

СОГЛАСОВАНО
Научный руководитель ИЦ СИИП
Университета ИТМО

_____ А. В. Бухановский
_____ 2023 г.

УТВЕРЖДАЮ
Директор СПб ФИЦ РАН

_____ А. Л. Ронжин
_____ 2023 г.

**БИБЛИОТЕКА АЛГОРИТМОВ СИЛЬНОГО ИИ
ДЛЯ ЗАДАЧ ПРОМЫШЛЕННОСТИ**

**КОМПОНЕНТ БИБЛИОТЕКИ АЛГОРИТМОВ СИЛЬНОГО ИИ В ЧАСТИ
АЛГОРИТМОВ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ПОВЕДЕНИЯ ЧЕЛОВЕКА
НА ОСНОВЕ ЕГО МУЛЬТИМОДАЛЬНЫХ ДАННЫХ, ОБЕСПЕЧИВАЮЩИХ
ОЦЕНИВАНИЕ УРОВНЯ ОТДЕЛЬНЫХ ПЕРСОНАЛЬНЫХ КАЧЕСТВ
ЛИЧНОСТИ ЧЕЛОВЕКА (OCEAN-AI)**

ОПИСАНИЕ ПРОГРАММЫ

ЛИСТ УТВЕРЖДЕНИЯ

RU.CHAБ. 00853-01 13 11-ЛУ

Представители
Организации-разработчика

Руководитель разработки

_____ А. А. Карпов
_____ 2023 г.

Нормоконтролер

_____ Н. А. Александрова
_____ 2023 г.

2023

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

УТВЕРЖДЕН
RU.СНАБ. 00853-01 13 11-ЛУ

**КОМПОНЕНТ БИБЛИОТЕКИ АЛГОРИТМОВ СИЛЬНОГО ИИ В ЧАСТИ
АЛГОРИТМОВ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ПОВЕДЕНИЯ ЧЕЛОВЕКА
НА ОСНОВЕ ЕГО МУЛЬТИМОДАЛЬНЫХ ДАННЫХ, ОБЕСПЕЧИВАЮЩИХ
ОЦЕНИВАНИЕ УРОВНЯ ОТДЕЛЬНЫХ ПЕРСОНАЛЬНЫХ КАЧЕСТВ
ЛИЧНОСТИ ЧЕЛОВЕКА (ОСЕАН-АІ)**

ОПИСАНИЕ ПРОГРАММЫ

RU.СНАБ. 00853-01 13 11

ЛИСТОВ 44

Подп. и дата	
Инв. №	
Взам. и	
Подп. и дата	
Инв. №	

2023

АННОТАЦИЯ

Документ содержит описание компонента библиотеки алгоритмов сильного ИИ RU.СНАБ. 00853-01 13 11 в части алгоритмов интеллектуального анализа поведения человека на основе его мультимодальных данных, обеспечивающих оценивание уровня отдельных персональных качеств личности человека (OCEAN-AI). Аббревиатура OCEAN означает пять оцениваемых компонентом OCEAN-AI персональных качеств личности человека (ПКЛЧ): «Открытость опыту» (от англ. Openness to experience/Openness), «Добросовестность» (от англ. Conscientiousness), «Экстраверсия» (от англ. Extraversion), «Доброжелательность» (от англ. Agreeableness), «Эмоциональная стабильность» (от англ. Non-Neuroticism); AI – искусственный интеллект (от англ. Artificial Intelligence). Компонент OCEAN-AI входит в состав библиотеки алгоритмов сильного ИИ и разработан в рамках плана Исследовательского центра в сфере искусственного интеллекта «Сильный ИИ в промышленности» (ИЦ ИИ) в рамках соглашения с АНО «Аналитический центр при Правительстве Российской Федерации» (ИГК 000000D730321P5Q0002), № 70–2021–00141, с целью создания одномодальных и мультимодальных систем, построенных на различных принципах (экспертных и нейросетевых), т.е. основанных на композитных моделях, для предсказания отдельных ПКЛЧ для решения следующих задач:

- 1) упрощения дистанционного отбора потенциальных кандидатов на профессии;
- 2) прогнозирования предпочтений по выбору промышленных потребительских товаров;
- 3) формирования эффективных рабочих коллективов.

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ	4
2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ	4
3. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ	6
4. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА	20
5. ВЫЗОВ И ЗАГРУЗКА	21
6. ВХОДНЫЕ ДАННЫЕ	38
7. ВЫХОДНЫЕ ДАННЫЕ	39
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ	41
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	42
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ	44

1. ОБЩИЕ СВЕДЕНИЯ

Компонент OCEAN-AI библиотеки алгоритмов сильного ИИ RU.СНАБ.00853-01 13 11 разработан в соответствии с мероприятием М1 «Выполнение опережающих исследований и разработок в области алгоритмов сильного ИИ» (1.1.6-1.1.7 Разработка и испытания экспериментального образца библиотеки алгоритмов сильного ИИ в части интеллектуальной детекции событий и смыслов на основе мультимодальных данных (текст, звук, изображения в разных диапазонах) для задач распознавания режимов работы и неисправностей оборудования) программы ИЦ ИИ «Сильный ИИ в промышленности» в рамках федерального проекта «Искусственный интеллект». OCEAN-AI предназначен для мультимодального анализа группы людей с целью оценивания их персональных качеств личности: «Открытость опыту» (от англ. Openness to experience/Openness), «Добросовестность» (от англ. Conscientiousness), «Экстраверсия» (от англ. Extraversion), «Доброжелательность» (от англ. Agreeableness), «Эмоциональная стабильность» (от англ. Non-Neuroticism). Аббревиатура OCEAN означает пять оцениваемых персональных качеств, AI – искусственный интеллект (от англ. Artificial Intelligence).

Компонент OCEAN-AI разработан на языке Python в средах разработки Microsoft Visual Studio 2022/PyCharm 2020.2.3/Jupyter Lab 3.4.5 с использованием библиотек: ipython 8.18.1, jupyterlab 3.4.5, tensorflow 2.15.0, keras 2.15.0, Keras-Applications 1.0.8, numpy 1.26.2, scipy 1.11.4, pandas 2.1.3, requests 2.31.0, opensmile 2.5.0, librosa 0.10.1, audioread 3.0.1, scikit-learn 1.3.2, opencv-contrib-python 4.8.1, pymediainfo 6.1.0, mediapipe 0.9.0, liwc 0.5.0, transformers 4.36.0, sentencepiece 0.1.99, torch 2.0.1+cpu, torchaudio 2.0.2+cpu. Допускается использование более новых версий библиотек при условии сохранения текущего функционала. Компонент OCEAN-AI размещен по адресу <https://gitlab.actcognitive.org/itmo-sai-code/oceanai>. Для его использования необходим компилятор Python 3.9/3.10.

2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

2.1. Назначение компонента

Компонент OCEAN-AI реализует функции сильного ИИ в части комплексирования акустических, визуальных и текстовых признаков, построенных на различных принципах (экспертных и нейросетевых), т.е. компонент OCEAN-AI реализует подходы композитного (гибридного) ИИ для анализа аудиовизуальной и текстовой информации с последующей оценкой ПКЛЧ.

2.2. Классы решаемых задач

Классы решаемых задач компонентом OCEAN-AI:

1. Предобработка аудиоинформации, вычисление акустических признаков и выдача по ним гипотез предсказаний оценок ПКЛЧ.
2. Предобработка видеоинформации, вычисление визуальных признаков и выдача по ним гипотез предсказаний оценок ПКЛЧ.
3. Предобработка текстовой информации, вычисление визуальных признаков и выдача по ним гипотез предсказаний оценок ПКЛЧ.
4. Выполнение мультимодального объединения признаков и получение мультимодальных гипотез предсказаний оценок ПКЛЧ для аудио-, видео- и текстовой информации.

2.3. Функциональные ограничения на применение

Компонент OCEAN-AI имеет следующие функциональные ограничения на применение:

- мультимодальные файлы должны храниться в файлах формата «avi», «mp4» или «mov»;
- текстовая информация с экспертной транскрипцией должна храниться в файлах формата «txt»;
- файлы, содержащие записи одного человека, должны иметь одинаковые имена без учета расширения файлов (т.е. должны быть одноименными);
- продолжительность мультимодальных файлов неограниченна, при этом оптимальная продолжительность для мультимодальных файлов на английском языке составляет 15 секунд, на русском языке – 2,5 секунды;
- площадь области лица в кадре должна составлять не менее 10% от площади кадра;
- количество человек в кадре – 1;
- рекомендованное расположение человека в кадре по пояс (голова и верхняя часть туловища);
- частота кадров в секунду для мультимодальных файлов должна быть не менее 6 и не более 30;
- высота видеокадров должна быть не менее 682 и не более 2160 пикселей (оптимальная высота видеокадра 720 пикселей);
- ширина видеокадров должна быть не менее 406 и не более 3840 пикселей (оптимальная ширина видеокадра 1280 пикселей);

- возраст человека должен быть не менее 16 и не более 60 лет.

Компонент OCEAN-AI функционирует в среде выполнения программного кода Jupyter Lab 3.4.5 и компиляторе Python 3.9/3.10.

2.4. Область применения

Компонент OCEAN-AI предоставляет оценки ПКЛЧ, которые могут быть применимыми в различных прикладных областях, связанных с промышленностью:

1. HR-процессы. С помощью оценок ПКЛЧ возможно ранжирование потенциальных кандидатов для выполнения профессиональных обязанностей для упрощения отбора высококвалифицированных кадров, а также формирование эффективных рабочих коллективов.

2. Таргетированный маркетинг. Люди со схожими персональными качествами и психотипом личности могут иметь схожие предпочтения, следовательно, возможно моделировать намерения покупателей при выборе определенных промышленных потребительских товаров на основе оценок их ПКЛЧ.

Также оценки ПКЛЧ могут быть применимы и в других прикладных областях: образование, медицина и здравоохранение, сфера развлечений, интеллектуальные персональные ассистенты (умные гаджеты, умные комнаты интеллектуальные пространства и т.д.).

3. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

3.1. Используемые методы

В состав компонента OCEAN-AI входят следующие алгоритмы:

1. Алгоритм анализа аудиоинформации (ААИ) для оценивания ПКЛЧ по голосовым характеристикам использует:

- общедоступные методы загрузки и считывания аудиоинформации и ее представления в log-mel спектрограммы [1], реализованные в библиотеке Librosa [2];
- общедоступный метод представления аудиоинформации в набор экспертных признаков библиотеки openSMILE [3];
- разработанный метод получения гипотез предсказаний оценок ПКЛЧ по экспертным признакам, основанный на рекуррентной (Long Short-Term Memory – LSTM [4]), полносвязной (Fully Connected Layer [5]) нейросетевых моделях;

- разработанный метод [6, 7] получения гипотез предсказаний оценок ПКЛЧ по нейросетевым признакам, основанный на log-mel спектрограммах, сверточной (VGG-16 [8]) и полносвязной нейросетевых моделях;

- разработанный метод комплексирования акустических признаков, построенных на различных принципах (экспертных и нейросетевых), основанный на полносвязной нейросетевой модели.

2. Алгоритм анализа видеоинформации (АВИ) для оценивания ПКЛЧ по лицевым характеристикам использует:

- общедоступные методы загрузки, считывания и обработки видеоинформации, реализованные в библиотеки OpenCV [9];

- общедоступный метод обнаружения двумерных координат (x, y) ключевых точек лица, основанный на сверточной (Face Mesh) нейросетевой модели библиотеки Mediapipe [10];

- разработанный метод [11] обнаружения областей лиц на основе сверточной (Yolo [12]) нейросетевой модели;

- разработанный метод [13] извлечения признаков на основе сверточной (ResNet-50 [14]) нейросетевой модели;

- разработанный метод извлечения экспертных признаков на основе двумерных координат (x, y) ключевых точек лица;

- разработанный метод получения гипотез предсказаний оценок ПКЛЧ по экспертным признакам, основанный на рекуррентной (LSTM) и полносвязной нейросетевых моделях;

- разработанный метод [15] получения гипотез предсказаний оценок ПКЛЧ по нейросетевым признакам, основанный на рекуррентной (LSTM) и полносвязной нейросетевых моделях;

- разработанный метод комплексирования визуальных признаков, построенных на различных принципах (экспертных и нейросетевых), основанный на полносвязной нейросетевой модели.

3. Алгоритм анализа тестовой информации (АТИ) для оценивания ПКЛЧ по текстовым характеристикам разговорной речи использует:

- общедоступный метод автоматического распознавания речи с целью извлечения текстовой информации (орфографических транскрипций речи) на основе модели трансформер (Whisper [16]), реализованный в библиотеки Transformers [17];

- общедоступный метод нейростевого векторного представления слов на основе модели трансформер (BERT [18]), реализованный в библиотеки Transformers [17];
- разработанный метод получения гипотез предсказаний оценок ПКЛЧ по экспертным признакам лингвистического подсчета статистики слов (Linguistic Inquiry and Word Count – LIWC [19]), основанный на рекуррентной (двунаправленной LSTM [20]) и полносвязной нейросетевой моделях, а также механизме самовнимания (Self-Attention [21]);
- разработанный метод получения гипотез предсказаний оценок ПКЛЧ по нейросетевым признакам, основанный на рекуррентной (двунаправленной LSTM) и полносвязной нейросетевой моделях, а также механизме самовнимания;
- разработанный метод комплексирования текстовых признаков, построенных на различных принципах (экспертных и нейросетевых), основанный на полносвязной нейросетевой модели.

4. Алгоритм мультимодального объединения информации (МОИ) для оценивания ПКЛЧ по голосовым, лицевым и текстовым характеристикам использует:

- разработанный метод объединения признаков, полученных из трех алгоритмов (ААИ, АВИ и АТИ), основанный на управляемом кросс-модальном механизме внимания [22] и мультимодальном взвешивании [23].

Компонент OCEAN-AI позволяет обрабатывать мультимодальные файлы, содержащие только аудио-, видео- или текстовую информацию, а также все три источника информации. Компонент OCEAN-AI не анализирует статические изображения.

Первый алгоритм ААИ предназначен для оценивания уровня ПКЛЧ по акустическим признакам разговорной речи человека. В алгоритме осуществляется необходимая предобработка аудиоинформации, вычисление акустических признаков и выдача по ним гипотез предсказаний оценок ПКЛЧ.

Алгоритм ААИ реализует функции сильного ИИ в части комплексирования акустических признаков, построенных на различных принципах (экспертных и нейросетевых), т.е. алгоритм реализует функции композитного (гибридного) ИИ для анализа аудиоинформации.

Алгоритм ААИ основывается на следующих математических соотношениях. На вход алгоритма поступает множество мультимодальных файлов $B = \{b_1, \dots, b_n\}$, где n – количество одноименных файлов в базе данных. Алгоритм ААИ работает с каждым аудиофайлом последовательно. С помощью открытой библиотеки Librosa [2] считывается аудиоинформация b_i^A из каждого файла b_i , где $i = 1, \dots, n$. Аудиоинформация из каждого

файла разделяется на короткие аудиофрагменты, представляющие множество $S_i^A = \{s_1^A, \dots, s_m^A\}$, где m – количество аудиофрагментов, которое может быть различно для каждого аудиофайла и зависит от его общей длительности. Разделение сигнала на перекрывающиеся аудиофрагменты s_j^A , где $j = 1, \dots, m$, выполняется скользящим окном постоянной длительности две секунды. Затем из аудиофрагмента одновременно вычисляются акустические экспертные (англ. hand-crafted) и нейросетевые признаки речи. Экспертные признаки извлекаются с помощью библиотеки openSMILE [3], в которой предусмотрено несколько наборов акустических признаков. В компоненте OCEAN-AI используется набор акустических признаков eGeMAPSv02. Этот набор признаков характеризует аффективно-физиологические изменения в речи и содержит 25 дескрипторов низкого уровня, которые извлекаются каждые 20 мс. Экспертные признаки для каждого аудиофрагмента составляют матрицу признаков размерностью $R \times G$, где R – количество векторов признаков, G – количество экспертных признаков в векторе. Полученные матрицы признаков подаются на рекуррентную нейросеть (LSTM [4]) и полносвязный нейросетевой слой [5] из пяти нейронов с линейной функцией активации [24]. На выходе нейросетевой модели вычисляются гипотезы предсказаний для пяти отдельных ПКЛЧ. Для каждого аудиофрагмента рассчитывается матрица гипотез предсказаний размерностью 1×5 . В качестве нейросетевого экстрактора признаков используется сверточная нейросеть (VGG-16 [8]), которая извлекает вектор признаков размерностью C из каждого аудиофрагмента, C – количество нейросетевых признаков в векторе. При этом предварительно каждый аудиофрагмент представляется в виде изображения log-mel спектрограммы, которое показывает развертку спектральной плотности мощности сигнала во времени.

