

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

Артур Харковчук Эдуардович

Отчет по учебному курсу "Тестирование ПО"

Тестирование сервисов системы промышленного мониторинга станков

Преподаватель:

к.ф.-м.н., доцент К. А. Кулаков

Петрозаводск 2019

1. Объект тестирования

1.1. Описание приложения

Объектом тестирования является система мониторинга, состоящая из сервисов и осуществляющая мониторинг за производственным оборудованием. Сервисы представляют набор модулей, работающих отдельно и выполняющих свои функции (отображение в веб, работа с различными наборами датчиков, модули видеокамеры, модули мониторинга событий, модули генерации сложных событий). Таким образом, будет происходить тестирование как всей системы в целом, так и каждого отдельного модуля в частности. Сервисы для пользователей в виде графиков, изображений и контекстной информации строятся в вебе.

Задачами тестирования видеосистемы является:

1. Наиболее полное представление сведений о системе конечному заказчику
2. Повышение качества модулей видеосистемы путем выполнения тестирования различных видов
3. Выявление ошибок и неоптимизированных участков в системе с целью их дальнейшего исправления

1.2. Описание сервисов

Сервис стриминга видео осуществляет подключение к камере по протоколу rtsp, получение кадров изображения и запись их zeroMQ по указанному порту для дальнейшего использования сервисами.

Сервис детектирования ошибки осуществляет получение кадров видео потока по указанному порту ZeroMQ, нахождение на кадре области с ошибкой, фильтрацию и распознавание с помощью средств нейронной сети. Найденный код ошибки публикуется в базе данных MongoDB.

1.3. Функции сервисов

Liborvideoprocessing – библиотека вспомогательных функций при подключении

1. loadconfig: чтение конфигураций из MongoDB, инициализация переменных

- def changeValues(dbObject) - задание значений дистанции и порогового значения модуля из базы данных
- def parseDB() - загрузка объекта базы данных, формирование адреса для подключения, определение сервиса

2. scan: сканирование MAC-адресов, полученных из MongoDB и их преобразование для в локальные IP-адреса для подключения с помощью OpenCV

- def mac_scan(hosts) - сканирование локальной сети (хост=**192.168.1.0/24**) с помощью nmap, парсинг результатов для вывода macs, ips
- def hosts_scan() - полное сканирование хостов локальной сети
- def mac_read(file) - чтение mac-адресов из файла
- def arp_scan() - сканирование локальной сети с помощью утилиты arp
- def mac2ip(mac) - преобразование mac-адреса в ip-адрес, сверка mac-адреса на входе со списком отсканированных mac-адресов и поиск соответствующего ip-адреса

3. selfdiagnosis: проверка доступности и работоспособности камер перед подключением к ним

- def testDevice(source) - попытка создания и подключения к источнику видео с помощью OpenCV

4. connect: определение параметров видеокамеры и подключение к ней

- def makeDir(storage, sid, camname) – создание директории для дальнейшего сохранения данных
- def setUp(address, dbObject) - извлечение параметров подключения из объекта базы данных, создание пути для сохранения фото/видео, создание видеорегистратора

5. logg: ведение логов с камер

- def flogging() функция ведения логов системных операций с последующим сохранением в txt-файл

6. events: работа с базой данных MongoDB, брокером сообщений RabbitMQ

- def get_obj(id) - получение объекта бд MongoDB (конфигурационный файл) с определенным id
- def send_in_rabbit(n, data, sid, raddr) - создание очереди и отправка данных в RabbitMQ
- def db_event(n, data, sid, raddr) - создание события и загрузка в коллекцию бд, заполнение полей payload в зависимости от сервиса/события

7. stream: передача потокового видео, соединение с другими сервисами видео

- def timenow() - преобразование текущей даты и времени в формат для сохранения в ФС и генерации событий
- def showfps() - отображение текущих кадров в секунду на изображении
- def read() - чтение кадров из потока в бесконечном while-цикле, передача кадра сервисам

