

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Петрозаводский государственный университет»

Институт математики и информационных технологий
Кафедра информатики и математического обеспечения

Симагин Игорь Александрович

Отчет по дисциплине «Верификация программного обеспечения»

Направление 09.03.04 — Программная инженерия

Преподаватель: к.ф.-м.н., доцент К.А. Кулаков

Петрозаводск
2022

Система отслеживания эмоций человека в кадре

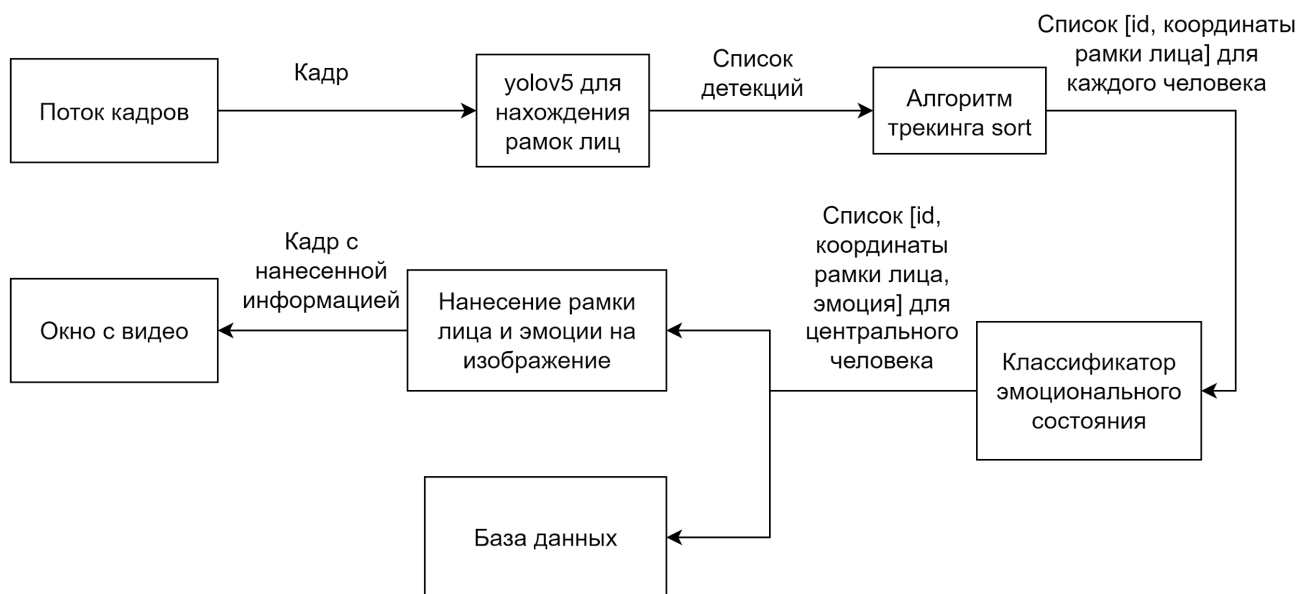
Описание объекта тестирования	3
Алгоритм работы	3
Архитектура	4
Структура базы данных	4
Сценарий использования	4
Требования к объекту тестирования	5
Тестовое окружение	5
Требования к окружению	5
Структура объекта тестирования	5
Модуль трекинга.	5
Модуль детекции.	6
Модуль отрисовки.	6
Модуль классификации эмоций.	6
Модуль работы с БД.	7
Модуль чтения кадров	7
Стратегия модульного тестирования	7
Стратегия интеграционного тестирования	7
Стратегия аттестационного тестирования	8
Стратегия специального тестирования	8
Тестирование модели классификации эмоций	8
Детальный план тестов	10
Блочное тестирование	10
Интеграционное тестирование	12
Аттестационное тестирование	13
Специальное тестирование	17
Нагрузочное тестирование	19
Покрытие кода тестами	21
Примеры тестов	22
Пример блочного теста проверки работы с БД:	22
Пример интеграционного теста:	22
Журнал тестирования	22
Журнал блочного тестирования	22
Журнал интеграционного тестирования	23
Журнал аттестационного тестирования	23
Журнал специального тестирования	24
Журнал нагрузочного тестирования	24
Журнал ошибок	24
Заключение	28