В работе [6] предложено несколько методов обработки паралингвистических явлений в речи. ПКЛЧ также относятся к паралингвистическим явлениям. В связи с тесной связью паралингвистических явлений между собой (пол, эмоциональные состояния, правдивость речи и персональные качества личности), предложенный в работе метод извлечения акустических признаков речевого сигнала на основе log-mel спектрограммы используется в компоненте OCEAN-AI.

Log-mel спектрограммы имеют размерность $D \times E \times 1$, где D – количество mel полосовых фильтров, значение E – длительность аудиофрагмента, значение 1 – количество каналов изображения (градации серого в данном случае). Перед подачей изображения мел-спектрограммы на сверточную нейросеть для извлечения нейросетевых признаков исходное изображение размерностью $D \times E \times 1$ преобразуется в изображение размерностью $H \times H \times 3$ путем дублирования каналов и масштабирования исходного

изображения до нужного размера, необходимого для входа сверточной нейросети. Каждое изображение размерностью $H \times H \times 3$ после прохождения ряда сверточных слоев преобразуется в вектор признаков размерностью C . Затем вектор признаков подается на полносвязную нейросеть для извлечения гипотез предсказаний. Нейросеть состоит из пяти нейронов и имеет линейную функцию активации. В работе [7] представлен метод извлечения нейросетевых признаков и получения гипотез предсказаний по ним на базе log-mel спектрограмм и VGG-16 для оценивания ПКЛЧ.

Выходными данными алгоритма ААИ являются несколько матриц. Для каждого аудиофрагмента извлекаются матрицы среднеуровневых признаков, построенных для низкоуровневых экспертных признаков $HC_i^A \in \mathbb{R}^{m \times F_1^A}$ и нейросетевых признаков $NN_i^A \in \mathbb{R}^{m \times F_2^A}$ (F_1^A и F_2^A векторы акустических среднеуровневых признаков) и матрица с объединенными гипотезами предсказаний $P_i^A \in \mathbb{R}^{m \times 10}$. Для матриц HC_i^A и NN_i^A извлекаются статистические функционалы (средние значения и стандартные отклонения), которые затем используются для алгоритма МОИ. Матрица P_i^A используется для одномодального взвешивания гипотез предсказаний, полученных от среднеуровневых акустических признаков, с помощью пяти однослойных полносвязных нейросетей с сигмоидной функцией активации [25].

Низкоуровневые признаки – это признаки общего назначения, полученные из необработанной аудио-, видео- и текстовой информации о человеке, используемые для решения разных задач анализа человека. В нашем случае это экспертные и нейросетевые признаки. Среднеуровневые признаки – это специальные признаки для решения конкретной задачи анализа человека. Они строятся на основе низкоуровневых признаков и направлены на моделирование ПКЛЧ. Высокоуровневые признаки – это гипотезы предсказаний оценок ПКЛЧ, созданные на основе среднеуровневых признаков и используемые для их интерпретации при решении практических задач. Гипотезы предсказаний вычисляются посредством полносвязных нейросетевых моделей, которые имеют веса и значения смещения, применяемые к входным среднеуровневым признакам. Полученные значения нормализуются с помощью сигмоидальной функции активации в диапазоне значений от 0 до 1, что соответствует оценкам ПКЛЧ.

Второй алгоритм АВИ предназначен для оценивания уровня ПКЛЧ по визуальным признакам мимики и лица человека. В алгоритме осуществляется необходимая предобработка видеoinформации, вычисление визуальных признаков и выдача по ним гипотез предсказаний оценок ПКЛЧ.

Алгоритм АВИ реализует функции сильного ИИ в части комплексирования визуальных признаков, построенных на различных принципах (экспертных и нейросетевых), т.е. алгоритм реализует функции композитного (гибридного) ИИ для анализа видеоинформации.

Алгоритм АВИ основывается на следующих математических соотношениях. На вход алгоритма поступает множество мультимодальных файлов $B = \{b_1, \dots, b_n\}$, содержащих n одноименных файлов. Алгоритм АВИ работает с каждым видеофайлом последовательно. С помощью открытой библиотеки OpenCV [9] считывается видеоинформация b_i^V из каждого файла b_i , где $i = 1, \dots, n$. Выполняется равномерное прореживание кадров, т.е. понижение частоты кадров в секунду, это необходимо сделать для соблюдения одинаковых условий для всех видеофайлов, поскольку частота кадров в них может быть различна. Для каждого кадра извлекаются визуальные экспертные и нейросетевые признаки. Векторы экспертных признаков строятся на основе 468 двумерных координат (x, y) ключевых точек лица, обнаруженных в кадре с помощью библиотеки Mediapipe [10].

Графические области лиц, необходимые для вычисления нейросетевых признаков, извлекаются с помощью метода, предложенного в работе [11]. Метод реализовывался с использованием динамических изображений лиц в условиях сильной их окклюзии, что гарантирует надежное обнаружение графической области лица в кадре.

В качестве визуальных экспертных признаков используются: угол наклона лица; угол асимметрии бровей – угол, образующийся между прямой, соединяющей центральные координаты бровей, и горизонтальной прямой, проведенной через центральную координату правой брови; угол асимметрии рта – угол, образующийся между прямой, соединяющей углы рта, и горизонтальной прямой, проведенной через правый угол рта; евклидовы расстояния между координатами пар ключевых точек лица (нос, глаза, брови и губы) и сами координаты ключевых точек. Описанные выше углы рассчитываются с использованием теоремы косинусов [26]. Вектор экспертных признаков имеет размерность Q , где Q – количество экспертных признаков в векторе.

Индивидуальные проявления последовательных эмоциональных реакций людей на одни и те же стимулы свидетельствует о различии в ПКЛЧ [27]. Например, люди испытывающие негативные эмоции имеют высокий уровень нейротизма и низкий уровень экстраверсии, тогда как люди, испытывающие преимущественно положительные эмоции, напротив, имеют высокий уровень экстраверсии и доброжелательности. В связи с тесной связью двух явлений, в качестве нейросетевого экстрактора признаков в компоненте OCEAN-AI используется эмоциональная модель, представленная в статье [13].

Модель извлекает вектор признаков размерностью W из каждого кадра. Вектора признаков записываются в соответствующие матрицы (экспертные или нейросетевые), размерность которых составляет $F \times Q$, и $F \times W$, соответственно, F – количество равномерно прореженных кадров. Далее формируются перекрывающиеся видеофрагменты, образующие множество $S_i^V = \{s_1^V, \dots, s_m^V\}$. Каждый видеофрагмент s_j^V , где $j = 1, \dots, m$, имеет L векторов экспертных или нейросетевых признаков. Таким образом матрица экспертных признаков имеет размерность $L \times Q$, нейросетевых – $L \times W$. Полученные видеофрагменты экспертных и нейросетевых признаков подаются на рекуррентные нейросети, состоящие из LSTM слоев и полносвязного слоя из пяти нейронов с линейной функцией активации, для получения гипотез предсказаний по визуальным признакам. В работе [15] представлен метод извлечения нейросетевых признаков и получения гипотез предсказаний по ним на базе нейросетевого экстрактора признаков [13] и LSTM для оценивания ПКЛЧ.

Выходными данными алгоритма АВИ являются несколько матриц. Для каждого видеофрагмента извлекаются матрицы среднеуровневых признаков, построенных для низкоуровневых экспертных признаков $HC_i^V \in \mathbb{R}^{m \times F_1^V}$ и нейросетевых признаков $NN_i^V \in \mathbb{R}^{m \times F_2^V}$ (F_1^V и F_2^V векторы визуальных среднеуровневых признаков) и матрица с объединенными гипотезами предсказаний $P_i^V \in \mathbb{R}^{m \times 10}$. Для матриц HC_i^V и NN_i^V извлекаются статистические функционалы (средние значения и стандартные отклонения), которые затем используются для алгоритма МОИ. Матрица P_i^V используется для одномодального взвешивания гипотез предсказаний, полученных от среднеуровневых визуальных признаков, с помощью пяти однослойных полносвязных нейросетей с сигмоидной функцией активации.

Третий алгоритм АТИ предназначен для оценивания уровня ПКЛЧ по текстовым признакам разговорной речи человека. В алгоритме осуществляется необходимая предобработка текстовой информации, вычисление текстовых признаков и выдача по ним гипотез предсказаний оценок ПКЛЧ.

Алгоритм АТИ реализует функции сильного ИИ в части комплексирования текстовых признаков, построенных на различных принципах (экспертных и нейросетевых), т.е. алгоритм реализует функции композитного (гибридного) ИИ для анализа текстовой информации.

Алгоритм АТИ основывается на следующих математических отношениях. На вход алгоритма поступает множество мультимодальных файлов $B = \{b_1, \dots, b_n\}$, содержащих n одноименных файлов. Алгоритм АТИ работает с каждым файлом последовательно.

Сначала алгоритм АТИ проверяет наличие текстового файла b_i^T , содержащего текстовую информацию (ручную транскрипцию), для текущего аудиофайла b_i^A . В случае отсутствия текстового файла b_i^T выполняется автоматическое распознавание речи с целью извлечения текстовой информации (орфографических транскрипций речи) с помощью открытой (свободно-доступно) нейросетевой модели (Whisper [16]). Дальнейшая обработка текстовой информации выполняется одинаково, вне зависимости от того, получена текстовая информация автоматическим или ручным способом. Текстовая информация подлежит предобработке – удалению знаков пунктуации, понижению регистра. Затем выполняется разделение текстовой информации по словам $b_i^T = \{b_{i_1}^T, \dots, b_{i_O}^T\}$, где O – количество слов в текстовом файле b_i^T . На следующем этапе для каждого слова извлекаются текстовые экспертные и нейросетевые признаки. Для извлечения экспертных признаков осуществляется лингвистический подсчет статистики слов (LIWC [19]). Для всего текстового файла строится матрица экспертных признаков размерностью $O \times M$, где M – количество экспертных признаков в векторе. Нейросетевые текстовые признаки извлекаются с помощью открытой нейросетевой модели для векторного представления слов (BERT [18]). Для всего текстового файла строится матрица нейросетевых признаков размерностью $O \times N$, где N – количество нейросетевых признаков в векторе. Полученные матрицы экспертных и нейросетевых признаков подаются на рекуррентные (двунаправленные LSTM) нейронные сети, с механизмом самовнимания и полносвязными слоями из пяти нейронов с сигмоидной функцией активации.

Выходными данными алгоритма АТИ являются несколько матриц. Матрицы среднеуровневых признаков, построенных для низкоуровневых экспертных признаков $HC_i^T \in \mathbb{R}^{1 \times F_1^T}$ и нейросетевых признаков $NN_i^T \in \mathbb{R}^{m \times F_2^T}$ (F_1^T и F_2^T – векторы визуальных среднеуровневых признаков) и матрица с объединенными гипотезами предсказаний $P_i^T \in \mathbb{R}^{1 \times 10}$. Матрицы HC_i^T и NN_i^T используются для алгоритма МОИ. Матрица P_i^T используется для одномодального взвешивания гипотез предсказаний, полученных от среднеуровневых текстовых признаков, с помощью однослойной полносвязной нейросети с сигмоидной функцией активации.

Четвертый алгоритм МОИ предназначен для комплексного оценивания уровня ПКЛЧ и является связующим звеном трех алгоритмов анализа информации (ААИ, АВИ и АТИ). Данный алгоритм выполняет объединение гипотез предсказаний, полученных от признаков, построенных на различных принципах (экспертных и нейросетевых), с помощью алгоритмов ААИ, АВИ и АТИ.

Алгоритм МОИ основывается на следующих математических соотношениях. На вход алгоритма поступает множество мультимодальных файлов $B = \{b_1, \dots, b_n\}$, где n – количество одноименных файлов в базе данных. Каждый файл проходит аудиовизуальный и текстовый анализ при помощи алгоритмов АВИ, ААИ и АТИ для извлечения гипотез предсказаний на уровне фрагментов файла аудио- и видеоинформации, а также целого файла текстовой информации. Мультимодальные файлы b_i^A и b_i^V , где $i = 1, \dots, n$, разделяются на перекрывающиеся фрагменты постоянной длительности две секунды, образуя два множества: $S_i^A = \{s_1^A, \dots, s_m^A\}$ – множество аудиофрагментов; $S_i^V = \{s_1^V, \dots, s_m^V\}$ – множество видеофрагментов. Количество коротких фрагментов данных в этих множествах одинаково и равно m . Алгоритмы АВИ и ААИ выводят матрицы среднеуровневых признаков HC_i^A , NN_i^A , HC_i^V и NN_i^V , для которых извлекаются статистические функционалы (средние значения и стандартные отклонения). Для каждого аудио-/текстового файла b_i^T АТИ выводит также две матрицы среднеуровневых признаков HC_i^T и NN_i^T .

Объединение признаков происходит в два этапа, сначала извлекается кросс-модальное внимание (три слоя внимания, анализирующие по две модальности) с помощью метода, предложенного в работе [22], затем выполняется мультимодальное взвешивание по трем модальностям с помощью метода, предложенного в работе [23].

Для пяти качеств используется одна однослойная полносвязная нейросеть. В результате работы алгоритма МОИ формирует вектор гипотез предсказаний оценок ПКЛЧ размерностью 1×5 для каждого мультимодального файла. Выходом алгоритма является матрица $U = \begin{pmatrix} u_{i1} & \dots & u_{i5} \\ \vdots & \ddots & \vdots \\ u_{n1} & \dots & u_{n5} \end{pmatrix}$, включающая $n \times 5$ объединенных гипотез предсказаний оценок отдельных ПКЛЧ.

3.2. Алгоритмы компонента

На рис. 1 приведена укрупненная блок-схема алгоритма ААИ. Алгоритм ААИ начинается со считывания аудиоинформации и разделения ее на аудиофрагменты. Далее запускается цикл, который завершается, когда все аудиофрагменты одного файла обработаны, а именно: вычислены изображения спектрограмм и экспертные признаки для всех аудиофрагментов; вычислены гипотезы предсказаний из спектрограмм и экспертных признаков. Затем формируются по два массива с гипотезами предсказаний и среднеуровневыми признаками для одного аудиофрагмента и каждого низкоуровневого

набора признаков (экспертного и нейросетевого), которые затем записываются в итоговые массивы, пополняемые по мере обработки аудиофрагментов.