8. save: сохранение файлов

- def timenow() - преобразование текущей даты и времени в формат для сохранения в ФС и генерации событий
- def convert_size(size_bytes) - преобразование размерностей файлов
- def save_video(frame, directory, sid, fps, width, height, start_time) - сохранение видео с указанием директории

or-video-streaming – сервис стриминга

Videostreaming: стриминг с rtsp потока в ZeroMQ

- def maininit() – создание нескольких потоков для стриминга
- def selfcheck(dbObject, service, i) – проверка параметров и стриминг потока

or-error-fixation - Модуль фиксации ошибки на экране

error: определение кода ошибки на экране ЧПУ

- search_conturs(frame) - проводит поиск четырехугольного замкнутого контура

- `normaliz(frame, directory, sid, camname, frames, width, height, approx)` - нормализует изображение
- `filter(image)` - фильтрует изображение кода ошибки для лучшего распознавания
- `recognition(frame, directory, sid, camname, frames, width, height)` - распознает текст с изображения

`or-head-control` – модуль определения присутствия на месте угловой головки и угла поворота угловой головки по видео-изображению

1.4. Архитектура программы

Модуль стриминга показан на рисунке 1.

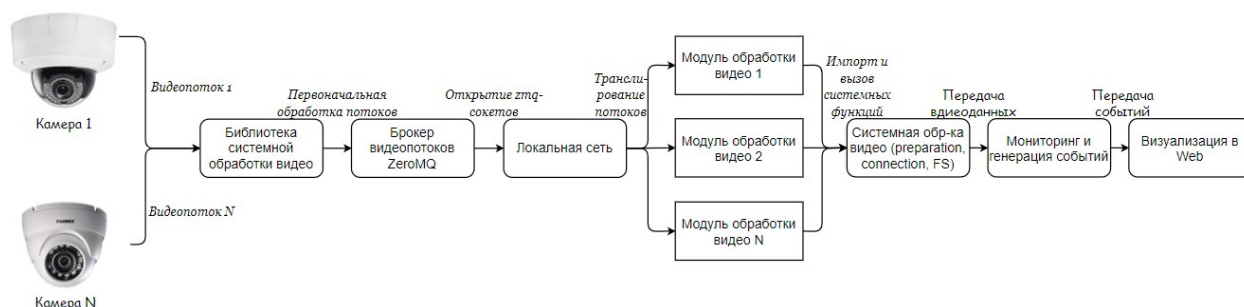


Рисунок 1. Работа модуля стриминга

1.5. Инструменты тестирования

Для тестирования будем использовать Pytest. Pytest — это основанная на Python среда тестирования, которая используется для написания и выполнения тестовых кодов.

1.6. Стратегия модульного тестирования

Первый вид тестирования, которому будет подвержена система, будет модульное тестирование. Данный вид тестирования будет применен к следующим функциям системы:

- 1) `arp_scan()`
- 2) `testDevice(source)`
- 3) `makeDir(storage, sid, camname)`
- 4) `showfps()`
- 5) `search_contours(frame)`

Более подробное описание проводимых тестов в разделе примеров тестов

1.7. Стратегия интеграционного тестирования

Второй этап - интеграционное тестирование. Для проведения интеграционного тестирования необходимо определить все возможные входные данные, соответствующие им ожидаемые результаты. Интеграционное тестирование будет проведено для следующих взаимодействий между функциями:

- 1) mac2ip(mac) и arp_scan()
- 2) read(sources, port) и showfps()
- 3) save_video(frame, directory, sid, fps, width, height, start_time) и makeDir(storage, sid, camname)
- 4) normaliz(frame, directory, sid, camname, frames, width, height, approx) и search_conturs(frame)

1.8. Стратегия аттестационного тестирования

В ходе аттестационного тестирования будет протестирована работоспособность приложения и его возможность осуществлять заявленный функционал. Будут проверяться все функциональные требования.

Аттестационное тестирование будет проводиться методом «живого человека». В роли такого человека выступает сам автор тестирования.

Тестирующий человек, по заранее заданным инструкциям, производит требуемые действия и сверяется с заранее заданными результатами. Тест считается пройденным, если ожидаемый результат совпадает с фактическим результатом. В противном случае тест считается не пройденным.