Описание объекта тестирования

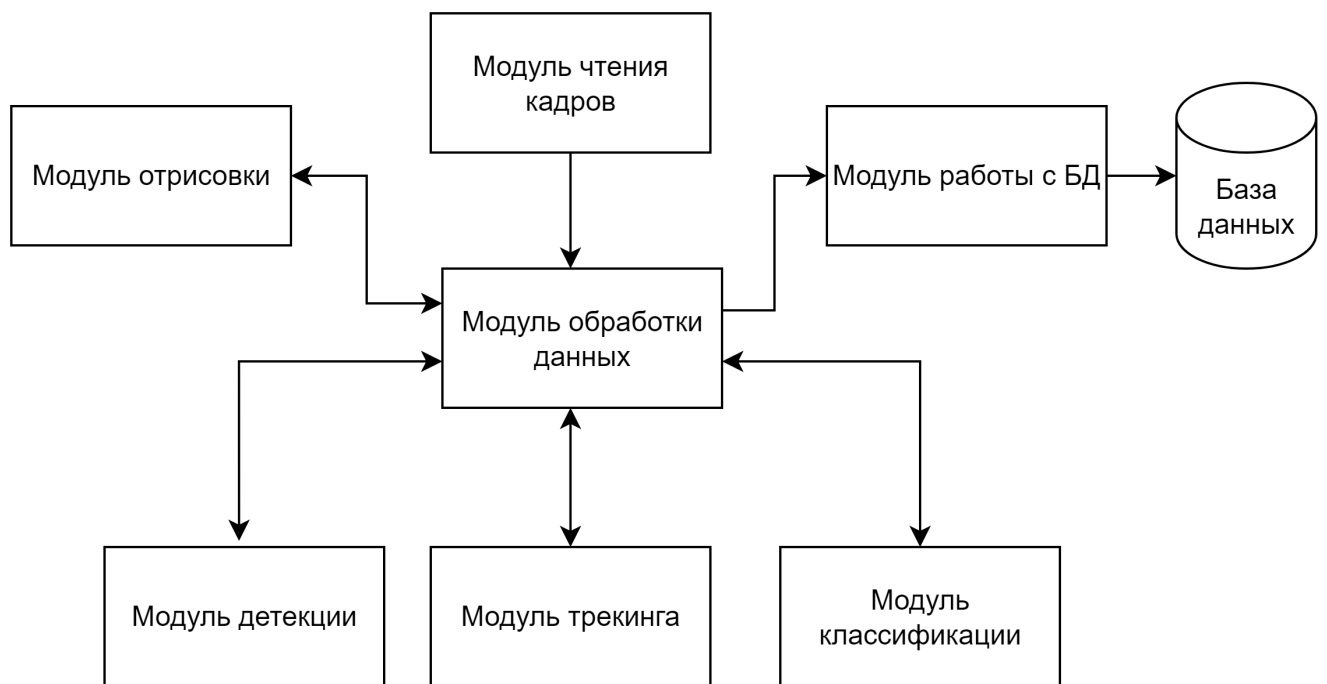
Объектом тестирования является система предназначена для распознавания эмоций центрального человека в кадре.

Помимо самого анализа, происходит также запись результатов в базу данных.

Алгоритм работы



Архитектура



Структура базы данных

В качестве базы данных использована СУБД PostgreSQL.
В БД содержится лишь одна таблица helmet_emotion.

helmet_emotion	
track_id	bigint
date	timestamp with time zone
emotion	varchar(10)
camera_id	integer
id	bigint

Сценарий использования

В кадре видны лица нескольких человек. В таком случае выбирается тот человек, лицо которого находится ближе всего к центру, после чего обрабатываются классифицируются эмоции лишь этого человека. - основной сценарий.

1. К камере подходит один или несколько человек
2. Выбирается человек, лицо которого находится ближе всего к центру
3. Распознаются эмоции этого центрального человека
4. Информация о распознанной эмоции записывается в базу данных

Требования к объекту тестирования

Тестовое окружение

Характеристики компьютера:

- Процессор - i5-12400f
- Видеокарта nvidia RTX 3060
- 32GB оперативной памяти
- ОС - windows 11
- Камера - logitech 720p

Версия языка Python 3.8.13

Виртуальное окружение создавалось с помощью пакета conda.

База данных была поднята в докер контейнере.

Версии библиотек представлены в **приложении 1**.

Требования к окружению

Система предполагает корректную работу при соблюдении следующих условий:

- Системные требования компьютера:
 - Процессор имеет 4 ядра и больше
 - Компьютер содержит видеоускоритель Nvidia, который содержит не менее 4 гигабайт видеопамяти с поддержкой cuda
 - Компьютер содержит не менее 8 гигабайт оперативной памяти
 - Платформа x86
- Требования к внешней среде:
 - Стабильное освещение
 - Камера должна висеть на уровне лица человека
 - Расстояние до человека должно быть не более 1.5 метров и не менее 0.5 метров для камеры с разрешением 720p.

Структура объекта тестирования

Модуль трекинга.

Список классов:

- EmotionTracker. Класс, содержащий публичные методы для работы с модулем трекинга, а также работающий с алгоритмом sort.

Методы:

- `__init__` - принимает следующие параметры:
 - `max_age=5` - время жизни трека без детекций,
 - `min_hits=2` - минимальное число раз до момента считания трека новым,

- `iou_threshold=0.2` - минимальное пересечение, чтобы считать, что новая рамка соответствует треку,
 - `window_size=7` - размер окна для оконного фильтра,
- `track` - функция вызывает метод для трекинга из алгоритма Sort

Модуль детекции.

Модуль содержит один `DetectionModel`, который является оберткой для модели детекции Yolo.

Методы:

- `__init__` - инициализирует модель yolo, принимает параметры:
 - `weight_path` - путь до весов модели
 - `device` - параметр определяет, на каком процессоре выполнять вычисления
 - `conf` - уверенность для отбрасывания детекций
 - `size` - разрешение модели при инференсе
- `__call__` - вызывает модель детекции и преобразует данные в нужный формат, принимает параметры:
 - `frame` - кадр

Модуль отрисовка.

Модуль содержит один класс `MyAnnotator`, который является оберткой для вывода функция `opencv`.

Методы:

- `put_text` - функция отрисовывает текст на кадре.

Параметры:

 - `frame` - кадр
 - `text` - текст
 - `position` - координаты начала текста
 - `scale=1` - размер шрифта
 - `color=(0, 0, 255)` - цвет
 - `thickness=2` - ширина линий
- `rectangle` - функция отрисовывает прямоугольник на кадре.

Параметры:

 - `frame` - кадр
 - `start` - верхний левый угол
 - `end` - нижний правый угол
 - `color=(0, 0, 255)` - цвет
 - `thickness=2` - ширина линий

Модуль классификации эмоций.

Модуль предназначен для работы с моделями классификации эмоций.

`EmotionClassifier`. Класс для внешнего взаимодействия с моделью.

Методы:

- `infer` - делает предобработку фотографии и предсказание вероятностей эмоций.

Параметры:

- img - кадр

Модуль работы с БД.

Модуль реализует асинхронную очередь для записи данных батчами.