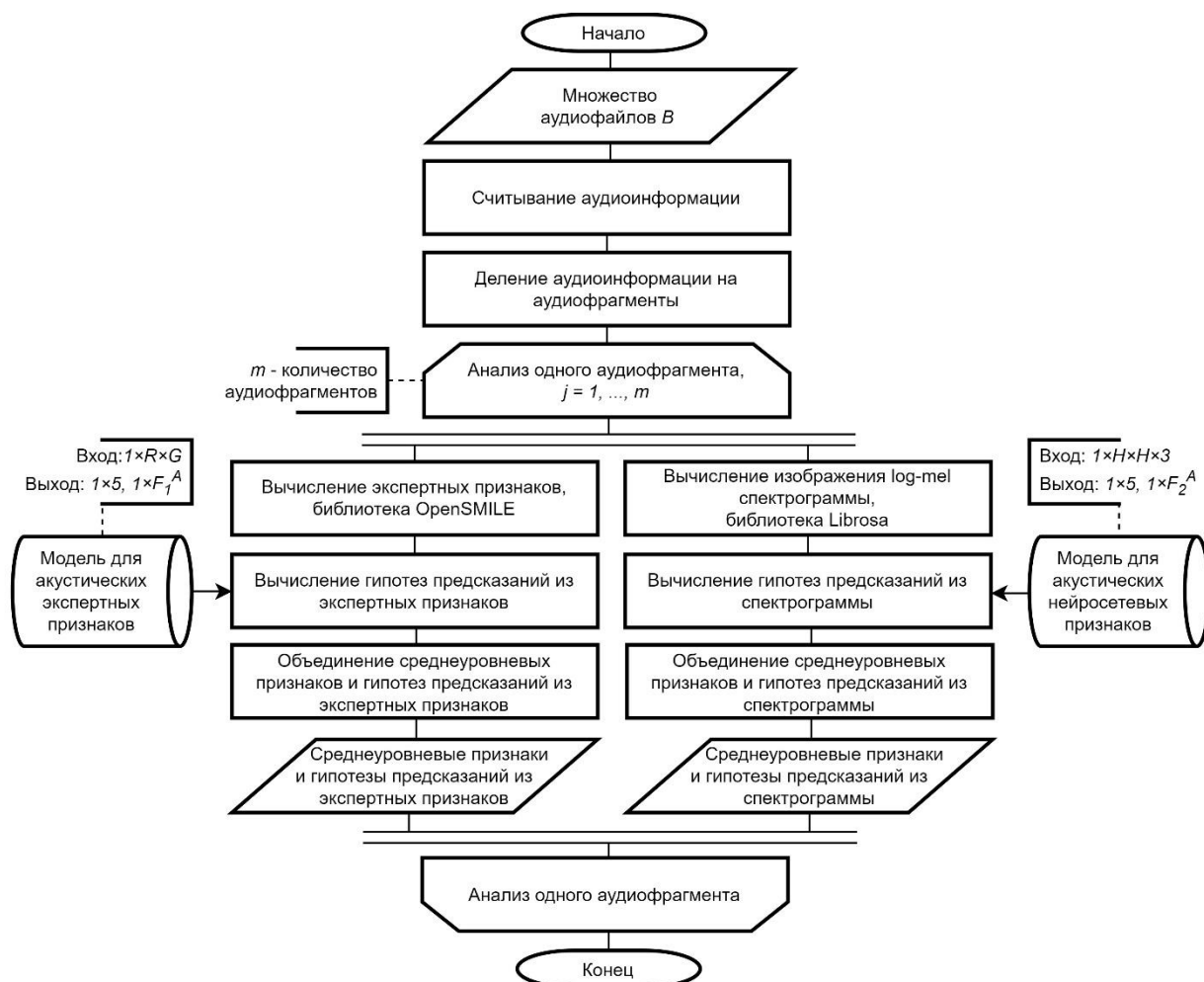


Рисунок 1 – Блок-схема алгоритма анализа аудиоинформации ААИ

На рис. 2 приведена укрупненная блок-схема алгоритма АВИ. Алгоритм АВИ начинается со считывания видеоинформации и прореживания кадров. Далее запускается цикл, который завершается, когда все кадры обработаны и записаны все экспертные и нейросетевые признаки. В цикле для каждого кадра обнаруживаются координаты ключевых точек лица (для экспертных признаков) и локализуется область лица (для нейросетевых признаков). Затем вычисляются низкоуровневые экспертные и нейросетевые признаки, которые записываются в соответствующие им матрицы векторов признаков. После завершения цикла матрицы признаков формируются в видеофрагменты по L векторов признаков в каждом и извлекаются среднеуровневые признаки и гипотезы предсказаний независимо друг от друга. Затем формируются по два массива с гипотезами предсказаний и среднеуровневыми признаками для одного видеофрагмента и каждого

низкоуровневого набора признаков (экспертного и нейросетевого), которые затем записываются в итоговые массивы, пополняемые по мере обработки видеофрагментов.

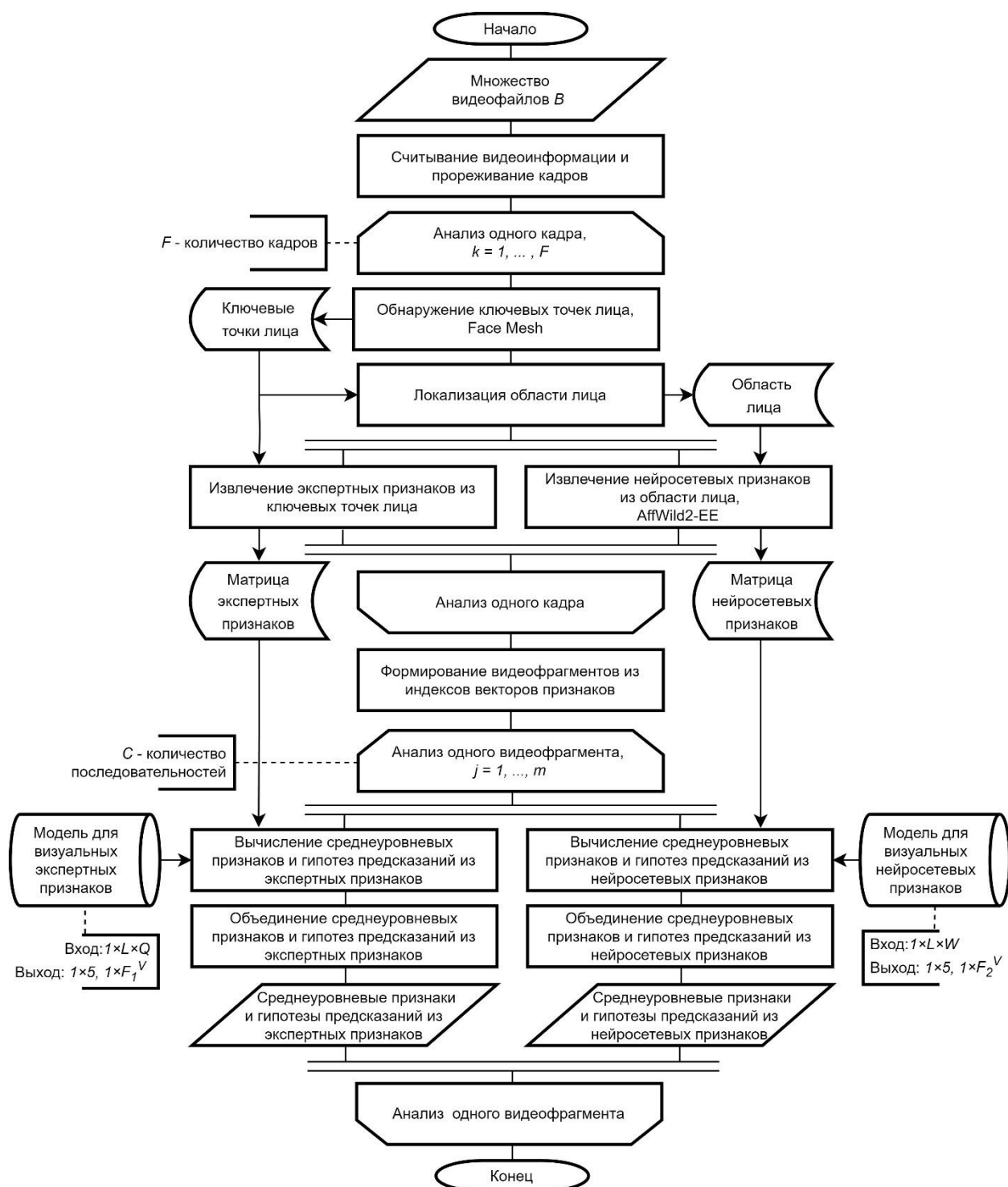


Рисунок 2 – Блок-схема алгоритма анализа видеоинформации АВИ

На рис. 3 приведена укрупненная блок-схема алгоритма АТИ. Алгоритм АТИ начинается с поиска текстового файла, при его отсутствии текстовая информация извлекается из аудиофайла, затем следует разделение текстовой информации на слова. Далее запускается цикл обработки, который завершается, когда все слова обработаны, а

именно: вычислены экспертные и нейросетевые признаки; вычислены гипотезы предсказаний из экспертных и нейросетевых признаков. Затем формируются по два массива с гипотезами предсказаний и среднеуровневыми признаками для аудио-/текстового файла и каждого низкоуровневого набора признаков (экспертного и нейросетевого), которые затем записываются в итоговые массивы.

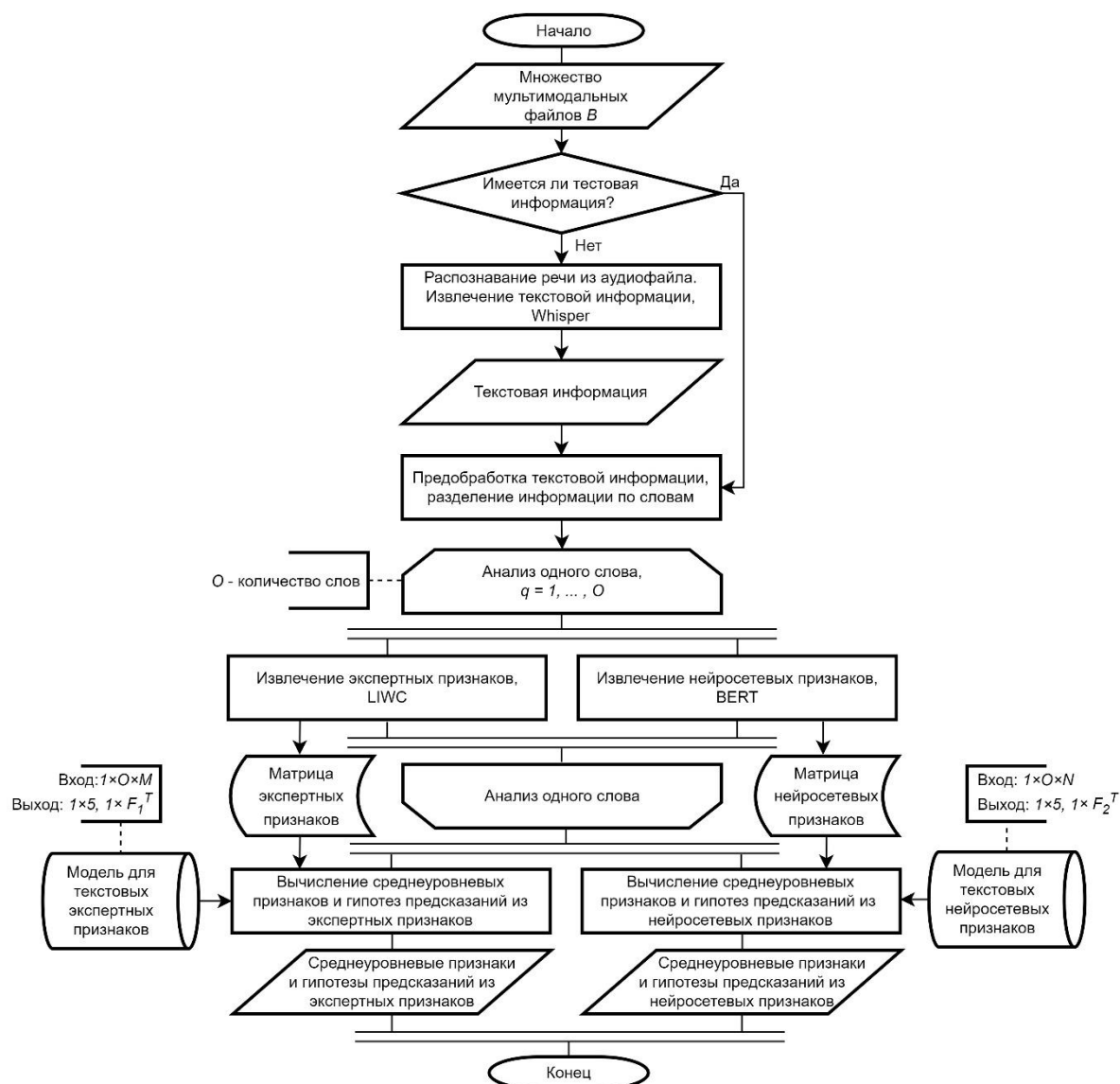


Рисунок 3 – Блок-схема алгоритма анализа текстовой информации АТИ

На рис. 4 приведена укрупненная блок-схема алгоритма МОИ. Алгоритм МОИ запускает цикл, в котором последовательно выполняются алгоритмы анализа информации ААИ, АВИ и АТИ для каждого мультимодального файла. Цикл мультимодального анализа и мультимодального объединения результатов заканчивается, когда все мультимодальные файлы из базы данных будут обработаны.

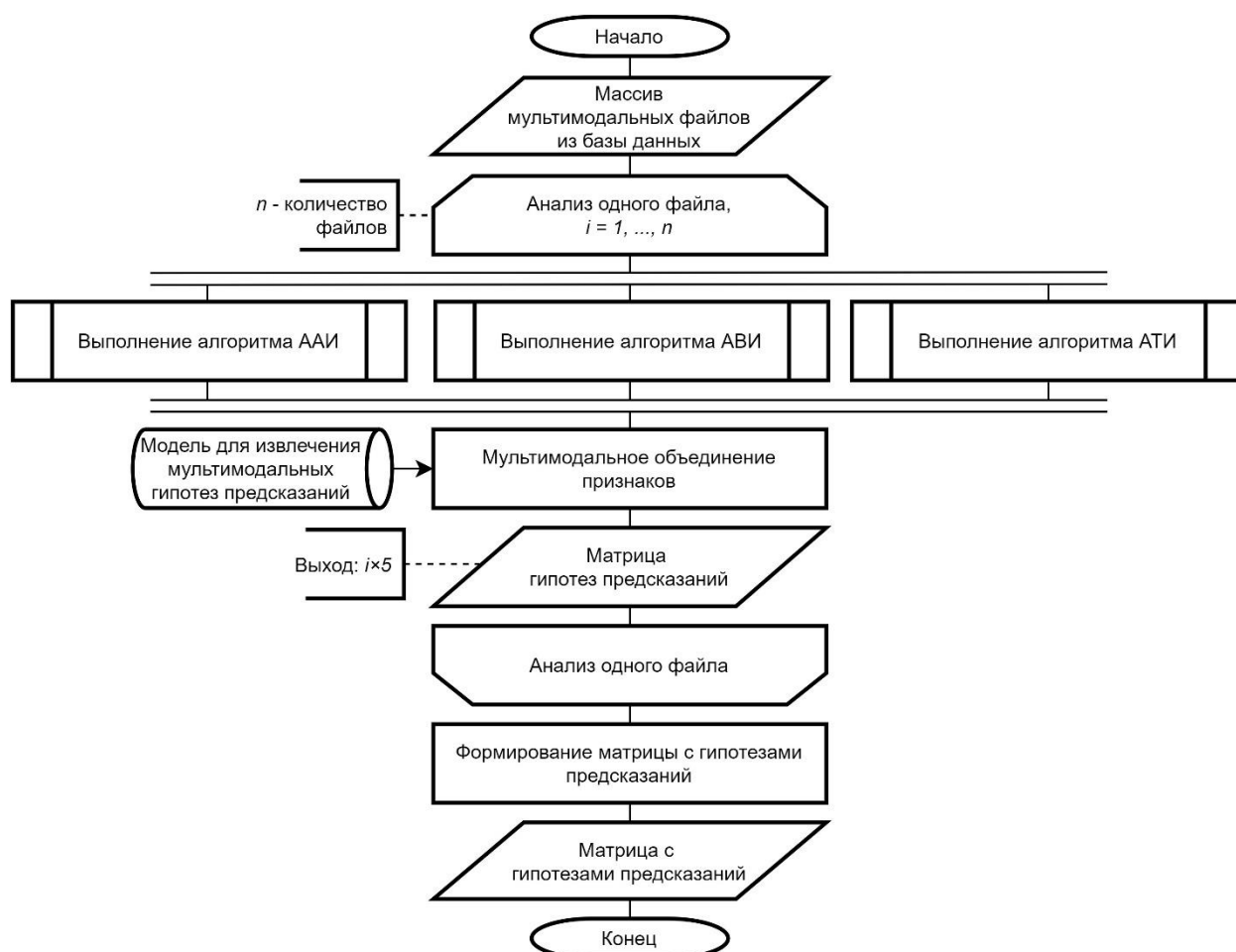


Рисунок 4 – Блок-схема алгоритма мультимодального объединения информации МОИ

3.3. Структура компонента

На рис. 5 приведена структура классов компонента OCEAN-AI, состоящая из 14 классов:

1. Класс `oceanai.modules.core.language.Language` отвечает за интернационализацию и локализацию.
2. Класс `oceanai.modules.core.messages.Messages` отвечает за вывод сообщений пользователю.
3. Класс `oceanai.modules.core.settings.Settings` отвечает за глобальные настройки компонента OCEAN-AI.
4. Класс `oceanai.modules.core.core.CoreMessages` отвечает за вывод сообщений пользователю, относящихся к классу `oceanai.modules.core.core.Core`.
5. Класс `oceanai.modules.core.core.Core` отвечает за определение общих свойств и методов компонента OCEAN-AI.
6. Класс `oceanai.modules.lab.download.DownloadMessages` отвечает за вывод сообщений, относящихся к классу `oceanai.modules.lab.download.Download`.

