = 'crf', crf_value: int = 23, presets_crf_encode: str = 'medium', new_name: str / None = None, speech_left_pad_ms: int = 0, speech_right_pad_ms: int = 0, force_reload: bool = True, clear_dirvosk_sr: bool = False, out: bool = True) → bool`

VAD + SR (Voice Activity Detector + Speech Recognition) или (детектирование голосовой активности и распознавание речи)

Параметры

- `depth (int)` – Глубина иерархии для получения данных
- `type_encode (str)` – Тип кодирования
- `crf_value (int)` – Качество кодирования (от **0** до **51**)
- `presets_crf_encode (str)` – Скорость кодирования и сжатия
- `new_name (str)` – Имя директории для разархивирования
- `speech_left_pad_ms (int)` – Внутренний левый отступ для итоговых речевых фрагментов

- `speech_right_pad_ms (int)` – Внутренний правый отступ для итоговых речевых фрагментов
- `force_reload (bool)` – Принудительная загрузка модели из сети
- `clear_dirvosk_sr (bool)` – Очистка директории для сохранения фрагментов аудиовизуального сигнала
- `out (bool)` –

Результат

True если детектирование голосовой активности и распознавание речи произведено, в обратном случае **False**

Тип результата

`bool`

3.15 Видео

```
class openav.modules.lab.video.VideoMessages(path_to_logs: str = <property object>, lang: str = <property object>)
```

Базовые классы: *Json*

Класс для сообщений

Параметры

- `path_to_logs (str)` – Смотреть *path_to_logs*
- `lang (str)` – Смотреть *lang*

```
class openav.modules.lab.video.Video(path_to_logs: str = <property object>, lang: str = <property object>)
```

Базовые классы: *VideoMessages*

Класс для обработки видеомодальности

Параметры

- `path_to_logs (str)` – Смотреть *path_to_logs*
- `lang (str)` – Смотреть *lang*

```
preprocess_video(depth: int = 1, resize: bool = True, resize_resample: str = 'nearest', size_width: int = 112, size_height: int = 112, color_mode: str = 'rgb', dpi: int = 1200, save_raw_data: bool = True, clear_dir_video: bool = False, out: bool = True) → bool
```

Предобработка речевых видеоданных

Параметры

- `depth (int)` – Глубина иерархии для получения данных
- `resize (bool)` – Изменение размера кадра с найденной областью губ
- `resize_resample (str)` – Фильтр для масштабирования
- `size_width (int)` – Ширина области губ
- `size_height (int)` – Высота области губ
- `color_mode (str)` – Цветовая гамма

- `dpi (int)` – DPI
- `save_raw_data (bool)` – Сохранение сырых данных с областями губ в формате .npy
- `clear_dir_video (bool)` – Очистка директории для сохранения видео данных после предобработки
- `out (bool)` –

Результат

True если предобработка речевых видеоданных произведена, в обратном случае **False**

Тип результата

bool

3.16 Мультимодальное объединение аудио- и видеомодальностей

```
class openav.modules.lab.audiovisual.AVMessages(path_to_logs: str = <property object>, lang: str = <property object>)
```

Базовые классы: *Audio*, *Video*

Класс для сообщений

Параметры

- `path_to_logs (str)` – Смотреть *path_to_logs*
- `lang (str)` – Смотреть *lang*

```
class openav.modules.lab.audiovisual.AV(path_to_logs: str = <property object>, lang: str = <property object>)
```

Базовые классы: *AVMessages*

Класс для мультимодального объединения аудио- и видеомодальностей

Параметры

- `path_to_logs (str)` – Смотреть *path_to_logs*
- `lang (str)` – Смотреть *lang*

```
test_audiovisual(subfolders: Dict[str, str], n_classes: int, classes: List[str], encoder_decoder: int,
                 max_segment: int, hidden_units: int, hidden_features: int, input_dim: int,
                 shape_audio: Dict[str, int], shape_video: Dict[str, int], save_confusion_matrix:
                 bool, path_to_save_confusion_matrix: str, figsize_confusion_matrix: Dict[str,
                 int], path_to_model: str, out: bool = True) → bool
```

Автоматическое тестирование на аудиовизуальных данных

Параметры

- `subfolders (Dict [str, str])` – Словарь с подкаталогами с данными
- `n_classes (int)` – Количество классов
- `classes (List [str])` – Список классов
- `encoder_decoder (int)` – Количество энкодеров и декодеров