Класс TrackingSaveQueue:

Методы:

- save - добавление данных в очередь для последующей отправки в бд

Параметры:

- data - объект типа EmotionTrack

Модуль чтения кадров

Модуль реализует обертку над классом VideoCapture из библиотеки opencv.

Класс:

MyVideoCapture.

Методы:

- __init__ - инициализирует поток чтения данных.

Параметры:

- name - имя файла или ссылка на поток
- read - читает кадр из потока, не имеет входных параметров.

Стратегия модульного тестирования

Для выполнения блочного тестирования будет использована библиотека pytest.

Тестироваться будут два модуля:

1. Модуль работы с бд TrackingSaveQueue, метод save. Для проведения тестов был поднят докер контейнер с postgresql.
2. Модуль трекинга EmotionTracker, метод track.

Модули, которые будут протестированы другими методами:

- EmotionClassifier - протестирован специальными тестами
- MyAnnotator, MyVideoCapture - протестированы в ходе аттестационного тестирования.
- DetectionModel - протестирован в ходе аттестационного, интеграционного и нагрузочного тестирования.

Стратегия интеграционного тестирования

При интеграционном тестировании будет проверена совместная работа модуля детекции и трекинга вместе.

Для тестирования подготовлено несколько видео с различным числом людей в кадре. Цель тестирования - убедиться, что количество уникальных идентификаторов людей будет совпадать с количеством людей в кадре.

На вход алгоритму детекции подается сырой кадр, а алгоритму трекинга подается набор координат лиц, полученный из алгоритма детекции.

Для тестирования будет использована библиотека pytest.

Связи, которые будут протестированы другими стратегиями:

- Модуль отрисовки-модуль обработки данных - аттестационное тестирование
- Модуль чтения кадров-модуль обработки данных - аттестационное тестирование
- Модуль работы с бд-модуль обработки данных - аттестационное тестирование
- Модуль детекции-модуль обработки данных - аттестационное тестирование
- Модуль трекинга-модуль обработки данных - аттестационное тестирование
- Модуль классификации-модуль обработки данных - аттестационное тестирование
- Модуль работы с бд-база данных - блочное тестирование

Стратегия аттестационного тестирования

Аттестационное тестирование будет проводиться в формате формального тестирования.

В рамках аттестационного тестирования приложение будет запущено в production-режиме, в котором будет выведено видео, на которое будет выведена обработанная информация.

Тестирование необходимо проводить с камерой разрешения 640x480 и освещением 300-600 lux.

Стратегия специального тестирования

Тестирование модели классификации эмоций

Для проведения специального тестирования модели классификации будет использованы 6 моделей, обученные и протестированные в идентичных условиях и с одинаковыми гипер параметрами обучения. Измерение метрик будет производиться с помощью библиотеки sklearn.

Для обучения и тестирования были использованы несколько датасетов, каждый из которых содержал изображения, классифицированные на 7 эмоций(далее - тренировочный датасет).

1. Radbound Faces Database (RFD). Датасет состоит из 8000 фото. Была сделана фотосессия, где каждого человека сфотографили с 5 различных углов одновременно: 0, 45, 90(анфас), 135, 180, а также человек смотрел в три стороны: лево, прямо, центр.
2. KDEF. Датасет состоит из 4900 фото эмоций
3. FER 2013. Датасет из соревнования с kaggle, содержит 28000 фото размера 48x48
4. AffectNet. полная версия датасета содержит 420 тысяч фотографий, собранных с интернета, в нем присутствует довольно сильный дисбаланс классов. В данном решении используется урезанная версия данного датасета, которая содержит 38 тысяч фото.
5. Датасет, собранный в ЦИИ ПетрГУ, который включает в себя 28 фотографий сотрудников.



Тестовый датасет:

1. KDEF. 15% дасета вошли в тестовую часть.
2. Rfd. 15% дасета вошли в тестовую часть. Фронтальная часть датасета была совмещена с kdef.
3. AffectNet. 3500 тысяч фото.
4. Датасет, содержащий 28 фото сотрудников ЦИИ ПетрГУ.

Для выбора модели классификации было протестировано 6 моделей:

1. ResNext
2. MixNet версии l и xl
3. EfficientNet версии b1 и b3
4. Xception

Модель	Входные данные при обучении	Время на обработку одного кадра в секундах	Результаты тестирования на датасетах по метрике F1		
			RFD + KDEF	28 фото сотрудников ЦИИ	AffectNet
resnext_50d	AffectNet + RFD + KDEF	0.01	0.96	0.54	0.58
mixnet_l		0.02	0.93	0.35	Нет результатов
mxinet_xl		0.027	0.93	0.35	Нет результатов
efficientnet_b3		0.019	0.93	0.4	Нет результатов
efficientnet_b1		0.009	0.91	0.45	Нет результатов
xception		0.007	0.92	0.4	Нет результатов

Таблица 1. Сводная таблица тестирование моделей классификации

Стратегия нагрузочного тестирования

Цель нагрузочного тестирования - проверить качество модели детекции. Проведение тестирования будет происходить с помощью библиотеки pytest, а также заранее подготовленных фото и видео для исполнения тестов.

Для проведения нагрузочного тестирования будет выполнено две группы тестов:

1. Проверить количество найденных лиц на изображении при различном разрешении
2. Проверить скорость работы модели при различном разрешении

Детальный план тестов

Блочное тестирование

ID Теста	Б1
Объект тестирования	EmotionTracker, метод track
Тип теста	Позитивный
Цель тестирования	Проверка параметра минимального количества числа появлений объекта для присвоения идентификатора
Входные данные	Нижний порог до присвоения id = 5, количество появлений объекта = 3
Ожидаемый результат	Экземпляр трекера вернет пустой набор треков.