7. Класс `oceanai.modules.lab.download.Download` отвечает за загрузку файлов.
8. Класс `oceanai.modules.lab.audio.AudioMessages` отвечает за вывод сообщений пользователю, относящихся к классу `oceanai.modules.lab.audio.Audio`.
9. Класс `oceanai.modules.lab.audio.Audio` представляет собой программную реализацию алгоритма ААИ и отвечает за предобработку аудиоинформации, вычисление акустических низкоуровневых и среднеуровневых признаков и вывод гипотез предсказаний по ним.
10. Класс `oceanai.modules.lab.video.VideoMessages` отвечает за вывод сообщений, относящихся к классу `oceanai.modules.lab.video.Video`.
11. Класс `oceanai.modules.lab.video.Video` представляет собой программную реализацию алгоритма АВИ и отвечает за предобработку видеоинформации, вычисление визуальных низкоуровневых и среднеуровневых признаков и вывод гипотез предсказаний по ним.
12. Класс `oceanai.modules.lab.text.TextMessages` отвечает за вывод сообщений пользователю, относящихся к классу `oceanai.modules.lab.text.Text`.
13. Класс `oceanai.modules.lab.text.Text` представляет собой программную реализацию алгоритма АТИ и отвечает за предобработку текстовой информации, вычисление низкоуровневых и среднеуровневых текстовых признаков и вывод гипотез предсказаний по ним.
14. Класс `oceanai.modules.lab.prediction.PredictionMessages` отвечает за вывод сообщений пользователю, относящихся к классу `oceanai.modules.lab.prediction.Prediction`.
15. Класс `oceanai.modules.lab.prediction.Prediction` представляет собой программную реализацию алгоритма МОИ и отвечает за мультимодальное объединение информации.
16. Класс `oceanai.modules.lab.build.Run` отвечает за сборку компонента OCEAN-AI. Компонент OCEAN-AI использует вспомогательные программы:
 - `opensmile`, `librosa`, `audioread` для обработки аудиоинформации и извлечения акустических признаков в классе `oceanai.modules.lab.audio.Audio`;
 - `opencv-contrib-python`, `pymediainfo` и `mediapipe` для обработки видеоинформации и получения ключевых точек лица в классе `oceanai.modules.lab.video.Video`;
 - `liwc`, `transformers`, `sentencepiece`, `torch` и `torchaudio` для обработки текстовой информации и получения признаков разговорной речи человека в классе `oceanai.modules.lab.text.Text`.

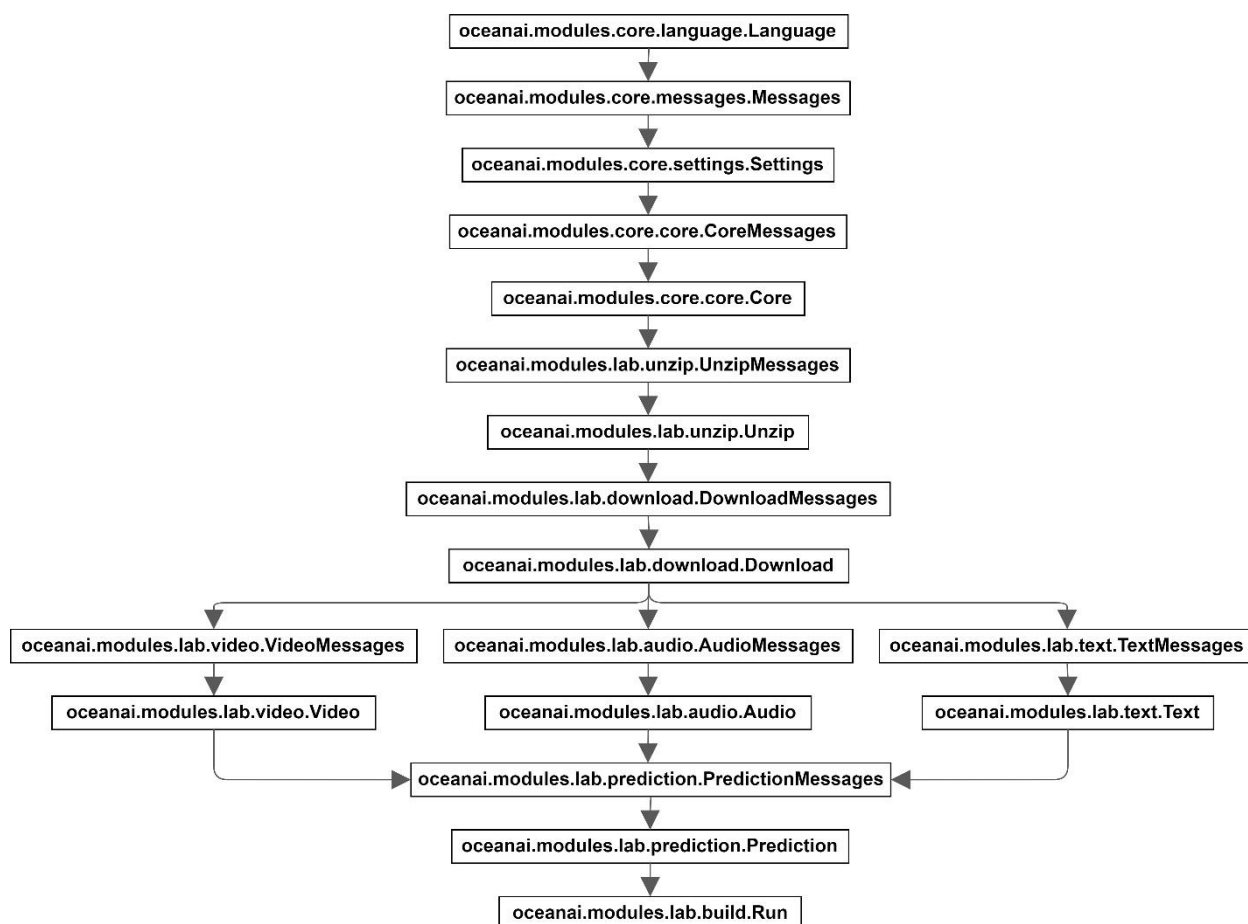


Рисунок 5 – Структура классов компонента OCEAN-AI

4. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

Для работы компонента OCEAN-AI необходимы следующие минимальные характеристики вычислительных систем, представленные в табл. 1.

Таблица 1 – Минимальные характеристики вычислительных систем, необходимые для работы компонента OCEAN-AI

Тип компьютера	Кол-во CPU x кол-во ядер	Тактовая частота CPU, ГГц	Кол-во GPU x кол-во ядер	Тактовая частота GPU, ГГц	Оперативная память, Гб	Дисковая память, Гб
Рабочая станция	1 x 8	3,8	1 x 3584	1,4	32	100

Компонент OCEAN-AI работает на CPU и GPU. Время обработки (от англ. Real Time Factor) одного мультимодального файла превышает реальное время, независимо от того, используется CPU или GPU. При использовании CPU время обработки составляет 0,6 секунд, при использовании GPU – 0,3 секунд. Использование GPU позволяет ускорить обработку мультимодальных файлов в два раза.

5. ВЫЗОВ И ЗАГРУЗКА

5.1. Базовые функции компонента

Компонент OCEAN-AI разрабатывался с использованием двух корпусов First Impressions V2 [28] и MuPTA [7], поэтому предоставляются веса моделей, обученные на двух корпусах. Для загрузки нужных весов моделей, обученных на корпусе First Impressions V2, необходимо использовать комбинацию обозначений **'fi'** (сокращение от названия корпуса First Impressions V2) и **'en'** в соответствующих аргументах в методах как показано ниже, для MuPTA – комбинацию **'mupta'** и **'ru'**. Для англоязычных мультимодальных данных допускается использование комбинации **'fi'** и **'en'**. Для русскоязычных мультимодальных данных – обе комбинации.

Обозначения **'fi'** и **'mupta'** задаются свойством `weights_for_big5_`, а также в методах `load_text_model_hc()`, `load_text_model_nn()` аргументом `corpus`. Обозначения **'ru'** и **'en'** задаются внутри методов `get_visual_features()`, `load_video_model_hc()`, `get_video_union_predictions()`, `get_text_features()`, `get_text_union_predictions()`, `get_av_union_predictions()`, `get_avt_union_predictions()` аргументом `lang`, а также используются при выборе языка интерфейса при создании экземпляра класса `oceanai.modules.lab.build.Run`. Ниже приведены примеры для загрузки весов моделей, обученных на корпусе First Impressions V2.

Создание экземпляра последнего класса из иерархии наследования классов (рис. 5) для сборки компонента OCEAN-AI:

```
from oceanai.modules.lab.build import Run

_b5 = Run(
    lang = 'ru',           # Язык интерфейса
    color_simple = #FFF,   # Цвет обычного текста (шестнадцатеричный код)
    color_info = #1776D2,  # Цвет текста, содержащего информацию (шестнадцатеричный код)
    color_err = #FF0000,   # Цвет текста, содержащего ошибку (шестнадцатеричный код)
    color_true = #008001,  # Цвет текста, содержащего положительную информацию
    (шестнадцатеричный код)
    bold_text = True,      # Жирное начертание текста
    num_to_df_display = 30, # Количество строк для отображения в таблицах
    text_runtime = Время выполнения, # Текст времени выполнения
    bold_text = True,      # Жирное начертание текста
    metadata = True,       # Отображение информации о библиотеке
)
```

5.1.1. Предобработка аудиоинформации, вычисление акустических признаков и выдача по ним гипотез предсказаний оценок ПКЛЧ.

Предобработка аудиоинформации и вычисление акустических экспертных признаков и log-mel спектрограмм:

```
# Настройки ядра
sr = 44100 # Частота дискретизации
```

```

# Путь к аудио или видеофайлу
path = '/Users/dl/GitHub/oceanai/oceanai/dataset/test80_01/glgfB3vFewc.004.mp4'
hc_features, melspectrogram_features = _b5.get_acoustic_features(
    path = path,      # Путь к аудио или видеофайлу
    sr = sr,          # Частота дискретизации
    window = 2,       # Размер окна сегмента сигнала (в секундах)
    step = 1,         # Шаг сдвига окна сегмента сигнала (в секундах)
    out = True,        # Отображение
    runtime = True,    # Подсчет времени выполнения
    run = True         # Блокировка выполнения
)

```

Для выдачи среднеуровневых признаков и гипотез предсказаний оценок ПКЛЧ по акустическим признакам, необходимо:

1. Сформировать нейросетевые архитектуры моделей:

- для вычисления среднеуровневых признаков и гипотез предсказаний оценок

ПКЛЧ по экспертным признакам:

```

res_load_model_hc = _b5.load_audio_model_hc(
    show_summary = False, # Отображение сформированной нейросетевой архитектуры модели
    out = True,          # Отображение
    runtime = True,      # Подсчет времени выполнения
    run = True           # Блокировка выполнения
)

```

- для вычисления среднеуровневых признаков и гипотез предсказаний оценок

ПКЛЧ из log-mel спектрограмм:

```

res_load_model_nn = _b5.load_audio_model_nn(
    show_summary = False, # Отображение сформированной нейросетевой архитектуры модели
    out = True,          # Отображение
    runtime = True,      # Подсчет времени выполнения
    run = True           # Блокировка выполнения
)

```

- для вычисления объединенных гипотез предсказаний оценок ПКЛЧ по

аудиоинформации и результатов оценивания методов:

```

res_load_models_b5 = _b5.load_audio_models_b5(
    show_summary = False, # Отображение сформированной нейросетевой архитектуры модели
    out = True,          # Отображение
    runtime = True,      # Подсчет времени выполнения
    run = True           # Блокировка выполнения
)

```

2. Загрузить веса нейросетевых моделей:

- для вычисления среднеуровневых признаков и гипотез предсказаний оценок

ПКЛЧ по экспертным признакам:

```

# Настройки ядра
_b5.path_to_save_ = './models' # Директория для сохранения файла
_b5.chunk_size_ = 2000000 # Размер загрузки файла из сети за 1 шаг
url = _b5.weights_for_big5_['audio']['fi']['hc']['sberdisk']
res_load_model_weights_hc = _b5.load_audio_model_weights_hc(
    url = url,          # Полный путь к файлу с весами нейросетевой модели
    force_reload = True, # Принудительная загрузка файла с весами нейросетевой модели из сети
    out = True,          # Отображение
    runtime = True,      # Подсчет времени выполнения
    run = True           # Блокировка выполнения
)

```

- для вычисления среднеуровневых признаков и гипотез предсказаний оценок

ПКЛЧ из log-mel спектрограмм:

```
# Настройки ядра
_b5.path_to_save_ = './models' # Директория для сохранения файла
_b5.chunk_size_ = 2000000 # Размер загрузки файла из сети за 1 шаг
url = _b5.weights_for_big5_['audio']['fi']['nn']['sberdisk']
res_load_model_weights_nn = _b5.load_audio_model_weights_nn(
    url = url, # Полный путь к файлу с весами нейросетевой модели
    force_reload = False, # Принудительная загрузка файла с весами нейросетевой модели из сети
    out = True, # Отображение
    runtime = True, # Подсчет времени выполнения
    run = True # Блокировка выполнения
)
```