- `max_segment (int)` – Максимальная длительность сегмента видео
- `hidden_units (int)` – Количество скрытых нейронов
- `hidden_features (int)` – Количество скрытых признаков
- `input_dim (int)` – Количество входных признаков
- `shape_audio (Dict[str, int])` – Входная размерность аудио лог-мел спектрограммы
- `shape_video (Dict[str, int])` – Входная размерность видеок кадров
- `save_confusion_matrix (bool)` – Сохранение матрицы спутывания
- `path_to_save_confusion_matrix (str)` – Путь к директории для сохранения матрицы спутывания
- `figsize_confusion_matrix (Dict[str, int])` – Настройки для формирования изображения матрицы спутывания
- `path_to_model (str)` – Путь к нейросетевой аудиовизуальной модели
- `out (bool)` –

Результат

True если автоматическое тестирование на аудиовизуальных данных произведено, в обратном случае **False**

Тип результата

bool

```
train_audiovisual(subfolders: Dict[str, str], n_classes: int, classes: List[str], encoder_decoder: int,
                  batch_size: int, max_segment: int, patience: int, epochs: int, seed: int,
                  leaning_rate: float, weight_decay: float, optimizer: str, hidden_units: int,
                  hidden_features: int, input_dim: int, shape_audio: Dict[str, int], shape_video:
                  Dict[str, int], path_to_model_fa: str, path_to_model_fv: str, requires_grad:
                  str, path_to_save_models: str, out: bool = True) → bool
```

Автоматическое обучение на аудиовизуальных данных

Параметры

- `subfolders (Dict[str, str])` – Словарь с подкаталогами с данными
- `n_classes (int)` – Количество классов
- `classes (List[str])` – Список классов
- `encoder_decoder (int)` – Количество энкодеров и декодеров
- `batch_size (int)` – Размер батча
- `max_segment (int)` – Максимальная длительность сегмента видео
- `patience (int)` – Количество неудачных эпох
- `epochs (int)` – Количество эпох
- `seed (int)` – Начальное состояние обучения
- `leaning_rate (float)` – Скорость обучения
- `weight_decay (float)` – Скорость обучения
- `optimizer (str)` – Оптимизатор
- `hidden_units (int)` – Количество скрытых нейронов

- `hidden_features (int)` – Количество скрытых признаков
- `input_dim (int)` – Количество входных признаков
- `shape_audio (Dict [str, int])` – Входная размерность аудио лог-мел спектрограммы
- `shape_video (Dict [str, int])` – Входная размерность видеокадров
- `path_to_model_fa (str)` – Путь к нейросетевой модели (аудио)
- `path_to_model_fv (str)` – Путь к нейросетевой модели (видео)
- `path_to_save_models (str)` – Путь к директории для сохранения моделей
- `requires_grad (str)` – Заморозка слоев для извлечения ауди и видео признаков
- `out (bool)` –

Результат

True если автоматическое обучение на аудиовизуальных данных произведено, в обратном случае **False**

Тип результата

`bool`

3.17 Сборка

```
class openav.modules.lab.build.Run(path_to_logs: str = <property object>, lang: str = <property object>)
```

Базовые классы: *AV*

Класс для сборки библиотеки

Параметры

- `path_to_logs (str)` – Смотреть *path_to_logs*
- `lang (str)` – Смотреть *lang*

Работа с Shell

Документация для файла `modules/trml/shell.py`

Пользовательские исключения

Документация для файла `modules/core/exceptions.py`

Логирование

Документация для файла `modules/core/logging.py`

Определение языка

Документация для файла `modules/core/language.py`

Сообщения

Документация для файла `modules/core/messages.py`

Настройки

Документация для файла `modules/core/settings.py`

Ядро

Документация для файла `modules/core/core.py`

Запись базы данных

Документация для файла `modules/dataset_recording/app.py`

Обработка архивов

Документация для файла `modules/file_manager/unzip.py`

Работа с файлами

Документация для файла `modules/file_manager/file_manager.py`

Загрузка файлов

Документация для файла `modules/file_manager/download.py`

Работа с JSON

Документация для файла `modules/file_manager/json_manager.py`

Работа с YAML

Документация для файла `modules/file_manager/yaml_manager.py`

Аудио

Документация для файла `modules/lab/audio.py`

Видео

Документация для файла `modules/lab/video.py`

Мультимодальное объединение аудио- и видеомодальностей

Документация для файла `modules/lab/audiovisual.py`

Сборка

Документация для файла `modules/lab/build.py`

Диаграмма классов

SVG диаграмм классов ...