ID Теста	Б2
Объект тестирования	EmotionTracker, метод track
Тип теста	Позитивный
Цель тестирования	Проверка параметра минимального количества числа появлений объекта для присвоения идентификатора
Входные данные	Нижний порог до присвоения id = 5, количество появлений объекта = 6
Ожидаемый результат	Экземпляр трекера вернет 1 трек.

ID Теста	Б3
Объект тестирования	TrackingSaveQueue, метод save
Тип теста	Позитивный
Цель тестирования	Проверить функционал модуля работы с бд, выполняющий добавление объектов в базу данных. Проверка добавления одного объекта
Входные данные	{'track_id': 1, 'emotion': 'abs', 'camera_id': 1}
Ожидаемый результат	В таблицу будет добавлен один объект

ID Теста	Б4
Объект тестирования	TrackingSaveQueue, метод save
Тип теста	Позитивный
Цель тестирования	Проверить функционал модуля работы с бд, выполняющий добавление объектов в базу данных. Проверка добавления объектов, число которых не превышает размер батча
Входные данные	d = [{'track_id': 1, 'emotion': 'abs', 'camera_id': 1}, {'track_id': 2, 'emotion': 'abs', 'camera_id': 1},]
Ожидаемый результат	В таблицу будет добавлено 2 объекта

ID Теста	Б5
Объект тестирования	TrackingSaveQueue, метод save
Тип теста	Позитивный
Цель тестирования	Проверить функционал модуля работы с бд, выполняющий добавление объектов в базу данных. Проверка добавления объектов, число которых превышает размер батча
Входные данные	d = [{'track_id': 1, 'emotion': 'abs', 'camera_id': 1}, {'track_id': 2, 'emotion': 'abs', 'camera_id': 1}, {'track_id': 3, 'emotion': 'abs', 'camera_id': 1}, {'track_id': 4, 'emotion': 'abs', 'camera_id': 1},]

Ожидаемый результат	В таблицу будет добавлено 4 объекта
---------------------	-------------------------------------

ID Теста	Б6
Объект тестирования	TrackingSaveQueue, метод save
Тип теста	Негативный
Цель тестирования	Проверить функционал модуля работы с бд, выполняющий добавление объектов в базу данных. Проверка добавления объектов, тип данных которых не совпадает с прописанным в бд
Входные данные	<pre>d = [{ 'track_id': 1, 'emotion': b'abs', 'camera_id': 1 },]</pre>
Ожидаемый результат	Строка не будет добавлена в таблицу и выведется ошибка

ID Теста	Б7
Объект тестирования	TrackingSaveQueue, метод save
Тип теста	Негативный
Цель тестирования	Проверить функционал модуля работы с бд, выполняющий добавление объектов в базу данных. Проверка добавления объектов, не заполнив одно из полей.
Входные данные	<pre>d = [{ 'emotion': b'abs', 'camera_id': 1 },]</pre>
Ожидаемый результат	Произойдет исключение TypeError при инициализации объекта.

Интеграционное тестирование

ID Теста	И1
Объекты тестирования	EmotionTracker метод track, DetectionModel метод detect . Кадр подается методу detect, результат детекции подается методу track.

Тип теста	Положительный
Цель тестирования	Проверка совместной работы алгоритма трекинга и детекции
Входные данные	Видео с двумя людьми в кадре
Ожидаемый результат	Количество уникальных айди после обработки составит 2

ID Теста	И2
Объекты тестирования	EmotionTracker метод track, DetectionModel метод detect . Кадр подается методу detect, результат детекции подается методу track.
Тип теста	Положительный
Цель тестирования	Проверка совместной работы алгоритма трекинга и детекции
Входные данные	Видео с одним человеком в кадре
Ожидаемый результат	Количество уникальных айди после обработки составит 1

ID Теста	И3
Объекты тестирования	EmotionTracker метод track, DetectionModel метод detect . Кадр подается методу detect, результат детекции подается методу track.
Тип теста	Негативный
Цель тестирования	Проверка совместной работы алгоритма трекинга и детекции
Входные данные	Передается пустой массив детекций модулю трекинга
Ожидаемый результат	метод track вернет пустое массив треков.

Аттестационное тестирование

ID Теста	A1
Объект тестирования	MyAnnotator

Тип теста	Общий
Цель тестирования	Проверка отрисовки прямоугольника вокруг лица
Входные данные	Видеопоток с лицом одного человека
Ожидаемый результат	Прямоугольник будет отрисован вокруг лица

ID Теста	A2
Объект тестирования	MyAnnotator
Тип теста	Общий
Цель тестирования	Проверка отрисовки эмоции около лица человека
Входные данные	Видеопоток с лицом одного человека
Ожидаемый результат	Будет отрисована классифицированная эмоция

ID Теста	A3
Объект тестирования	DetectionModel
Тип теста	Общий
Цель тестирования	Проверка модели детекции при повороте лица +- 90 градусов по оси Z
Входные данные	Видеопоток с лицом одного человека
Ожидаемый результат	Лицо будет задетектировано во всех положения

ID Теста	A4
Объект тестирования	DetectionModel, EmotionTracker
Тип теста	Общий
Цель тестирования	Проверка модели детекции и трекинга при движении человека
Входные данные	Видеопоток с лицом одного человека
Ожидаемый результат	Id человека не меняется при перемещении

ID Теста	A5
Объект тестирования	DetectionModel, EmotionTracker
Тип теста	Общий
Цель тестирования	Проверка модели детекции и трекинга при наличии нескольких людей в кадре.
Входные данные	Видеопоток с лицом 2 людей
Ожидаемый результат	id 2 людей не будут меняться друг с другом