- для вычисления объединенных гипотез предсказаний оценок ПКЛЧ по аудиоинформации и результатов оценивания методов:

```
# Настройки ядра
_b5.path_to_save_ = './models' # Директория для сохранения файла
_b5.chunk_size_ = 2000000 # Размер загрузки файла из сети за 1 шаг
url_OPE = _b5.weights_for_big5_['audio']['fi']['b5']['openness']['sberdisk']
url_CON = _b5.weights_for_big5_['audio']['fi']['b5']['conscientiousness']['sberdisk']
url_EXT = _b5.weights_for_big5_['audio']['fi']['b5']['extraversion']['sberdisk']
url_AGR = _b5.weights_for_big5_['audio']['fi']['b5']['agreeableness']['sberdisk']
url_NNEU = _b5.weights_for_big5_['audio']['fi']['b5']['non_neuroticism']['sberdisk']
res_load_models_weights_b5 = _b5.load_audio_models_weights_b5(
    url_openness = url_OPE, # Открытость опыту
    url_conscientiousness = url_CON, # Добросовестность
    url_extraversion = url_EXT, # Экстраверсия
    url_agreeableness = url_AGR, # Доброжелательность
    url_non_neuroticism = url_NNEU, # Эмоциональная стабильность
    force_reload = False, # Принудительная загрузка файла с весами нейросетевой модели из сети
    out = True, # Отображение
    runtime = True, # Подсчет времени выполнения
    run = True # Блокировка выполнения
)
```

3. Вызвать функцию (метод) для получения объединенных гипотез предсказаний оценок ПКЛЧ по аудиоинформации и результатов оценивания методов:

```
# Настройки ядра
_b5.path_to_dataset_ = '/Users/dl/GitHub/oceanai/oceanai/dataset/First_Impressions'
# Директория набора данных
# Директории не входящие в выборку
_b5.ignore_dirs_ = []
# Названия ключей для DataFrame набора данных
_b5.keys_dataset_ = ['Path', 'Openness', 'Conscientiousness', 'Extraversion', 'Agreeableness', 'Non-Neuroticism']
_b5.ext_ = ['.mp4'] # Расширения искоемых файлов
_b5.path_to_logs_ = './logs' # Директория для сохранения LOG файлов
# Полный путь к файлу с верными предсказаниями для подсчета точности
url_accuracy = _b5.true_traits_['fi']['sberdisk'] # Ключ 'fi' или 'mupta'
res_get_audio_union_predictions = _b5.get_audio_union_predictions(
    depth = 2, # Глубина иерархии для получения аудио и видеоданных
    recursive = False, # Рекурсивный поиск данных
    sr = 44100, # Частота дискретизации
    window = 2, # Размер окна сегмента сигнала (в секундах)
    step = 1, # Шаг сдвига окна сегмента сигнала (в секундах)
    accuracy = True, # Вычисление точности
    url_accuracy = url_accuracy,
    logs = True, # При необходимости формировать LOG файл
    out = True, # Отображение
    runtime = True, # Подсчет времени выполнения
    run = True # Блокировка выполнения
)
```


5.1.2. Предобработка видеoinформации, вычисление визуальных признаков и выдача по ним гипотез предсказаний оценок ПКЛЧ.

Для выполнения предварительной обработки видеoinформации и вычисления визуальных признаков необходимо:

1. Сформировать нейросетевую архитектуру модели для извлечения нейросетевых признаков по видеомодальности:

```
res_load_model_deep_fe = _b5.load_video_model_deep_fe(
    show_summary = False, # Отображение сформированной нейросетевой архитектуры модели
    out = True,           # Отображение
    runtime = True,       # Подсчет времени выполнения
    run = True            # Блокировка выполнения
)
```

2. Загрузить веса нейросетевой модели для извлечения нейросетевых признаков по видеомодальности:

```
# Настройки ядра
_b5.path_to_save_ = './models' # Директория для сохранения файла
_b5.chunk_size_ = 2000000 # Размер загрузки файла из сети за 1 шаг
url = _b5.weights_for_big5_['video']['fi']['fe']['sberdisk']
res_load_model_weights_deep_fe = _b5.load_video_model_weights_deep_fe(
    url = url,                # Полный путь к файлу с весами нейросетевой модели
    force_reload = True,     # Принудительная загрузка файла с весами нейросетевой модели из сети
    out = True,              # Отображение
    runtime = True,          # Подсчет времени выполнения
    run = True               # Блокировка выполнения
)
```

3. Вызвать функцию (метод) для предобработки видеoinформации и вычисления визуальных экспертных и нейросетевых признаков:

```
# Путь к видеофайлу
path = '/Users/dl/GitHub/oceanai/oceanai/dataset/test80_01/glgfB3vFewc.004.mp4'
hc_features, nn_features = _b5.get_visual_features(
    path = path,              # Путь к видеофайлу
    reduction_fps = 5,        # Понижение кадровой частоты
    window = 10,              # Размер окна сегмента сигнала (в кадрах)
    step = 5,                 # Шаг сдвига окна сегмента сигнала (в кадрах)
    lang = 'en',              # Выбор языка для моделей First Impressions V2 'en' для MuPTA 'ru'
    out = True,               # Отображение
    runtime = True,           # Подсчет времени выполнения
    run = True                # Блокировка выполнения
)
```

Для выдачи среднеуровневых признаков и гипотез предсказаний оценок ПКЛЧ по визуальным признакам необходимо:

1. Сформировать нейросетевые архитектуры моделей:

- для вычисления среднеуровневых признаков и гипотез предсказаний оценок

ПКЛЧ по экспертным признакам:

```
res_load_model_hc = _b5.load_video_model_hc(
    lang = 'en',              # Выбор языка для моделей First Impressions V2 'en' для MuPTA 'ru'
    show_summary = False,    # Отображение сформированной нейросетевой архитектуры модели
    out = True,               # Отображение
    runtime = True,          # Подсчет времени выполнения
    run = True                # Блокировка выполнения
)
```

)

- для вычисления среднеуровневых признаков и гипотез предсказаний оценок

ПКЛЧ по нейросетевым признакам:

```
res_load_model_nn = _b5.load_video_model_nn(
    show_summary = False, # Отображение сформированной нейросетевой архитектуры модели
    out = True,           # Отображение
    runtime = True,       # Подсчет времени выполнения
    run = True            # Блокировка выполнения
)
```

- для вычисления объединенных гипотез предсказаний оценок ПКЛЧ по видеоинформации и результатов оценивания методов:

```
res_load_models_b5 = _b5.load_video_models_b5(
    show_summary = False, # Отображение сформированной нейросетевой архитектуры модели
    out = True,           # Отображение
    runtime = True,       # Подсчет времени выполнения
    run = True            # Блокировка выполнения
)
```

2. Загрузить веса нейросетевых моделей:

- для вычисления среднеуровневых признаков и гипотез предсказаний оценок

ПКЛЧ по экспертным признакам:

```
# Настройки ядра
_b5.path_to_save_ = './models' # Директория для сохранения файла
_b5.chunk_size_ = 2000000 # Размер загрузки файла из сети за 1 шаг
url = _b5.weights_for_big5_['video']['fi']['hc']['sberdisk']
res_load_model_weights_hc = _b5.load_video_model_weights_hc(
    url = url, # Полный путь к файлу с весами нейросетевой модели
    force_reload = True, # Принудительная загрузка файла с весами нейросетевой модели из сети
    out = True, # Отображение
    runtime = True, # Подсчет времени выполнения
    run = True # Блокировка выполнения
)
```

- для вычисления среднеуровневых признаков и гипотез предсказаний оценок

ПКЛЧ по нейросетевым признакам:

```
# Настройки ядра
_b5.path_to_save_ = './models' # Директория для сохранения файла
_b5.chunk_size_ = 2000000 # Размер загрузки файла из сети за 1 шаг
url = _b5.weights_for_big5_['video']['fi']['nn']['sberdisk']
res_load_model_weights_nn = _b5.load_video_model_weights_nn(
    url = url, # Полный путь к файлу с весами нейросетевой модели
    force_reload = True, # Принудительная загрузка файла с весами нейросетевой модели из сети
    out = True, # Отображение
    runtime = True, # Подсчет времени выполнения
    run = True # Блокировка выполнения
)
```

- для вычисления объединенных гипотез предсказаний оценок ПКЛЧ по видеоинформации и результатов оценивания методов:

```
# Настройки ядра
_b5.path_to_save_ = './models' # Директория для сохранения файла
_b5.chunk_size_ = 2000000 # Размер загрузки файла из сети за 1 шаг
url_OPE = _b5.weights_for_big5_['video']['fi']['b5']['openness']['sberdisk']
url_CON = _b5.weights_for_big5_['video']['fi']['b5']['conscientiousness']['sberdisk']
url_EXT = _b5.weights_for_big5_['video']['fi']['b5']['extraversion']['sberdisk']
url_AGR = _b5.weights_for_big5_['video']['fi']['b5']['agreeableness']['sberdisk']
url_NNEU = _b5.weights_for_big5_['video']['fi']['b5']['non_neuroticism']['sberdisk']
res_load_models_weights_b5 = _b5.load_video_models_weights_b5(
```

```

url_openness = url_OPE,          # Открытость опыту
url_conscientiousness = url_CON, # Добросовестность
url_extraversion = url_EXT,      # Экстраверсия
url_agreeableness = url_AGR,     # Доброжелательность
url_non_neuroticism = url_NNEU,  # Эмоциональная стабильность
force_reload = True,             # Принудительная загрузка файла с весами нейросетевой модели из сети
out = True,                      # Отображение
runtime = True,                 # Подсчет времени выполнения
run = True # Блокировка выполнения
)

```

3. Вызвать функцию (метод) для получения объединенных гипотез предсказаний оценок ПКЛЧ по видеоинформации и результатов оценивания методов:

```

# Настройки ядра
_b5.path_to_dataset_ = '/Users/dl/GitHub/oceanai/oceanai/dataset/First_Impressions'
# Директория набора данных
# Директории не входящие в выборку
_b5.ignore_dirs_ = []
# Названия ключей для DataFrame набора данных
_b5.keys_dataset_ = ['Path', 'Openness', 'Conscientiousness', 'Extraversion', 'Agreeableness',
'Non-Neuroticism']
_b5.ext_ = ['.mp4'] # Расширения искоемых файлов
_b5.path_to_logs_ = './logs' # Директория для сохранения LOG файлов
# Полный путь к файлу с верными предсказаниями для подсчета точности
url_accuracy = _b5.true_traits_['fi']['sberdisk']
res_get_video_union_predictions = _b5.get_video_union_predictions(
    depth = 1,          # Глубина иерархии для получения аудио и видеоданных
    recursive = False,  # Рекурсивный поиск данных
    reduction_fps = 5,  # Понижение кадровой частоты
    window = 10,        # Размер окна сегмента сигнала (в секундах)
    step = 5,           # Шаг сдвига окна сегмента сигнала (в секундах)
    lang = 'en',        # Выбор языка для моделей First Impressions V2 'en' для MuPTA 'ru'
    accuracy = True,     # Вычисление точности
    url_accuracy = url_accuracy,
    logs = True,         # При необходимости формировать LOG файл
    out = True,          # Отображение
    runtime = True,      # Подсчет времени выполнения
    run = True           # Блокировка выполнения
)

```

5.1.3. Предобработка текстовой информации, вычисление текстовых признаков и выдача по ним гипотез предсказаний оценок ПКЛЧ.

Для выполнения предварительной обработки текстовой информации и вычисления визуальных признаков необходимо:

1. Загрузить словарь с экспертными признаками:

```

res_load_text_features = _b5.load_text_features(
    force_reload = True, # Принудительная загрузка файла
    out = True,         # Отображение
    runtime = True,     # Подсчет времени выполнения
    run = True          # Блокировка выполнения
)

```

2. Сформировать токенизатор и нейросетевую модель машинного перевода (RU -> EN):

```

res_setup_translation_model = _b5.setup_translation_model(
    out = True,         # Отображение
    runtime = True,     # Подсчет времени выполнения
    run = True          # Блокировка выполнения
)

```

3. Сформировать токенизатор и нейросетевую модель BERT (для кодирования слов):

```
res_setup_translation_model = _b5.setup_bert_encoder(
    force_reload = True, # Принудительная загрузка файла
    out = True,         # Отображение
    runtime = True,      # Подсчет времени выполнения
    run = True           # Блокировка выполнения
)
```

4. Вызвать функцию (метод) для предобработки текстовой информации и вычисления текстовых экспертных и нейросетевых признаков:

```
# Путь к видеофайлу
path = '/Users/dl/GitHub/oceanai/oceanai/dataset/test80_01/glgfB3vFewc.004.mp4'
hc_features, nn_features = _b5.get_text_features(
    path = path,          # Путь к видеофайлу
    asr = False,          # Распознавание речи
    lang = 'en',          # Выбор языка корпуса 'en' или 'ru'
    out = True,           # Отображение
    runtime = True,       # Подсчет времени выполнения
    run = True            # Блокировка выполнения
)
```

Для выдачи среднеуровневых признаков и гипотез предсказаний оценок ПКЛЧ по текстовым признакам необходимо:

4. Сформировать нейросетевые архитектуры моделей:

- для вычисления среднеуровневых признаков и гипотез предсказаний оценок

ПКЛЧ по экспертным признакам:

```
res_load_text_model_hc-fi = _b5.load_text_model_hc(
    corpus = 'fi',        # Выбор весов моделей 'fi' или 'mupta'
    show_summary = False, # Отображение сформированной нейросетевой архитектуры модели
    out = True,           # Отображение
    runtime = True,       # Подсчет времени выполнения
    run = True            # Блокировка выполнения
)
```

- для вычисления среднеуровневых признаков и гипотез предсказаний оценок

ПКЛЧ по нейросетевым признакам:

```
res_load_model_nn = _b5.load_text_model_nn(
    corpus = 'fi',        # Выбор весов моделей 'fi' или 'mupta'
    show_summary = False, # Отображение сформированной нейросетевой архитектуры модели
    out = True,           # Отображение
    runtime = True,       # Подсчет времени выполнения
    run = True            # Блокировка выполнения
)
```

• для вычисления объединенных гипотез предсказаний оценок ПКЛЧ по текстовой информации и результатов оценивания методов:

```
res_load_models_b5 = _b5.load_text_model_b5(
    show_summary = False, # Отображение сформированной нейросетевой архитектуры модели
    out = True,           # Отображение
    runtime = True,       # Подсчет времени выполнения
    run = True            # Блокировка выполнения
)
```

5. Загрузить веса нейросетевых моделей:

- для вычисления среднеуровневых признаков и гипотез предсказаний оценок

ПКЛЧ по экспертным признакам:

```
# Настройки ядра
_b5.path_to_save_ = './models' # Директория для сохранения файла
_b5.chunk_size_ = 2000000 # Размер загрузки файла из сети за 1 шаг
url = _b5.weights_for_big5_['text']['fi']['hc']['sberdisk']
res_load_model_weights_hc = _b5.load_text_model_weights_hc(
    url = url, # Полный путь к файлу с весами нейросетевой модели
    force_reload = True, # Принудительная загрузка файла с весами нейросетевой модели из сети
    out = True, # Отображение
    runtime = True, # Подсчет времени выполнения
    run = True # Блокировка выполнения
)
```

- для вычисления среднеуровневых признаков и гипотез предсказаний оценок

ПКЛЧ по нейросетевым признакам:

```
# Настройки ядра
_b5.path_to_save_ = './models' # Директория для сохранения файла
_b5.chunk_size_ = 2000000 # Размер загрузки файла из сети за 1 шаг
url = _b5.weights_for_big5_['text']['fi']['nn']['sberdisk']
res_load_model_weights_nn = _b5.load_text_model_weights_nn(
    url = url, # Полный путь к файлу с весами нейросетевой модели
    force_reload = True, # Принудительная загрузка файла с весами нейросетевой модели из сети
    out = True, # Отображение
    runtime = True, # Подсчет времени выполнения
    run = True # Блокировка выполнения
)
```