1.9. Стратегия нагрузочного тестирования

Нагрузочное тестирование – это процесс умышленной нагрузки системы, с целью определения показателей производительности, времени отклика, проверки соответствия требованиям, которые были предъявлены к данной системе или отдельному устройству. Целью данного тестирования является оценка производительности и работоспособности тестируемого модуля. Нагрузочные тесты:

- 1) Проверка работоспособности сервиса стриминга со 2, 10, 50 камерами;
- 2) Проверка работоспособность сервиса нахождения кода ошибки, если на вход подается изображение 1000 на 1000, 5000 на 5000 и 10000 на 10000 пикселей.

1.10. Критерий прохождения теста

Тест считается успешно пройденным, если ожидаемый и фактический результаты совпадают. Если тест завершился неудачей и тест реализован правильно, то производится заключение о найденной ошибке. Тестирование считается пройденным, если во время его прохождения не выявлено критических ошибок.

1.11. Оборудование для проведения тестирования

Для проведения тестирования используется настольный компьютер с установленной программой реализацией стенда и необходимые пакеты для работы системы мониторинга.

1.12. Критерий приостановки тестов

Тестирование должно быть приостановлено, если количество не пройденных тестов превысит 10% от их общего количества. Тестирование должно быть приостановлено при обнаружении критических ошибок.

1.13. Критерий возобновления тестирования

Тестирование возобновляется после исправления ошибок, выявленных при предыдущем тестировании.

2. Тестирование

2.1. Модульное тестирование

Тест	1
Цель теста (описание)	Проверить работоспособность функции по нахождению мак и ip адресов
Тип теста	Позитивный
Объект тестирования (модуль, интерфейс или функциональность)	Функция - agr_scan()
Входные параметры	-
Косвенные данные	Установлена системная функция agr
Ожидаемый результат	Два массива(с мак адресами и с ip адресами)

Тест	2
Цель теста (описание)	Проверить поведение функции при отсутствие системной функции agr
Тип теста	Негативный
Объект тестирования (модуль, интерфейс или функциональность)	Функция - agr_scan()
Входные параметры	-
Косвенные данные	Не установлена системная функция agr
Ожидаемый результат	Ошибка: системная функция agr не установлена

Тест	3
Цель теста (описание)	Попытка создания и подключения к источнику видео с помощью OpenCV
Тип теста	Позитивный
Объект тестирования (модуль, интерфейс или функциональность)	Функция - testDevice(source)
Входные параметры	Путь до файла формата mp4, или сформированная строка для rtsp соединения
Косвенные данные	нет
Ожидаемый результат	Значение True так как подключение прошло успешно

Тест	4
Цель теста (описание)	Попытка создания и подключения к несуществующему источнику видео с помощью OpenCV
Тип теста	Негативный
Объект тестирования (модуль, интерфейс или функциональность)	Функция - testDevice(source)
Входные параметры	Путь до не существующего файла или файла формата не mp4
Косвенные данные	нет
Ожидаемый результат	Значение False так как подключение не произошло

Тест	5
Цель теста (описание)	Создание директории для дальнейшего сохранения данных
Тип теста	Позитивный

Объект тестирования (модуль, интерфейс или функциональность)	Функция - makeDir(storage, sid, camname)
Входные параметры	Storage – Путь до директории где находится хранилище директорий для сохранения Sid – id сервиса Camname – имя камеры для создания индивидуальной директории для записи данной камеры
Косвенные данные	нет
Ожидаемый результат	Созданная директория

Тест	6
Цель теста (описание)	Проверить работоспособность функции при запрете на запись файла
Тип теста	Негативный
Объект тестирования (модуль, интерфейс или функциональность)	Функция - makeDir(storage, sid, camname)
Входные параметры	Storage – Путь до директории где находится хранилище директорий для сохранения(где запрещено создание файлов) Sid – id сервиса Camname – имя камеры для создания индивидуальной директории для записи данной камеры
Косвенные данные	нет
Ожидаемый результат	Ошибка: файл не может быть создан, прекращение работы программы