ID Теста	A6
Объект тестирования	DetectionModel, EmotionTracker
Тип теста	Общий
Цель тестирования	Проверка модели детекции и трекинга при наличии нескольких людей в кадре.
Входные данные	Видеопоток с лицом 2 людей
Ожидаемый результат	id 2 людей не будут меняться друг с другом

ID Теста	A7
Объект тестирования	DetectionModel
Тип теста	Общий
Цель тестирования	Закрыть нижнюю половину лица человека и проверить, что лицо все еще будет задетектировано
Входные данные	Видеопоток с лицом одного человека
Ожидаемый результат	Лицо будет задетектировано, если закрыть нижнюю половину лица

ID Теста	A8
Объект тестирования	DetectionModel, EmotionTracker
Тип теста	Общий
Цель тестирования	Сымитировать потерю детекции человека на несколько кадров путем полного закрытия лица, чтобы проверить,

	что при следующей детекции у этого человека айди не поменяется
Входные данные	Видеопоток с лицом одного человека
Ожидаемый результат	Айди человека не поменяется

ID Теста	A9
Объект тестирования	DetectionModel, EmotionTracker, EmotionClassifier
Тип теста	Общий
Цель тестирования	Проверка появления резких скачков у классифицированных эмоций.
Входные данные	Видеопоток с лицом одного человека, который улыбается в течении одной секунды.
Ожидаемый результат	Классифицированная эмоция должна быть радостью, а также не должно быть сделано ни одного перехода к другой эмоции.

ID Теста	A10
Объект тестирования	DetectionModel, EmotionTracker, EmotionClassifier, TrackingSaveQueue
Тип теста	Общий
Цель тестирования	Проверка записи данных в БД в production режиме.
Входные данные	Видеопоток с лицом одного человека, который улыбается в течении одной секунды. У модуля EmotionTracker параметр window_size=7. Количество кадров в секунду составляет 24.
Ожидаемый результат	В таблицу будет записано 3 строки с эмоцией "Happy".

ID Теста	A11
Объект тестирования	DetectionModel
Тип теста	Негативный
Цель тестирования	Проверка наличия ложно-положительных детекций.
Входные данные	Видеопоток, в котором в течение минуты не появляется

	ни одного человека.
Ожидаемый результат	В течение минуты не будет зарегистрировано ни одной детекции.

Специальное тестирование

ID Теста	C1
Объект тестирования	Модель классификации
Тип теста	Общий
Цель тестирования	Проверка точности работы модели
Входные данные	Модель resnext_50d, более подробно в таблице 1
Ожидаемый результат	Метрика F1 на датасете RFD+KDEF превысит 0.95

ID Теста	C2
Объект тестирования	Модель классификации
Тип теста	Общий
Цель тестирования	Проверка точности работы модели
Входные данные	Модель mixnet_l, более подробно в таблице 1
Ожидаемый результат	Метрика F1 на датасете RFD+KDEF превысит 0.95

ID Теста	C3
Объект тестирования	Модель классификации
Тип теста	Общий
Цель тестирования	Проверка точности работы модели
Входные данные	Модель mxnet_xl, более подробно в таблице 1
Ожидаемый результат	Метрика F1 на датасете RFD+KDEF превысит 0.95

ID Теста	C4
Объект тестирования	Модель классификации
Тип теста	Общий
Цель тестирования	Проверка точности работы модели
Входные данные	Модель efficientnet_b3, более подробно в таблице 1
Ожидаемый результат	Метрика F1 на датасете RFD+KDEF превысит 0.95

ID Теста	C5
Объект тестирования	Модель классификации
Тип теста	Общий
Цель тестирования	Проверка точности работы модели
Входные данные	Модель efficientnet_b1, более подробно в таблице 1
Ожидаемый результат	Метрика F1 на датасете RFD+KDEF превысит 0.95

ID Теста	C6
Объект тестирования	Модель классификации
Тип теста	Общий
Цель тестирования	Проверка точности работы модели
Входные данные	Модель xception, более подробно в таблице 1
Ожидаемый результат	Метрика F1 на датасете RFD+KDEF превысит 0.95

Нагрузочное тестирование

ID Теста	H1
Объект тестирования	DetectionModel, метод detect
Тип теста	Позитивный
Цель тестирования	Проверка минимального разрешения для поиска лиц
Входные данные	Изображение 600x600, 16 лиц на изображении.
Ожидаемый результат	Будут найдены все 16 лиц

ID Теста	H2
Объект тестирования	DetectionModel, метод detect
Тип теста	Позитивный
Цель тестирования	Проверка минимального разрешения для поиска лиц
Входные данные	Изображение 300x300, 16 лиц на изображении.
Ожидаемый результат	Будут найдены все 16 лиц

ID Теста	H3
Объект тестирования	DetectionModel, метод detect
Тип теста	Негативный
Цель тестирования	Проверка минимального разрешения для поиска лиц
Входные данные	Изображение 200x200, 16 лиц на изображении.
Ожидаемый результат	Будут найдены все 16 лиц

ID Теста	H4
Объект тестирования	DetectionModel, метод detect
Тип теста	Негативный
Цель тестирования	Проверка минимального разрешения для поиска лиц
Входные данные	Изображение 100x100, 16 лиц на изображении.