- для вычисления объединенных гипотез предсказаний оценок ПКЛЧ по текстовой информации и результатов оценивания методов:

```
# Настройки ядра
_b5.path_to_save_ = './models' # Директория для сохранения файла
_b5.chunk_size_ = 2000000 # Размер загрузки файла из сети за 1 шаг
url = _b5.weights_for_big5_['text']['fi']['b5']['sberdisk']
res_load_text_model_weights_b5 = _b5.load_text_model_weights_b5(
    url = url, # Полный путь к файлу с весами нейросетевой модели
    force_reload = True, # Принудительная загрузка файла с весами нейросетевой модели из сети
    out = True, # Отображение
    runtime = True, # Подсчет времени выполнения
    run = True # Блокировка выполнения
)
```

6. Вызвать функцию (метод) для получения объединенных гипотез предсказаний оценок ПКЛЧ по текстовой информации и результатов оценивания методов:

```
# Настройки ядра
_b5.path_to_dataset_ = '/Users/dl/GitHub/oceanai/oceanai/dataset/First_Impressions'
# Директория набора данных
# Директории не входящие в выборку
_b5.ignore_dirs_ = []
# Названия ключей для DataFrame набора данных
_b5.keys_dataset_ = ['Path', 'Openness', 'Conscientiousness', 'Extraversion', 'Agreeableness', 'Non-Neuroticism']
_b5.ext_ = ['.mp4'] # Расширения искомых файлов
_b5.path_to_logs_ = './logs' # Директория для сохранения LOG файлов
# Полный путь к файлу с верными предсказаниями для подсчета точности
url_accuracy = _b5.true_traits_['fi']['sberdisk']
res_get_text_union_predictions = _b5.get_text_union_predictions(
    depth = 1, # Глубина иерархии для получения аудио и видеоданных
    recursive = False, # Рекурсивный поиск данных
    asr = False, # Распознавание речи
    lang = 'en', # Выбор языка для моделей First Impressions V2 'en' для MuPTA 'ru'
    accuracy = True, # Вычисление точности
)
```

```

url_accuracy = url_accuracy,
logs = True,      # При необходимости формировать LOG файл
out = True,       # Отображение
runtime = True,   # Подсчет времени выполнения
run = True        # Блокировка выполнения
)

```

5.1.4. Выполнение мультимодального объединения среднеуровневых признаков, полученных от аудио-, видео- и текстовой информации, и получение мультимодальных гипотез предсказаний оценок ПКЛЧ.

Для выполнения мультимодального объединения среднеуровневых признаков и получения гипотез предсказаний оценок ПКЛЧ необходимо:

1. Сформировать нейросетевую архитектуру модели для выдачи мультимодальных гипотез предсказаний оценок ПКЛЧ и результатов оценивания методов:

```

res_load_avt_model_b5 = _b5.load_avt_model_b5(
    show_summary = False, # Отображение сформированной нейросетевой архитектуры модели
    out = True, # Отображение
    runtime = True, # Подсчет времени выполнения
    run = True # Блокировка выполнения
)

```

2. Загрузить веса нейросетевой модели для получения мультимодальных гипотез предсказаний оценок ПКЛЧ и результатов оценивания методов:

```

# Настройки ядра
_b5.path_to_save_ = './models' # Директория для сохранения файла
_b5.chunk_size_ = 2000000 # Размер загрузки файла из сети за 1 шаг
url = _b5.weights_for_big5_['avt']['fi']['b5']['openness']['sberdisk']
res_load_avt_model_weights_b5 = _b5.load_avt_model_weights_b5(
    url = url, # Полный путь к файлу с весами нейросетевой модели
    force_reload = False, # Принудительная загрузка файла с весами нейросетевой модели из сети
    out = True, # Отображение
    runtime = True, # Подсчет времени выполнения
    run = True # Блокировка выполнения
)

```

3. Вызвать функцию (метод) для получения мультимодальных гипотез предсказаний оценок ПКЛЧ и результатов оценивания методов:

```

_b5.path_to_dataset_ = '/Users/dl/GitHub/OCEANAI.private/oceanai/dataset//First_Impressions' #
Директория набора данных
# Названия ключей для DataFrame набора данных
_b5.keys_dataset_ = ['Path', 'Openness', 'Conscientiousness', 'Extraversion', 'Agreeableness',
'Non-Neuroticism']
_b5.ext_ = ['.mp4'] # Расширения искомых файлов
_b5.path_to_logs_ = './logs' # Директория для сохранения LOG файлов
# Полный путь к файлу с верными предсказаниями для подсчета точности
url_accuracy = _b5.true_traits_['fi']['sberdisk']
_b5.get_avt_union_predictions(
    depth = 2, # Глубина иерархии для получения аудио и видеоданных
    recursive = False, # Рекурсивный поиск данных
    sr = 44100, # Частота дискретизации
    window_audio = 2, # Размер окна сегмента аудио сигнала (в секундах)
    step_audio = 1, # Шаг сдвига окна сегмента аудио сигнала (в секундах)
    reduction_fps = 5, # Понижение кадровой частоты
    window_video = 10, # Размер окна сегмента видео сигнала (в секундах)
    step_video = 5, # Шаг сдвига окна сегмента видео сигнала (в секундах)
    asr = False, # Распознавание речи
    lang = 'en', # Выбор языка для моделей First Impressions V2 'en' для MuPTA 'ru'
)

```



```

accuracy = True,      # Вычисление точности
url_accuracy = url_accuracy,
logs = True,          # При необходимости формировать LOG файл
out = True,           # Отображение
runtime = True,       # Подсчет времени выполнения
run = True            # Блокировка выполнения
)

```

5.2. Пайплайны компонента на основе его функций

Структура (пайплайн) компонента OCEAN-AI для оценивания ПКЛЧ по мультимодальным данным приложен на рисунке 6.

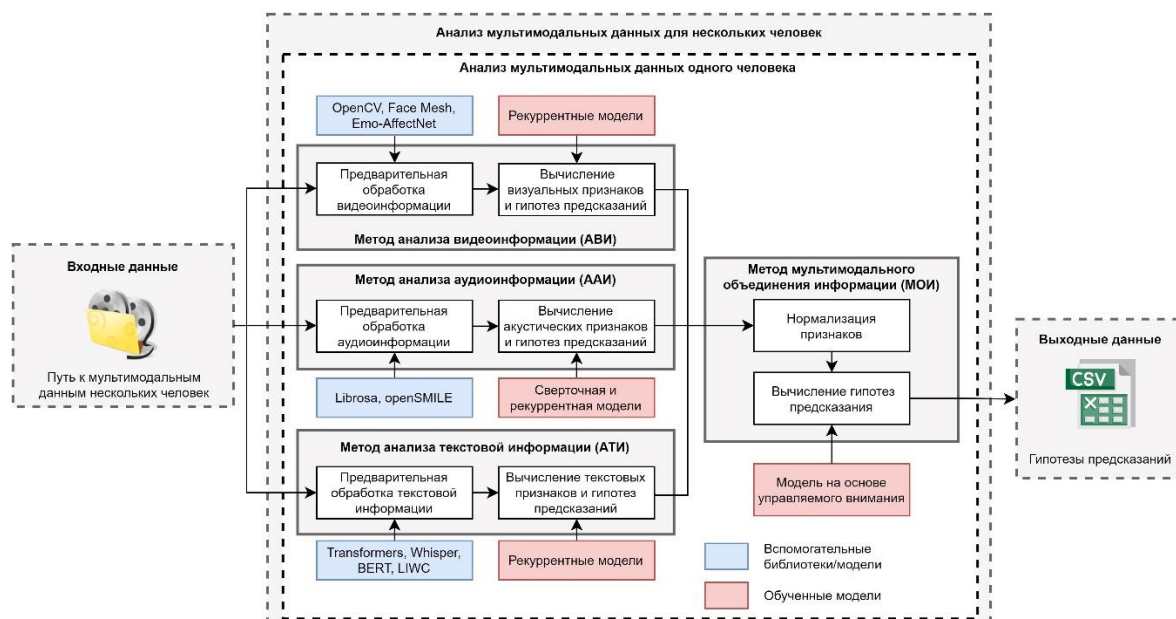


Рисунок 6 – Структура (пайплайн) компонента OCEAN-AI для оценивания ПКЛЧ по мультимодальным данным

Согласно рис. 6 на вход компонента OCEAN-AI поступает множество мультимодальных данных нескольких человек. Компонент обрабатывает мультимодальные данные последовательно с помощью алгоритмов ААИ, АВИ, АТИ и МОИ. Ниже подробно рассмотрена работа каждого алгоритма компонента OCEAN-AI.

5.2.1. Структура (пайплайн) для алгоритма анализа аудиоинформации ААИ

Структура (пайплайн) компонента OCEAN-AI для алгоритма ААИ для оценивания ПКЛЧ по голосовым характеристикам представлена на рис. 7.



Рисунок 7 – Структура (пайплайн) компонента OCEAN-AI для алгоритма анализа аудиоинформации ААИ для оценивания ПКЛЧ по голосовым характеристикам

Для получения оценок ПКЛЧ по аудиоинформации согласно рис. 7 для 10 тестовых мультимодальных файлов из корпуса First Impressions V2 [28] необходимо выполнить последовательность действий:

```
# Импорт необходимых инструментов
import os
import pandas as pd

# Импорт модуля
from oceanai.modules.lab.build import Run

# Создание экземпляра класса
_b5 = Run()

# Настройка ядра
_b5.path_to_save_ = './models' # Директория для сохранения файла
_b5.chunk_size_ = 2000000      # Размер загрузки файла из сети за 1 шаг

# Формирование аудиомоделей
res_load_model_hc = _b5.load_audio_model_hc()
res_load_model_nn = _b5.load_audio_model_nn()

# Загрузка весов аудиомоделей
url = _b5.weights_for_big5_['audio']['fi']['hc']['sberdisk']
res_load_model_weights_hc = _b5.load_audio_model_weights_hc(url = url)

url = _b5.weights_for_big5_['audio']['fi']['nn']['sberdisk']
res_load_model_weights_nn = _b5.load_audio_model_weights_nn(url = url)

# Формирование модели для анализа аудиоинформации
res_load_model_b5 = _b5.load_audio_models_b5()

# Загрузка весов модели для анализа аудиоинформации
url_OPE = _b5.weights_for_big5_['audio']['fi']['b5']['openness']['sberdisk']
url_CON = _b5.weights_for_big5_['audio']['fi']['b5']['conscientiousness']['sberdisk']
url_EXT = _b5.weights_for_big5_['audio']['fi']['b5']['extraversion']['sberdisk']
url_AGR = _b5.weights_for_big5_['audio']['fi']['b5']['agreeableness']['sberdisk']
url_NNEU = _b5.weights_for_big5_['audio']['fi']['b5']['non_neuroticism']['sberdisk']

res_load_models_weights_b5 = _b5.load_audio_models_weights_b5(
    url_openness = url_OPE,          # Открытость опыту
    url_conscientiousness = url_CON,  # Добросовестность
    url_extraversion = url_EXT,       # Экстраверсия
    url_agreeableness = url_AGR,      # Доброжелательность
    url_non_neuroticism = url_NNEU,   # Эмоциональная стабильность
)
```



```

PATH_TO_DIR = './video_FI/'
PATH_SAVE_VIDEO = './video_FI/test/'

_b5.path_to_save_ = PATH_SAVE_VIDEO

# Загрузка 10 тестовых мультимодальных файлов из корпуса First Impressions V2
# URL: https://chalearnlap.cvc.uab.cat/dataset/24/description/
domain = 'https://download.sberdisk.ru/download/file/'
tets_name_files = [
    '429713680?token=FqHdMLSSh7zYSZt&filename=_plk5k7PBEg.003.mp4',
    '429713681?token=Hz9b4lQkrLfic33&filename=be0DQawtVKE.002.mp4',
    '429713683?token=EgUXS9Xs8xHm5gz&filename=2d6btbaNdfo.000.mp4',
    '429713684?token=1U26753kmPYdIgt&filename=300gK3CnzW0.003.mp4',
    '429713685?token=LyigAWLTzDNwKJO&filename=300gK3CnzW0.001.mp4',
    '429713686?token=EpFRbCKHyuc4HPu&filename=cLaZxEf1nE4.004.mp4',
    '429713687?token=FNTkwqBr4jOS95l&filename=g24JGYuT74A.004.mp4',
    '429713688?token=qDT95nz7hfm2Nki&filename=JZNMxa3OKHY.000.mp4',
    '429713689?token=noLguEGXDpbckhg&filename=nvlqJbHk_Lc.003.mp4',
    '429713679?token=9L7RQ0hgdJlcek6&filename=4vdJGgZpj4k.003.mp4'
]

for curr_files in tets_name_files:
    _b5.download_file_from_url(url = domain + curr_files, out = False)

# Получение прогнозов
_b5.path_to_dataset_ = PATH_TO_DIR # Директория набора данных
_b5.ext_ = ['.mp4'] # Расширения искоемых файлов

# Полный путь к файлу с верными предсказаниями для подсчета точности
url_accuracy = _b5.true_traits_['fi']['sberdisk']

_b5.get_audio_union_predictions(url_accuracy = url_accuracy)

```

5.2.2. Структура (пайплайн) для алгоритма анализа видеoinформации АВИ

Структура (пайплайн) компонента OCEAN-AI для алгоритма АВИ для оценивания ПКЛЧ по лицевым характеристикам представлена на рис. 8.