Набор аудиовизуальных данных

В текущем разделе представлены примеры аннотированных аудиовизуальных данных из набора данных, с которыми выполнялись экспериментальные исследования. Весь набор данных можно загрузить по следующей ссылке:

Полный набор данных

Примеры фраз из набора данных

Фраза «Выключить компьютер»

Фраза «Калибровка»

Фраза «Отпустить правую»

Фраза «Двойной клик»

Фраза «Выделить все»

Команда разработчиков

Библиотека OpenAV разработана и поддерживается исследовательской группой из лаборатории речевых и мультимодальных интерфейсов (ЛРМИ) Санкт-Петербургского Федерального исследовательского центра Российской академии наук (СПб ФИЦ РАН):



Иванько Денис

Руководитель проекта. Старший научный сотрудник Федерального государственного бюджетного учреждения науки «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН)



Рюмин Дмитрий

Главный разработчик программного обеспечения. Старший научный сотрудник Федерального государственного бюджетного учреждения науки «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН)



Маркитантов Максим

Программист, разработчик программного обеспечения. Младший научный сотрудник Федерального государственного бюджетного учреждения науки «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН)

Шилов Николай

Разработчик алгоритмов и методов обработки видеоданных. Старший научный сотрудник Федерального государственного бюджетного учреждения науки «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН)



Карпов Алексей

Разработчик алгоритмов и методов обработки акустических данных, а также методов объединения аудиовизуальных данных. Главный научный сотрудник Федерального государственного бюджетного учреждения науки «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН)

Ответы на часто задаваемые вопросы

Это список часто задаваемых вопросов о *OpenAV* и ответов на них.

OpenAV это?

Ответ . . .

Изменения версий

Предупреждение: Changelog was not built because `sphinx_github_changelog_token` parameter is missing in the documentation configuration.

Совет: Find the project changelog [here](#).

O

`openav.modules.core.exceptions`, [37](#)

`openav.modules.dataset_recording.app`, [47](#)

A

`add_line()` (статический метод `openav.modules.trml.shell.Shell`), 37
`Audio` (класс в `openav.modules.lab.audio`), 54
`AudioMessages` (класс в `openav.modules.lab.audio`), 54
`augmentation()` (метод `openav.modules.lab.audio.Audio`), 54
`AV` (класс в `openav.modules.lab.audiovisual`), 59
`AVMessages` (класс в `openav.modules.lab.audiovisual`), 59

B

`BlurError`, 37
`build_args()` (метод `openav.modules.core.core.Core`), 44

C

`check_create_logger` (`openav.modules.core.logging.Logging` property), 39
`chunk_size` (`openav.modules.core.settings.Settings` property), 41
`clear()` (статический метод `openav.modules.trml.shell.Shell`), 37
`clear_folder()` (метод `openav.modules.file_manager.file_manager.FileManager`), 49
`clear_line` (`openav.modules.core.settings.Settings` property), 41
`clear_shell()` (метод `openav.modules.core.core.Core`), 44
`color_blue` (`openav.modules.core.settings.Settings` property), 41
`color_green` (`openav.modules.core.settings.Settings` property), 42
`color_red` (`openav.modules.core.settings.Settings` property), 42
`ContrastError`, 37

`Core` (класс в `openav.modules.core.core`), 44

`CoreMessages` (класс в `openav.modules.core.core`), 44

`create_folder()` (метод `openav.modules.file_manager.file_manager.FileManager`), 49

`CropPercentsError`, 37

`CropPXError`, 37

`CustomException`, 38

D

`df_pkgs` (`openav.modules.core.core.Core` property), 45

`dir_va_names` (`openav.modules.core.settings.Settings` property), 42

`Download` (класс в `openav.modules.file_manager.download`), 51

`download_file_from_url()` (метод `openav.modules.file_manager.download.Download`), 51

`download_processed_video()` (в модуле `openav.modules.dataset_recording.app`), 47