Ожидаемый результат	Будут найдены все 16 лиц
---------------------	--------------------------

ID Теста	H5
Объект тестирования	DetectionModel, метод detect
Тип теста	Позитивный
Цель тестирования	Проверить скорость работы модели на большом разрешении изображения
Входные данные	Изображение 1280x1280
Ожидаемый результат	Время на обработку одного кадра около 0.019с

ID Теста	H6
Объект тестирования	DetectionModel, метод detect
Тип теста	Позитивный
Цель тестирования	Проверить скорость работы модели на среднем разрешении изображения
Входные данные	Изображение 900x900
Ожидаемый результат	Время на обработку одного кадра около 0.014с

ID Теста	H7
Объект тестирования	DetectionModel, метод detect
Тип теста	Позитивный
Цель тестирования	Проверить скорость работы модели на маленьком разрешении изображения
Входные данные	Изображение 600x600
Ожидаемый результат	Время на обработку одного кадра около 0.011с

Покрытие кода тестами

Оценка покрытия тестами была выполнена с помощью библиотеки `pytest-cov`.

Результат оценки модулей для работы с нейронными сетями:

Name	Stmts	Miss	Cover

my_utils__init__.py	0	0	100%
my_utils\annotation.py	8	8	0%
my_utils\detection_model.py	16	0	100%
my_utils\emo_track.py	7	0	100%
my_utils\models.py	56	56	0%
my_utils\my_dataset.py	63	63	0%
my_utils\my_db.py	14	14	0%
my_utils\other.py	16	16	0%
my_utils\sort.py	208	64	69%
my_utils\streamer.py	32	32	0%
my_utils\tracker.py	116	68	41%

TOTAL	536	321	40%

Результат оценки модуля для работы с бд:

Name	Stmts	Miss	Cover

libs__init__.py	0	0	100%
libs\db__init__.py	0	0	100%
libs\db\connection.py	13	0	100%
libs\db\models.py	14	0	100%
libs\db\save_queue.py	62	11	82%
libs\db\tracking.py	31	6	81%

TOTAL	120	17	86%

Примеры тестов

Пример блочного теста проверки работы с БД:

```
def test_few_objects_more_than_batch():
    session.query(models.EmotionTable).delete()
    session.commit()
    d = [
        {'track_id': 1, 'emotion': 'abs', 'camera_id': 1},
        {'track_id': 2, 'emotion': 'abs', 'camera_id': 1},
        {'track_id': 3, 'emotion': 'abs', 'camera_id': 1},
        {'track_id': 4, 'emotion': 'abs', 'camera_id': 1},
    ]
    for i in range(len(d)):
        save_queue.save(EmotionTrack(**d[i]))

    time.sleep(0.2)
    items = session.query(models.EmotionTable).order_by(models.EmotionTable.track_id).all()
    assert all(map(lambda x: compare_object_with_dict(*x), zip(d, items)))
```

Пример интеграционного теста:

```
def test_two_persons():
    cap = cv2.VideoCapture(r'C:\Users\igors\dev\university\emotion_analysis\app_refactored\tests\two_humans.avi')
    tracker = EmotionTracker(min_hits=min_hits, max_age=10)
    unique_ids = set()
    while True:
        ret, frame = cap.read()
        if not ret:
            break

        frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

        model_out = model(frame)
        tracks = tracker.track(model_out)
        for i in range(len(tracks)):
            unique_ids.add(tracks[i][-1])

    assert len(unique_ids) == 2, "wrong amount of faces"
```

Журнал тестирования

Журнал блочного тестирования

ID теста	Дата	Результат	Отчет
Б1	10.12.2022	Пройден	

Б2	10.12.2022	Пройден	
Б3	10.12.2022	Не пройден	Строка не была добавлены в таблицу
Б4	10.12.2022	Не пройден	Строки не были добавлены в таблицу
Б5	10.12.2022	Пройден	
Б6	10.12.2022	Не пройден	Объект был добавлен в бд и тип данных поменялся на string
Б7	10.12.2022	Пройден	

Журнал интеграционного тестирования

ID теста	Дата	Результат	Отчет
И1	10.12.2022	Пройден	
И2	10.12.2022	Пройден	
И3	10.12.2022	Пройден	

Журнал аттестационного тестирования

ID теста	Дата	Результат	Отчет
А1	12.12.2022	Пройден	
А2	12.12.2022	Пройден	
А3	12.12.2022	Пройден	
А4	12.12.2022	Пройден	
А5	12.12.2022	Пройден	
А6	12.12.2022	Пройден	
А7	12.12.2022	Пройден	
А8	12.12.2022	Пройден	
А9	12.12.2022	Пройден	
А10	12.12.2022	Пройден	

A11	12.12.2022	Пройден	
-----	------------	---------	--

Журнал специального тестирования

ID теста	Дата	Результат	Отчет
C1	22.11.2022	Пройден	Метрика F1 равна 0.96
C2	22.11.2022	Не пройден	Метрика F1 равна 0.93
C3	22.11.2022	Не пройден	Метрика F1 равна 0.93
C4	22.11.2022	Не пройден	Метрика F1 равна 0.93
C5	22.11.2022	Не пройден	Метрика F1 равна 0.91
C6	22.11.2022	Не пройден	Метрика F1 равна 0.92

Журнал нагрузочного тестирования

ID теста	Дата	Результат	Отчет
H1	02.11.2022	Пройден	
H2	02.11.2022	Пройден	
H3	02.11.2022	Пройден	
H4	02.11.2022	Не пройден	Найдено только 15 лиц
H5	02.11.2022	Пройден	Скорость работы 0.0189
H6	02.11.2022	Пройден	Скорость работы 0.0134
H7	02.11.2022	Пройден	Скорость работы 0.0115

Журнал ошибок

№ отчета об ошибке	1
Номер теста	Б3

Объект тестирования	TrackingSaveQueue
Входные параметры	Запрос на добавление 1 объекта в бд
Ожидаемый результат	1 строка будет добавлена в таблицу
Фактический результат	Строка не была добавлена в таблицу
Воспроизводимость	Всегда
Статус	Решено, необходимо было поставить параметр у асинхронной очереди block=False