Тест	7
------	---

Цель теста (описание)	Проверить работоспособность функции при длинном имени файла
Тип теста	Негативный
Объект тестирования (модуль, интерфейс или функциональность)	Функция - makeDir(storage, sid, camname)
Входные параметры	Storage – Путь до директории где находится хранилище директорий для сохранения Sid – id сервиса Camname – имя камеры для создания индивидуальной директории для записи данной камеры(длина имени превышает 256 символов)
Косвенные данные	нет
Ожидаемый результат	Ошибка: файл не может быть создан из-за неподходящего имени, прекращение работы программы

Тест	8
Цель теста (описание)	Проверить работоспособность функции showfps(), которая накладывает на кадр количество fps
Тип теста	Позитивный
Объект тестирования (модуль, интерфейс или функциональность)	Функция - showfps(frame, fps)
Входные параметры	Frame – кадр Fps – значение количества fps
Косвенные данные	нет
Ожидаемый результат	Кадр с нанесенным количеством fps

Тест	9
Цель теста (описание)	Проверить работоспособность

	функции showfps(), при передаче в качества кадра не изображение
Тип теста	Негативный
Объект тестирования (модуль, интерфейс или функциональность)	Функция - showfps(frame, fps)
Входные параметры	Frame – значение типа str Fps – значение количества fps
Косвенные данные	нет
Ожидаемый результат	Ошибка: frame не является изображением, прекращение работы программы

Тест	10
Цель теста (описание)	Провести поиск четырехугольного замкнутого контура
Тип теста	Позитивный
Объект тестирования (модуль, интерфейс или функциональность)	Функция - search_conturs(frame)
Входные параметры	Frame – изображение
Косвенные данные	нет
Ожидаемый результат	двухмерный массив – контур части изображения

Тест	11
Цель теста (описание)	Провести поиск четырехугольного замкнутого контура на изображении одного цвета
Тип теста	Негативный
Объект тестирования (модуль, интерфейс или функциональность)	Функция - search_conturs(frame)

Входные параметры	Frame – изображение(Черного цвета)
Косвенные данные	нет
Ожидаемый результат	Значение – None(контур не найден)

2.2. Интеграционное тестирование

Тест	101
Цель теста (описание)	Проверить работоспособность функции mac2ip(mac)
Тип теста	Позитивный
Объект тестирования (модуль, интерфейс или функциональность)	Функция - mac2ip(mac) Вызывает - arp_scan()
Входные параметры	Mac – мак адрес камеры
Косвенные данные	нет
Ожидаемый результат	Значение Ip камеры

Тест	102
Цель теста (описание)	Проверить работоспособность функции mac2ip(mac) при неверном mac
Тип теста	Негативный
Объект тестирования (модуль, интерфейс или функциональность)	Функция - mac2ip(mac) Вызывает - arp_scan()
Входные параметры	Mac – неверный мак адрес камеры
Косвенные данные	нет
Ожидаемый результат	None – значение камеры не найдено, выполнение программы закончено

Тест	103
Цель теста (описание)	Проверить работоспособность функции read(sources, port), чтение кадров из потока в бесконечном while-цикле, передача кадра сервисам
Тип теста	Позитивный
Объект тестирования (модуль, интерфейс или функциональность)	Функция - read(sources, port) Вызывает - showfps()
Входные параметры	sources - Поток на подключение port – порт подключения для zeroMQ
Косвенные данные	нет
Ожидаемый результат	Создание потока zeroMQ с кадрами видео потока

Тест	104
Цель теста (описание)	Проверить работоспособность функции read(sources, port), чтение кадров из несуществующего потока в бесконечном while-цикле, передача кадра сервисам
Тип теста	Негативный
Объект тестирования (модуль, интерфейс или функциональность)	Функция - read(sources, port) Вызывает - showfps()
Входные параметры	sources - Поток на подключение(несуществующий) port – порт подключения для zeroMQ
Косвенные данные	нет
Ожидаемый результат	Ожидание подключения к потоку