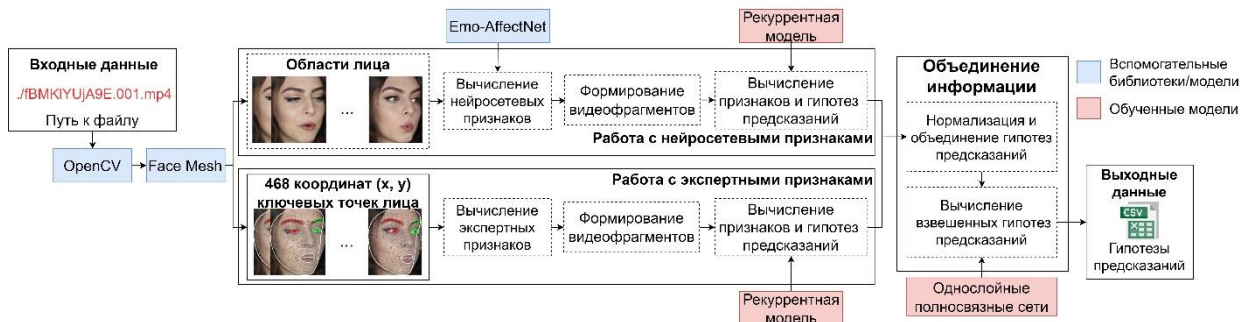


Рисунок 8 – Структура (пайплайн) компонента OCEAN-AI для алгоритма анализа видеoinформации АВИ для оценивания ПКЛЧ по лицевым характеристикам

Для получения оценок ПКЛЧ по видеoinформации согласно рис. 8 для 10 тестовых мультимодальных файлов из корпуса First Impressions V2 необходимо выполнить последовательность действий:

```

# Импорт необходимых инструментов
import os
import pandas as pd

```

```

# Импорт модуля
from oceanai.modules.lab.build import Run

# Создание экземпляра класса
_b5 = Run()

# Настройка ядра
_b5.path_to_save_ = './models' # Директория для сохранения файла
_b5.chunk_size_ = 2000000      # Размер загрузки файла из сети за 1 шаг

# Формирование видеомоделей
res_load_model_hc = _b5.load_video_model_hc(lang='en')
res_load_model_deep_fe = _b5.load_video_model_deep_fe()
res_load_model_nn = _b5.load_video_model_nn()

# Загрузка весов видеомоделей
url = _b5.weights_for_big5_['video']['fi']['hc']['sberdisk']
res_load_model_weights_hc = _b5.load_video_model_weights_hc(url = url)

url = _b5.weights_for_big5_['video']['fi']['fe']['sberdisk']
res_load_model_weights_deep_fe = _b5.load_video_model_weights_deep_fe(url = url)

url = _b5.weights_for_big5_['video']['fi']['nn']['sberdisk']
res_load_model_weights_nn = _b5.load_video_model_weights_nn(url = url)

# Формирование модели для анализа видеoinформации
res_load_model_b5 = _b5.load_video_models_b5()

# Загрузка весов модели для анализа видеoinформации
url_OPE = _b5.weights_for_big5_['video']['fi']['b5']['openness']['sberdisk']
url_CON = _b5.weights_for_big5_['video']['fi']['b5']['conscientiousness']['sberdisk']
url_EXT = _b5.weights_for_big5_['video']['fi']['b5']['extraversion']['sberdisk']
url_AGR = _b5.weights_for_big5_['video']['fi']['b5']['agreeableness']['sberdisk']
url_NNEU = _b5.weights_for_big5_['video']['fi']['b5']['non_neuroticism']['sberdisk']

res_load_models_weights_b5 = _b5.load_video_models_weights_b5(
    url_openness = url_OPE,      # Открытость опыту
    url_conscientiousness = url_CON, # Добросовестность
    url_extraversion = url_EXT,   # Экстраверсия
    url_agreeableness = url_AGR,  # Доброжелательность
    url_non_neuroticism = url_NNEU, # Эмоциональная стабильность
)

PATH_TO_DIR = './video_FI/'
PATH_SAVE_VIDEO = './video_FI/test/'

_b5.path_to_save_ = PATH_SAVE_VIDEO

# Загрузка 10 тестовых мультимодальных файлов из корпуса First Impressions V2
# URL: https://chalearnlap.cvc.uab.cat/dataset/24/description/
domain = 'https://download.sberdisk.ru/download/file/'
tets_name_files = [
    '429713680?token=FqHdMLSSh7zYSZt&filename=_plk5k7PBEG.003.mp4',
    '429713681?token=Hz9b4lQkrLf33&filename=be0DQawtVKE.002.mp4',
    '429713683?token=EgUXS9Xs8xHm5gz&filename=2d6btbaNdfo.000.mp4',
    '429713684?token=1U26753kmPYdIgt&filename=300gK3CnzW0.003.mp4',
    '429713685?token=LyigAWLTzDNwKJO&filename=300gK3CnzW0.001.mp4',
    '429713686?token=EpFRbCKHyuc4HPu&filename=cLaZxEf1nE4.004.mp4',
    '429713687?token=FNTkwqBr4jOS95l&filename=g24JGYuT74A.004.mp4',
    '429713688?token=qDT95nz7hf2Nki&filename=JZNMxa30KHY.000.mp4',
    '429713689?token=noLguEGXDpbCKhg&filename=nv1qJbHk_Lc.003.mp4',
    '429713679?token=9L7RQ0hgdJlcek6&filename=4vdJGgZpj4k.003.mp4'
]

for curr_files in tets_name_files:
    _b5.download_file_from_url(url = domain + curr_files, out = False)

```

```
# Получение прогнозов
_b5.path_to_dataset_ = PATH_TO_DIR # Директория набора данных
_b5.ext_ = ['.mp4'] # Расширения искомых файлов

# Полный путь к файлу с верными предсказаниями для подсчета точности
url_accuracy = _b5.true_traits_['fi']['sberdisk']

_b5.get_video_union_predictions(url_accuracy = url_accuracy, lang='en')
```

5.2.3. Структура (пайплайн) для алгоритма анализа текстовой информации АТИ

Структура (пайплайн) компонента OCEAN-AI для алгоритма АТИ для оценивания ПКЛЧ по текстовым признакам разговорной речи человека представлена на рис. 9.

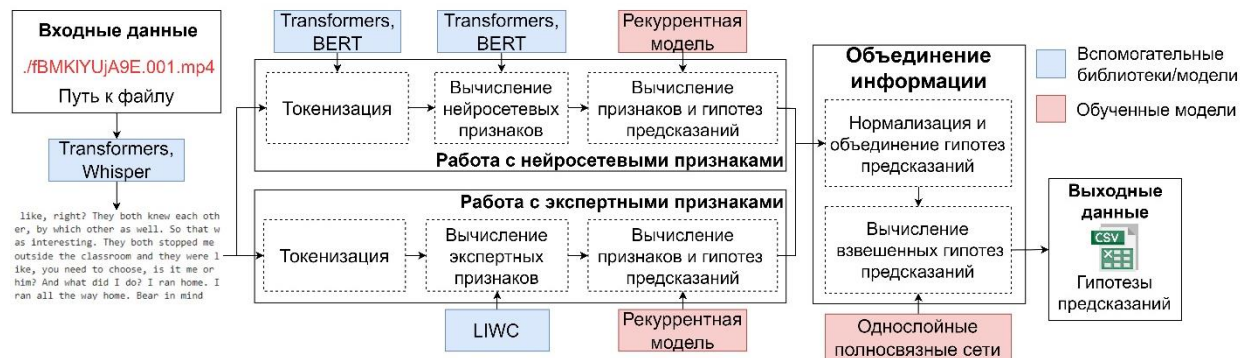


Рисунок 9 – Структура (пайплайн) компонента OCEAN-AI для алгоритма анализа видеоинформации АТИ для оценивания ПКЛЧ по признакам разговорной речи человека

Для получения оценок ПКЛЧ по текстовой информации согласно рис. 9 для 10 тестовых мультимодальных файлов из корпуса First Impressions V2 необходимо выполнить последовательность действий:

```
# Импорт необходимых инструментов
import os
import pandas as pd

# Импорт модуля
from oceanai.modules.lab.build import Run

# Создание экземпляра класса
_b5 = Run()

# Настройка ядра
_b5.path_to_save_ = './models' # Директория для сохранения файла
_b5.chunk_size_ = 2000000 # Размер загрузки файла из сети за 1 шаг

# Формирование вспомогательных моделей для обработки текстовой информации
_b5.load_text_features()
_b5.setup_translation_model()
_b5.setup_bert_encoder()

# Формирование текстовых моделей
res_load_model_hc = _b5.load_text_model_hc(corpus='fi')
res_load_model_nn = _b5.load_text_model_nn(corpus='fi')

# Загрузка весов текстовых моделей
url = _b5.weights_for_big5_['text']['fi']['hc']['sberdisk']
```

```

res_load_model_weights_hc = _b5.load_text_model_weights_hc(url = url)

url = _b5.weights_for_big5_['text']['fi']['nn']['sberdisk']
res_load_model_weights_nn = _b5.load_text_model_weights_nn(url = url)

# Формирование модели для анализа текстовой информации
res_load_model_b5 = _b5.load_text_model_b5()

# Загрузка весов модели для анализа текстовой информации
url = _b5.weights_for_big5_['text']['fi']['b5']['sberdisk']
res_load_model_weights_b5 = _b5.load_text_model_weights_b5(url = url)

PATH_TO_DIR = './video_FI/'
PATH_SAVE_VIDEO = './video_FI/test/'

_b5.path_to_save_ = PATH_SAVE_VIDEO

# Загрузка 10 тестовых мультимодальных файлов из корня First Impressions V2
# URL: https://chalearnlap.cvc.uab.cat/dataset/24/description/
domain = 'https://download.sberdisk.ru/download/file/'
tets_name_files = [
    '429713680?token=FqHdMLSSh7zYSZt&filename=_plk5k7PBEg.003.mp4',
    '429713681?token=Hz9b4lQkrLfic33&filename=be0DQawtVKE.002.mp4',
    '429713683?token=EgUXS9Xs8xHm5gz&filename=2d6btbaNdfo.000.mp4',
    '429713684?token=1U26753kmPYdIgt&filename=300gK3CnzW0.003.mp4',
    '429713685?token=LyigAWLTzDNwKJO&filename=300gK3CnzW0.001.mp4',
    '429713686?token=EpFRbCKHyuc4HPu&filename=cLaZxEf1nE4.004.mp4',
    '429713687?token=FNTkwqBr4jOS95l&filename=g24JGYuT74A.004.mp4',
    '429713688?token=qDT95nz7hfm2Nki&filename=JZNMxa30KHY.000.mp4',
    '429713689?token=noLguEGXDpbckHg&filename=nvlqJbHk_Lc.003.mp4',
    '429713679?token=9L7RQ0hgdJlcek6&filename=4vdJGgZpj4k.003.mp4'
]

for curr_files in tets_name_files:
    _b5.download_file_from_url(url = domain + curr_files, out = False)

# Получение прогнозов
_b5.path_to_dataset_ = PATH_TO_DIR # Директория набора данных
_b5.ext_ = ['.mp4'] # Расширения искоемых файлов

# Полный путь к файлу с верными предсказаниями для подсчета точности
url_accuracy = _b5.true_traits_['fi']['sberdisk']

_b5.get_text_union_predictions(url_accuracy = url_accuracy, lang='en')

```

5.2.4. Структура (пайплайн) для алгоритма мультимодального объединения информации МОИ

Структура (пайплайн) компонента OCEAN-AI для алгоритма МОИ для оценивания ПКЛЧ по мультимодальным характеристикам представлена на рис. 10.

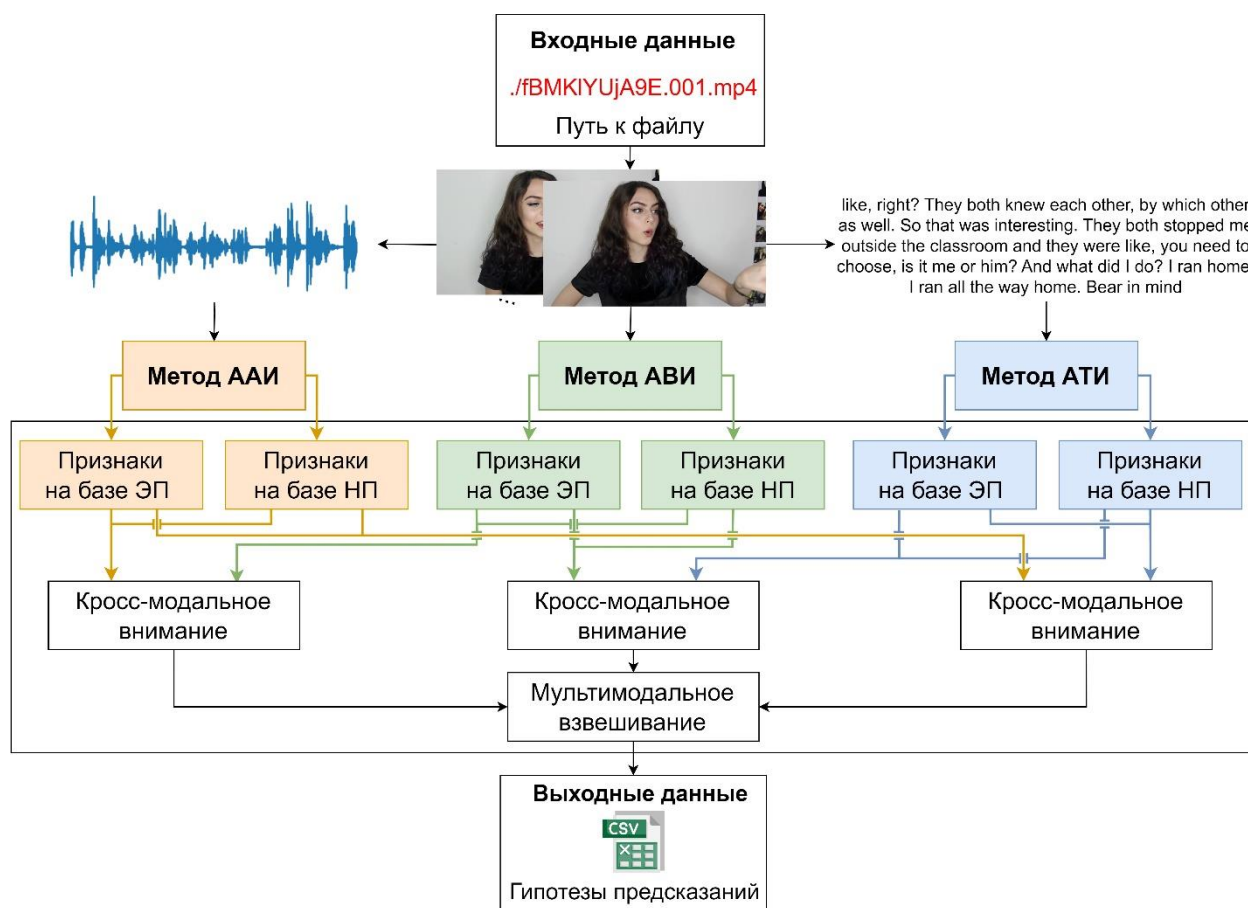


Рисунок 10 – Структура (пайплайн) компонента OCEAN-AI для алгоритма мультимодального объединения информации МОИ для оценивания ПКЛЧ по мультимодальным характеристикам

Для получения мультимодальных оценок ПКЛЧ согласно рис. 10 для 10 тестовых мультимодальных файлов из корпуса First Impressions V2 необходимо выполнить последовательность действий:

```
from oceanai.modules.lab.build import Run

# Создание экземпляра класса
_b5 = Run()

# Настройка ядра
_b5.path_to_save_ = './models' # Директория для сохранения файла
_b5.chunk_size_ = 2000000 # Размер загрузки файла из сети за 1 шаг

# Формирование аудиомоделей
res_load_model_hc = _b5.load_audio_model_hc()
res_load_model_nn = _b5.load_audio_model_nn()

# Загрузка весов аудиомоделей
url = _b5.weights_for_big5['audio']['fi']['hc']['sberdisk']
res_load_model_weights_hc = _b5.load_audio_model_weights_hc(url = url)

url = _b5.weights_for_big5['audio']['fi']['nn']['sberdisk']
res_load_model_weights_nn = _b5.load_audio_model_weights_nn(url = url)

# Формирование видеомоделей
res_load_model_hc = _b5.load_video_model_hc(lang='en')
```

```

res_load_model_deep_fe = _b5.load_video_model_deep_fe()
res_load_model_nn = _b5.load_video_model_nn()

# Загрузка весов видеомоделей
url = _b5.weights_for_big5_['video']['fi']['hc']['sberdisk']
res_load_model_weights_hc = _b5.load_video_model_weights_hc(url = url)

url = _b5.weights_for_big5_['video']['fi']['fe']['sberdisk']
res_load_model_weights_deep_fe = _b5.load_video_model_weights_deep_fe(url = url)

url = _b5.weights_for_big5_['video']['fi']['nn']['sberdisk']
res_load_model_weights_nn = _b5.load_video_model_weights_nn(url = url)

# Формирование вспомогательных моделей для обработки текстовой информации
_b5.load_text_features()
_b5.setup_translation_model()
_b5.setup_bert_encoder()

# Формирование текстовых моделей
res_load_model_hc = _b5.load_text_model_hc(corpus='fi')
res_load_model_nn = _b5.load_text_model_nn(corpus='fi')

# Загрузка весов текстовых моделей
url = _b5.weights_for_big5_['text']['fi']['hc']['sberdisk']
res_load_model_weights_hc = _b5.load_text_model_weights_hc(url = url)

url = _b5.weights_for_big5_['text']['fi']['nn']['sberdisk']
res_load_model_weights_nn = _b5.load_text_model_weights_nn(url = url)

# Формирование модели для мультимодального объединения информации
res_load_model_b5 = _b5.load_avt_model_b5()

# Загрузка весов модели для мультимодального объединения информации
url = _b5.weights_for_big5_['avt']['fi']['b5']['sberdisk']
res_load_model_weights_b5 = _b5.load_avt_model_weights_b5(url = url)

PATH_TO_DIR = './video_FI/'
PATH_SAVE_VIDEO = './video_FI/test/'

_b5.path_to_save_ = PATH_SAVE_VIDEO

# Загрузка 10 тестовых мультимодальных файлов из корпуса First Impressions V2
# URL: https://chalearnlap.cvc.uab.cat/dataset/24/description/
domain = 'https://download.sberdisk.ru/download/file/'
tets_name_files = [
    '429713680?token=FqHdMLSSh7zYSZt&filename=_plk5k7PBEg.003.mp4',
    '429713681?token=Hz9b4lQkrLfic33&filename=be0DQawtVKE.002.mp4',
    '429713683?token=EgUXS9Xs8xHm5gz&filename=2d6btbaNdfo.000.mp4',
    '429713684?token=1U26753kmPYdIgt&filename=300gK3CnzW0.003.mp4',
    '429713685?token=LyigAWLTzDNwKJO&filename=300gK3CnzW0.001.mp4',
    '429713686?token=EpFRbCKHyuc4HPu&filename=cLaZxEf1nE4.004.mp4',
    '429713687?token=FNTkwqBr4jOS95l&filename=g24JGYuT74A.004.mp4',
    '429713688?token=qDT95nz7hfm2Nki&filename=JZNMxa30KHY.000.mp4',
    '429713689?token=noLguEGXDpbckHg&filename=nvlqJbHk_Lc.003.mp4',
    '429713679?token=9L7RQ0hgdJlcek6&filename=4vdJGgZpj4k.003.mp4'
]

for curr_files in tets_name_files:
    _b5.download_file_from_url(url = domain + curr_files, out = False)

# Получение прогнозов
_b5.path_to_dataset_ = PATH_TO_DIR # Директория набора данных
_b5.ext_ = ['.mp4'] # Расширения искоемых файлов

# Полный путь к файлу с верными предсказаниями для подсчета точности
url_accuracy = _b5.true_traits_['fi']['sberdisk']
_b5.get_avt_predictions(url_accuracy = url_accuracy, lang='en')