№ отчета об ошибке	2
Номер теста	Б4
Объект тестирования	TrackingSaveQueue
Входные параметры	Запрос на добавление нескольких объектов в бд, количество которых не превышает размер батча.
Ожидаемый результат	Строки будут добавлены в таблицу
Фактический результат	Строка была добавлена в таблицу и тип автоматически изменился на string
Воспроизводимость	Всегда
Статус	Решено, необходимо было поставить параметр у асинхронной очереди block=False

№ отчета об ошибке	3
Номер теста	Б6
Объект тестирования	TrackingSaveQueue
Входные параметры	Название эмоции в виде байт-строки
Ожидаемый результат	Строка не будет добавлена в таблицу и выведется ошибка
Фактический результат	Строка была добавлена в таблицу и тип автоматически изменился на string
Воспроизводимость	Всегда
Статус	Не решено

№ отчета об ошибке	4
Номер теста	C2
Объект тестирования	Модель классификации
Входные параметры	Модель mixnet_l, более подробно в таблице 1
Ожидаемый результат	Метрика F1 на датасете RFD+KDEF превысит 0.95
Фактический результат	Метрика F1 равна 0.93
Воспроизводимость	Всегда
Статус	Выбрана другая модель

№ отчета об ошибке	5
Номер теста	C4
Объект тестирования	Модель классификации
Входные параметры	Модель mxnet_xl, более подробно в таблице 1
Ожидаемый результат	Метрика F1 на датасете RFD+KDEF превысит 0.95
Фактический результат	Метрика F1 равна 0.93
Воспроизводимость	Всегда
Статус	Выбрана другая модель

№ отчета об ошибке	6
Номер теста	C4
Объект тестирования	Модель классификации
Входные параметры	Модель efficientnet_b3, более подробно в таблице 1
Ожидаемый результат	Метрика F1 на датасете RFD+KDEF превысит 0.95
Фактический результат	Метрика F1 равна 0.93
Воспроизводимость	Всегда
Статус	Выбрана другая модель

№ отчета об ошибке	7
Номер теста	C5
Объект тестирования	Модель классификации
Входные параметры	Модель efficientnet_b1, более подробно в таблице 1
Ожидаемый результат	Метрика F1 на датасете RFD+KDEF превысит 0.95
Фактический результат	Метрика F1 равна 0.91
Воспроизводимость	Всегда
Статус	Выбрана другая модель

№ отчета об ошибке	8
Номер теста	C6
Объект тестирования	Модель классификации
Входные параметры	Модель xception, более подробно в таблице 1
Ожидаемый результат	Метрика F1 на датасете RFD+KDEF превысит 0.95
Фактический результат	Метрика F1 равна 0.92
Воспроизводимость	Всегда
Статус	Выбрана другая модель

№ отчета об ошибке	9
Номер теста	H4
Объект тестирования	DetectionModel
Входные параметры	Изображение 100x100, 16 лиц на изображении.
Ожидаемый результат	Будут найдены все 16 лиц
Фактический результат	Найдены только 15 лиц
Воспроизводимость	Всегда

Статус	Не решено. Необходимо использовать более высокое разрешение.
--------	--

Заключение

В ходе тестирования приложения для классификации эмоций людей были проведены 34 теста, найдено 3 критические ошибки в модуле работы с бд, две из которых были исправлены.

Также в ходе специального тестирования была выбрана лучшая модель классификации по метрике F1 для данной задачи, ею является модель resnext50d.

В ходе нагрузочного тестирования было обнаружено минимальное разрешение входного изображения, при котором модель детектит все лица, которое равняется 200.

Приложения

Приложение 1. версии библиотек.

```
absl-py=1.3.0=pypi_0
accelerate=0.13.2=pypi_0
aiohttp=3.8.3=pypi_0
aiosignal=1.2.0=pypi_0
albumintations=1.3.0=pypi_0
asttokens=2.0.8=pypi_0
async-timeout=4.0.2=pypi_0
attrs=22.1.0=pypi_0
backcall=0.2.0=pyhd3eb1b0_0
blas=1.0=mkl
brotli=1.0.9=pypi_0
brotlipy=0.7.0=py38h2bbff1b_1003
ca-certificates=2022.07.19=haa95532_0
cachetools=5.2.0=pypi_0
catalyst=22.4=pypi_0
certifi=2022.9.24=py38haa95532_0
cffi=1.15.1=py38h2bbff1b_0
charset-normalizer=2.0.4=pyhd3eb1b0_0
colorama=0.4.5=py38haa95532_0
contourpy=1.0.5=pypi_0
cryptography=37.0.1=py38h21b164f_0
cudatoolkit=11.6.0=hc0ea762_10
cyclor=0.11.0=pypi_0
debugpy=1.5.1=py38hd77b12b_0
decorator=5.1.1=pyhd3eb1b0_0
efficientnet-pytorch=0.7.1=pypi_0
entrypoints=0.4=py38haa95532_0
```