`download_timing_data()` (в модуле `openav.modules.dataset_recording.app`), 47

`DownloadMessages` (класс в `openav.modules.file_manager.download`), 51

E

`ext_search_files` (`openav.modules.core.settings.Settings` property), 42

F

`FileManager` (класс в `openav.modules.file_manager.file_manager`), 49

FileManagerMessages (класс в locales (openav.modules.core.language.Language
 openav.modules.file_manager.file_manager), property), 40
 49
 FlipLRProbabilityError, 38
 FlipUDProbabilityError, 38
 Logging (класс в openav.modules.core.logging), 39
 logger_gui (openav.modules.core.logging.Logging
 property), 39
G **M**
 get_paths() (метод openav.modules.file_manager.file_manager.FileManager), 45
 50
 get_questions() (в модуле openav.modules.core.core.Core), 46
 openav.modules.dataset_recording.app), message_info() (метод
 47 message_line() (метод
 openav.modules.core.core.Core), 46
 message_metadata_info() (метод
 openav.modules.core.core.Core), 46
 message_progressbar() (метод
 openav.modules.core.core.Core), 46
 message_true() (метод
 openav.modules.core.core.Core), 47
 ignore_dirs (openav.modules.core.settings.Settings
 property), 42
 index() (в модуле openav.modules.dataset_recording.app), 47
 inv_args() (метод openav.modules.core.core.Core), 45
 Messages (класс в openav.modules.core.messages), 41
 InvalidContentLength, 38
 MixUpAlphaError, 38
 is_notebook (openav.modules.core.core.Core
 property), 45
 IsNestedCatalogsNotFound, 38
 IsNestedDirectoryANotFound, 38
 IsNestedDirectoryVNotFound, 38
J **O**
 Json (класс в openav.modules.file_manager.json_manager), 52
 path_to_dataset (openav.modules.core.settings.Settings
 property), 42
 JsonMessages (класс в openav.modules.file_manager.json_manager), 52
 path_to_dataset_vad
 (openav.modules.core.settings.Settings
 property), 43
 path_to_dataset_vosk_sr
 (openav.modules.core.settings.Settings
 property), 43
 path_to_locales (openav.modules.core.language.Language
 property), 40
 path_to_logs (openav.modules.core.logging.Logging
 property), 39
 path_to_save_models
 (openav.modules.core.settings.Settings
 property), 43
 path_to_unzip (openav.modules.file_manager.unzip.Unzip
 property), 48
 preprocess_audio() (метод
 openav.modules.lab.audio.Audio), 55
 preprocess_video() (метод
 openav.modules.lab.video.Video), 58
 lang (openav.modules.core.language.Language
 property), 40
 Language (класс в openav.modules.core.language), 40
 libs_vers() (метод openav.modules.core.core.Core), 45
 load_json() (метод openav.modules.file_manager.json_manager.Json), 52
 load_json_resources() (метод openav.modules.file_manager.json_manager.Json), 52
 load_yaml() (метод openav.modules.file_manager.yaml_manager.Yaml), 53
 load_yaml_resources() (метод openav.modules.file_manager.yaml_manager.Yaml), 53
 PresetCFREncodeVideoError, 38

R

`re_inv_chars()` (статический метод `openav.modules.core.logging.Logging`), 40

`re_inv_chars()` (статический метод `openav.modules.file_manager.file_manager.FileManager`), 50

`read_questions_from_csv()` (в модуле `openav.modules.dataset_recording.app`), 47

`RotateError`, 38

`Run` (класс в `openav.modules.lab.build`), 61

S

`SamplingRateError`, 38

`ScaleError`, 39

`search_file()` (метод `openav.modules.file_manager.file_manager.FileManager`), 50

`search_files()` (метод `openav.modules.file_manager.file_manager.FileManager`), 50

`Settings` (класс в `openav.modules.core.settings`), 41

`Shell` (класс в `openav.modules.trml.shell`), 37

`SRInputTypeError`, 38

`store_timing_data()` (в модуле `openav.modules.dataset_recording.app`), 48

T

`test_audiovisual()` (метод `openav.modules.lab.audiovisual.AV`), 59

`text_bold` (`openav.modules.core.settings.Settings` свойство), 43

`text_end` (`openav.modules.core.settings.Settings` свойство), 43

`train_audiovisual()` (метод `openav.modules.lab.audiovisual.AV`), 60

`TypeEncodeVideoError`, 39

`TypeMessagesError`, 39

U

`Unzip` (класс в `openav.modules.file_manager.unzip`), 48

`unzip()` (метод `openav.modules.file_manager.unzip.Unzip`), 48

`UnzipMessages` (класс в `openav.modules.file_manager.unzip`), 48

`upload()` (в модуле `openav.modules.dataset_recording.app`), 48

V

`vad()` (метод `openav.modules.lab.audio.Audio`), 56

`Video` (класс в `openav.modules.lab.video`), 58

`VideoMessages` (класс в `openav.modules.lab.video`), 58

`Vosk()` (метод `openav.modules.lab.audio.Audio`), 56

`vosk_dict_language_sr` (`openav.modules.lab.audio.Audio` свойство), 57

`vosk_language_sr` (`openav.modules.lab.audio.Audio` свойство), 57

`vosk_sr()` (метод `openav.modules.lab.audio.Audio`), 57

W

`WindowSizeSamplesError`, 39

Y

`YamlManager`,

`Yaml` (класс в `openav.modules.file_manager.yaml_manager`), 53

`YamlMessages` (класс в `openav.modules.file_manager.yaml_manager`), 53

модуль

`openav.modules.core.exceptions`, 37

`openav.modules.dataset_recording.app`, 47