```


6. ВХОДНЫЕ ДАННЫЕ

6.1. Состав и структура входных данных

В табл. 2 представлены состав и структура входных данных.

Таблица 2 – Состав и структура входных данных

Наименование данных	Обозначение	Структура данных	Способ ввода данных	Ограничения
Аудио-данные	b_i^A , $i = 1, \dots, n$	Переменная-массив audio класса oceanai.modules.l ab.audio.Audio	Файл с аудиоданными	Формат wav, mp4, avi, mov. Продолжительность сигнала до 170 сек.
Видео-данные	b_i^V , $i = 1, \dots, n$	Переменная-массив frame класса oceanai.modules.l ab.audio.Video	Файл с видеоданными	Формат mp4, avi, mov. Продолжительность сигнала до 170 сек.
Текстовые данные	b_i^T , $i = 1, \dots, n$	Переменная-массив text класса oceanai.modules.l ab.text.Text	Файл с текстовыми данными	Формат txt, mp4, avi, mov. Продолжительность сигнала для форматов mp4, avi, mov до 170 сек. Длина текстового файла не более 2000 символов.

6.2. Подготовка входных данных

Чтобы обеспечить библиотеку данными, необходимо указать папку размещения соответствующих мультимодальных файлов через свойство `path_to_dataset_` класса `oceanai.modules.core.settings.Settings`, расширение к файлам устанавливается через свойство `ext_` класса `oceanai.modules.core.settings.Settings`, глубина иерархии устанавливается аргументом `depth` методов `get_av_union_predictions()`, `get_avt_union_predictions()`, `get_audio_union_predictions()`, `get_video_union_predictions()` и `get_text_union_predictions()`.

```

_b5 = Run()
_b5.path_to_dataset_ = '/Users/dl/GitHub/oceanai/oceanai/dataset/First Impressions'
_b5.ext_ = ['.mp4'] # Расширения искомых файлов
res_get_avt_union_predictions = _b5.get_avt_union_predictions(
    depth = 2, # Глубина иерархии для получения аудио и видеоданных
    # Другие аргументы данного метода не относятся к подготовке данных
    # ...
)

```

7. ВЫХОДНЫЕ ДАННЫЕ

7.1. Состав и структура выходных данных

Выходными данными компонента OCEAN-AI являются файлы в формате DataFrame библиотеки Pandas. Один DataFrame содержит информацию о полных путях к мультимодальным файлам и предсказанным оценкам ПКЛЧ в табличном виде. Второй DataFrame содержит информацию о оценочных результатах работы компонента OCEAN-AI в показателях оценивания MAE и Accuracy, рассчитанных для каждого качества и их средние значения. В табл. 3 представлены состав и структура выходных данных.

Таблица 3 – Состав и структура выходных данных.

Наименование данных	Обозначение	Структура данных	Способ вывода данных	Ограничения
Объединенные гипотезы предсказаний	<i>U</i>	Переменная-матрица (DataFrame) <code>_df_files</code> класса <code>oceanai.modules.lab.prediction.Prediction</code>	Вызов метода <code>get_av_union_predictions()</code> класса <code>oceanai.modules.lab.prediction.Prediction</code>	Значения в интервале [0,1]
Показатели оценивания методов	<i>Z</i>	Переменная-матрица (DataFrame) <code>_df_accuracy</code> класса <code>oceanai.modules.lab.prediction.Prediction</code>	Вызов метода <code>get_av_union_predictions()</code> класса <code>oceanai.modules.lab.prediction.Prediction</code>	Значения в интервале [0,1]

7.2. Интерпретация выходных данных

Пользователь может задать порядок вывода ПКЛЧ обозначив их через свойство `keys_dataset_` класса `Settings`, также указать путь к сохранению выходных файлов через свойство `path_to_logs_` класса `Settings`, указать путь к экспертным оценкам ПКЛЧ для подсчёта точности в аргументе `url_accuracy` метода `get_avt_union_predictions()`, и установить аргумент `logs` на `True` в методе `get_avt_union_predictions()` для сохранения файлов. Язык для использования необходимых весов моделей устанавливается через аргумент `lang` на `'en'` или `'ru'` в методе `get_avt_union_predictions()`. Предусматривается использование не только «англоязычных моделей» для англоязычных входных данных либо «русскоязычных моделей» на русскоязычных входных данных, но и «англоязычных моделей» на русскоязычных входных данных. Отображения файлов регулируется аргументом `out` метода `get_video_union_predictions()`. Выходные данные интерпретируются в формате «csv».

```
_b5 = Run()
_b5.keys_dataset_ = ['Path', 'Openness', 'Conscientiousness', 'Extraversion', 'Agreeableness', 'Non-Neuroticism']
```



```

_b5.path_to_logs_ = './logs' # Директория для сохранения LOG файлов
res_get_avt_union_predictions = _b5.get_avt_predictions(
    logs = True, # При необходимости формировать LOG файл
    out = True, # Отображение
    lang = 'en', # Выбор языка для моделей First Impressions V2 'en' для MuPTA 'ru'
    # Другие аргументы данного метода не относятся к интерпретации выходных данных
    # ...
)

```

Примеры отображения «csv» файлов представлены на рис. 11 и 12.

	Path	Openness	Conscientiousness	Extraversion	Agreeableness	Non-Neuroticism
Person ID						
1	2d6btbaNdfo.000.mp4	0.545981	0.655463	0.412282	0.591633	0.566958
2	300gK3CnzW0.001.mp4	0.482044	0.461589	0.372908	0.528315	0.459843
3	300gK3CnzW0.003.mp4	0.447993	0.474451	0.337972	0.497484	0.413
4	4vdJGgZpj4k.003.mp4	0.59249	0.617516	0.539873	0.611519	0.582145
5	be0DQawtVKE.002.mp4	0.579003	0.567649	0.496231	0.608233	0.571018
6	cLaZxEf1nE4.004.mp4	0.653647	0.563365	0.565574	0.582277	0.590217
7	g24JGYuT74A.004.mp4	0.53703	0.329784	0.437795	0.512874	0.471008
8	JZNMxa3OKHY.000.mp4	0.634548	0.567335	0.571362	0.587481	0.584822
9	nvIqJbHk_Lc.003.mp4	0.553215	0.503285	0.428273	0.49168	0.475411
10	_plk5k7PBEG.003.mp4	0.555685	0.555988	0.463666	0.601584	0.54942

Рисунок 11 – Пример DataFrame с предсказанными оценки ПКЛЧ

Рис. 11 показывает следующую информацию: Person ID – порядковый номер человека; Path – название мультимодального файла; Openness («Открытость опыту»), Conscientiousness («Добросовестность»), Extraversion («Экстраверсия»), Agreeableness («Доброжелательность») и Non-Neuroticism («Эмоциональная стабильность») – пять ПКЛЧ, оцениваемые компонентом OCEAN-AI, и соответствующие им гипотезы предсказаний для каждой человека.

	Openness	Conscientiousness	Extraversion	Agreeableness	Non-Neuroticism	Mean
Metrics						
MAE	0.0522	0.0595	0.0825	0.0722	0.0554	0.0644
Accuracy	0.9478	0.9405	0.9175	0.9278	0.9446	0.9356

Рисунок 12 – Пример DataFrame с показателями оценивания компонента OCEAN-AI для 10 тестовых мультимодальных файлов

Рис. 2 показывает следующую информацию: Metrics – показатели оценивания метода ААИ; MAE – средняя абсолютная ошибка; Accuracy – 1-MAE. Показатели оценивания вычисляются независимо для каждого из пяти ПКЛЧ. Mean – среднее значение по всем показателям эффективности.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

AI	–	Искусственный интеллект (Artificial Intelligence)
OCEAN	–	Персональные качества личности человека: Открытость опыту (Openness to experience/Openness), Добросовестность (Conscientiousness), Экстраверсия (Extraversion), Доброжелательность (Agreeableness), Эмоциональная стабильность (Non-Neuroticism)
OCEAN-AI	–	Библиотека алгоритмов интеллектуального анализа поведения человека на основе его мультимодальных данных, обеспечивающих оценивание уровня отдельных персональных качеств личности человека
LIWC	–	Linguistic Inquiry and Word Count
LSTM	–	Нейросетевая модель с длинной кратковременной памятью (Long Short-Term Memory)
MAE	–	Средняя абсолютная ошибка (Mean Absolute Error)
ААИ	–	Анализ аудиоинформации
АВИ	–	Анализ видеоинформации
АТИ	–	Анализ текстовой информации
МОИ	–	Мультимодальное объединение информации
ПКЛЧ	–	Персональные качества личности человека

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Bradski G. The OpenCV Library // Dr. Dobb's Journal: Software Tools for the Professional Programmer, 2000, vol. 25(11), pp. 120–123.
2. McFee B. et al. Librosa: Audio and Music Signal Analysis in Python // 14th Python in Science Conference, 2015, vol. 8, pp. 18–25.
3. Eyben F., Wöllmer M., Schuller B. OpenSMILE: The Munich Versatile and Fast Open-Source Audio Feature Extractor // 18th ACM International Conference on Multimedia, 2010, pp. 1459–1462.
4. Hochreiter S., Schmidhuber J. Long Short-Term Memory // Neural Computation, 1997, vol. 9(8), pp. 1735–1780.
5. Rosenblatt F. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain // Psychological Review, 1958, vol. 65(6), p. 386.
6. Velichko A., Markitantov M., Kaya H., Karpov A. Complex Paralinguistic Analysis of Speech: Predicting Gender, Emotions and Deception in a Hierarchical Framework // Interspeech, 2022, pp. 4735–4739.
7. Ryumina E., Ryumin D., Markitantov M., Kaya H., Karpov A. Multimodal Personality Traits Assessment (MuPTA) Corpus: The Impact of Spontaneous and Read Speech // Interspeech, 2023, pp. 4049–4053.
8. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition // 3rd International Conference on Learning Representations, 2015, pp. 1–14.
9. Bradski G. The OpenCV Library // Dr. Dobb's Journal: Software Tools for the Professional Programmer, 2000, vol. 25(11), pp. 120–123.
10. Ablavatski A., Grishchenko I., Kartynnik Y., et al. Attention Mesh: High Fidelity Face Mesh Prediction in Real-Time // 4rd CVPR Workshop on Computer Vision for Augmented and Virtual Reality, 2020, pp. 1–4.
11. Markitantov M., Ryumina E., Ryumin D et al. Biometric Russian Audio-Visual Extended MASKS (BRAVE-MASKS) Corpus: Multimodal Mask Type Recognition Task // Interspeech, 2022, pp. 1756–1760.
12. Jiang P., Ergu D., Liu F., Cai Y., Ma B. A Review of Yolo algorithm developments // Procedia Computer Science, 2022, 199, 1066-1073.
13. Ryumina E., Dresvyanskiy D., Karpov A. In Search of a Robust Facial Expressions Recognition Model: A Large-Scale Visual Cross-Corpus Study // Neurocomputing - 2022, Vol. 514, pp. 435–450.

14. He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition // IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
15. Ryumina E., Markitantov M., Ryumin D., Karpov A. OCEAN-AI framework with EmoFormer cross-hemiface attention approach for personality traits assessment // Expert Systems with Applications, 2024, vol. 239, pp. 122441.
16. Radford A. et al. Robust speech recognition via large-scale weak supervision // In International Conference on Machine Learning, 2023, pp. 28492–28518.
17. Wolf T., Debut L., Sanh V. et al. Huggingface's transformers: State-of-the-art natural language processing // arXiv preprint arXiv:1910.03771, 2019, pp. 1–8.
18. Devlin J., Chang M., Lee K., Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding // Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2019, vol. 1, pp. 4171–4186.
19. Pennebaker J. W., Francis M. E., Booth R. J. Linguistic inquiry and word count: LIWC 2001 // Mahway: Lawrence Erlbaum Associates, 2001, vol. 71, pp. 1–22.
20. Graves A., Fernández S., Schmidhuber J. Bidirectional LSTM networks for improved phoneme classification and recognition // International Conference on Artificial Neural Networks? 2005, pp. 799–804.
21. Vaswani A., Shazeer N., Parmar N., et al. Attention is all you need // Conference on Neural Information Processing Systems (NIPS), 2017, pp. 1–11.
22. Ryumina E., Markitantov M., Karpov A. Multi-Corpus Learning for Audio–Visual Emotions and Sentiment Recognition // Mathematics, 2023, Vol. 11, No. 16, pp. 3519.
23. Ivanko D., Ryumin D., Kashevnik A., Axyonov A., Kitenko A., Lashkov I., Karpov A. DAVIS: Driver's Audio-Visual Speech recognition // Interspeech, 2022, 1141–1142.
24. Sharma S., Sharma S., Athaiya A. Activation Functions in Neural Networks // Towards Data Science, 2017, vol. 6(12), pp. 310–316.
25. Han J., Moraga C. The influence of the sigmoid function parameters on the speed of backpropagation learning // International Workshop on Artificial Neural Networks, 1995, pp. 195–201).
26. Lee J. R. The Law of Cosines in a Tetrahedron // The Pure and Applied Mathematics, 1997, vol. 4(1), pp. 1–6.
27. Costa P.T., McCrae R.R. A five-factor theory of personality. The five-factor model of personality: Theoretical perspectives, 1999, 2, 51–87.
28. Escalante H.J. et al. Modeling, recognizing, and explaining apparent personality from videos // IEEE Transactions on Affective Computing, 2022, vol. 13(2), pp. 894–911.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]