executing=1.1.1=pypi_0
filelock=3.8.0=pypi_0
filterpy=1.4.5=pypi_0
fonttools=4.38.0=pypi_0
freetype=2.10.4=hd328e21_0
frozenlist=1.3.1=pypi_0
gevent=22.10.1=pypi_0
geventhttpclient=2.0.8=pypi_0
google-auth=2.13.0=pypi_0
google-auth-oauthlib=0.4.6=pypi_0
greenlet=1.1.3.post0=pypi_0
grpcio=1.41.0=pypi_0
huggingface-hub=0.10.1=pypi_0
hydra-slayer=0.4.0=pypi_0
idna=3.4=py38haa95532_0
imageio=2.22.2=pypi_0
importlib-metadata=5.0.0=pypi_0
intel-openmp=2021.4.0=haa95532_3556
ipykernel=6.15.2=py38haa95532_0
ipython=8.5.0=pypi_0
jedi=0.18.1=py38haa95532_1
joblib=1.2.0=pypi_0
jpeg=9e=h2bbff1b_0
jupyter_client=7.3.5=py38haa95532_0
jupyter_core=4.11.1=py38haa95532_0
kiwisolver=1.4.4=pypi_0
lerc=3.0=hd77b12b_0
libdeflate=1.8=h2bbff1b_5
libpng=1.6.37=h2a8f88b_0
libsodium=1.0.18=h62dcd97_0
libtiff=4.4.0=h8a3f274_0
libuv=1.40.0=he774522_0
libwebp=1.2.4=h2bbff1b_0
libwebp-base=1.2.4=h2bbff1b_0
lmbd=1.3.0=pypi_0
lz4-c=1.9.3=h2bbff1b_1
markdown=3.4.1=pypi_0
markupsafe=2.1.1=pypi_0
matplotlib=3.6.0=pypi_0
matplotlib-inline=0.1.6=py38haa95532_0
mkl=2021.4.0=haa95532_640
mkl-service=2.4.0=py38h2bbff1b_0
mkl_fft=1.3.1=py38h277e83a_0
mkl_random=1.2.2=py38hf11a4ad_0
multidict=6.0.2=pypi_0
nest-asyncio=1.5.5=py38haa95532_0
networkx=2.8.7=pypi_0
numpy=1.23.3=py38h3b20f71_0

numpy-base=1.23.3=py38h4da318b_0
oauthlib=3.2.2=pypi_0
opencv-python=4.6.0.66=pypi_0
opencv-python-headless=4.6.0.66=pypi_0
openssl=1.1.1q=h2bbff1b_0
packaging=21.3=pyhd3eb1b0_0
pandas=1.5.1=pypi_0
parso=0.8.3=pyhd3eb1b0_0
pickleshare=0.7.5=pyhd3eb1b0_1003
pillow=9.2.0=py38hdc2b20a_1
pip=22.2.2=py38haa95532_0
prettytable=3.4.1=pypi_0
prompt-toolkit=3.0.31=pypi_0
protobuf=3.19.6=pypi_0
psutil=5.9.3=pypi_0
psycopg2=2.9.4=pypi_0
pure_eval=0.2.2=pyhd3eb1b0_0
pyarrow=9.0.0=pypi_0
pyasn1=0.4.8=pypi_0
pyasn1-modules=0.2.8=pypi_0
pyparser=2.21=pyhd3eb1b0_0
pygments=2.13.0=pypi_0
pyopenssl=22.0.0=pyhd3eb1b0_0
pyparsing=3.0.9=py38haa95532_0
pysocks=1.7.1=py38haa95532_0
python=3.8.13=h6244533_0
python-dateutil=2.8.2=pyhd3eb1b0_0
python-rapidjson=1.9=pypi_0
python_abi=3.8=2_cp38
pytorch=1.12.1=py3.8_cuda11.6_cudnn8_0
pytorch-mutex=1.0=cuda
pytz=2022.5=pypi_0
pywavelets=1.4.1=pypi_0
pywin32=302=py38h2bbff1b_2
pyyaml=6.0=pypi_0
pymzmq=23.2.0=py38hd77b12b_0
qudida=0.0.4=pypi_0
regex=2022.9.13=pypi_0
requests=2.28.1=py38haa95532_0
requests-oauthlib=1.3.1=pypi_0
rsa=4.9=pypi_0
scikit-image=0.19.3=pypi_0
scikit-learn=1.1.2=pypi_0
scipy=1.9.3=pypi_0
seaborn=0.12.1=pypi_0
setuptools=63.4.1=py38haa95532_0
six=1.16.0=pyhd3eb1b0_1
sqlalchemy=1.4.42=pypi_0

sqlite=3.39.3=h2bbff1b_0
stack-data=0.5.1=pypi_0
stack_data=0.2.0=pyhd3eb1b0_0
tensorboard=2.10.1=pypi_0
tensorboard-data-server=0.6.1=pypi_0
tensorboard-plugin-wit=1.8.1=pypi_0
tensorboardx=2.5.1=pypi_0
threadpoolctl=3.1.0=pypi_0
tiff=2022.10.10=pypi_0
timm=0.6.11=pypi_0
tk=8.6.12=h2bbff1b_0
tokenizers=0.13.1=pypi_0
torchaudio=0.12.1=py38_cu116
torchtoolbox=0.1.8.2=pypi_0
torchvision=0.13.1=py38_cu116
tornado=6.2=py38h2bbff1b_0
tqdm=4.64.1=pypi_0
traitlets=5.5.0=pypi_0
transformers=4.23.1=pypi_0
tritonclient=2.26.0=pypi_0
typing_extensions=4.3.0=py38haa95532_0
urllib3=1.26.12=py38haa95532_0
vc=14.2=h21ff451_1
vs2015_runtime=14.27.29016=h5e58377_2
wcwidth=0.2.5=pyhd3eb1b0_0
werkzeug=2.2.2=pypi_0
wheel=0.37.1=pyhd3eb1b0_0
win_inet_pton=1.1.0=py38haa95532_0
wincertstore=0.2=py38haa95532_2
xz=5.2.6=h8cc25b3_0
yarl=1.8.1=pypi_0
zeromq=4.3.4=hd77b12b_0
zipp=3.10.0=pypi_0
zlib=1.2.13=h8cc25b3_0
zope-event=4.5.0=pypi_0
zope-interface=5.5.0=pypi_0
zstd=1.5.2=h19a0ad4_0