Тест	105
Цель теста (описание)	Проверить работоспособность функции save_video, сохранение видео с указанием директории
Тип теста	Позитивный
Объект тестирования (модуль, интерфейс или функциональность)	Функция - save_video(frame, directory, sid, fps, width, height, start_time) Вызывает - makeDir(storage, sid, camname)
Входные параметры	Frame – изображение Directory – директория где находятся записи камер Sid – айди сервиса Fps -количество кадров в секунду width, height – ширина/высота изображения start_time – время начала записи
Косвенные данные	нет
Ожидаемый результат	создание файла в указанной директории и передача ссылки на поток для записи в файл

Тест	106
Цель теста (описание)	Проверить работоспособность функции normaliz, провести нормализации изображения
Тип теста	Позитивный
Объект тестирования (модуль, интерфейс или функциональность)	Функция - normaliz(frame, directory, sid, camname, frames, width, height, approx) Вызывает - search_conturs(frame)
Входные параметры	Frame – изображение Directory – директория где находятся записи камер Sid – айди сервиса camname -название камеры width, height – ширина/высота изображения approx – двухмерный массив –

	контур части изображения
Косвенные данные	нет
Ожидаемый результат	нормализованное изображение кода ошибки

2.3. Аттестационное тестирование

Тест	1001
Цель теста (описание)	Проверить работоспособность сервиса стриминга
Тип теста	Позитивный
Объект тестирования (модуль, интерфейс или функциональность)	Сервис or-video-streaming
Входные параметры	В базу данных занесены следующие верные параметры: Количество используемых видео камер Ip адрес, мак адрес и порт трансляции каждой из них Тип видеокамеры Источник видео Протокол камеры Логин камеры Пароль камеры Количество кадров в секунду Количество используемых кадров в секунду
Косвенные данные	нет
Ожидаемый результат	Создаются видео потоки к которым могут подключаться сервисы

Тест	1002
Цель теста (описание)	Проверить работоспособность сервиса стриминга с неверными параметрами протокола
Тип теста	Негативный

Объект тестирования (модуль, интерфейс или функциональность)	Сервис or-video-streaming
Входные параметры	<p>В базу данных занесены следующие параметры:</p> <p>Количество используемых видео камер</p> <p>Ip адрес, мак адрес и порт трансляции каждой из них</p> <p>Тип видеокамеры</p> <p>Источник видео</p> <p>Протокол камеры(заносятся протокол RDP)</p> <p>Логин камеры</p> <p>Пароль камеры</p> <p>Количество кадров в секунду</p> <p>Количество используемых кадров в секунду</p>
Косвенные данные	нет
Ожидаемый результат	Сервис не может подключиться к камерам, работа сервисов завершается

Тест	1003
Цель теста (описание)	Проверить работоспособность сервиса стриминга с неверными параметрами кадров в секунду
Тип теста	Негативный
Объект тестирования (модуль, интерфейс или функциональность)	Сервис or-video-streaming
Входные параметры	<p>В базу данных занесены следующие параметры:</p> <p>Количество используемых видео камер</p> <p>Ip адрес, мак адрес и порт трансляции каждой из них</p> <p>Тип видеокамеры</p> <p>Источник видео</p> <p>Протокол камеры</p> <p>Логин камеры</p>

	Пароль камеры Количество кадров в секунду Количество используемых кадров в секунду (указывается значение больше количества кадров в секунду)
Косвенные данные	нет
Ожидаемый результат	Ошибка, сервис заканчивает работу

Тест	1004
Цель теста (описание)	Проверить работоспособность сервиса стриминга с неверными параметрами, где пароль от камеры не верен
Тип теста	Негативный
Объект тестирования (модуль, интерфейс или функциональность)	Сервис or-video-streaming
Входные параметры	В базу данных занесены следующие параметры: Количество используемых видео камер Ip адрес, мак адрес и порт трансляции каждой из них Тип видеокамеры Источник видео Протокол камеры(заносятся протокол RDP) Логин камеры Пароль камеры(указывается неверный пароль) Количество кадров в секунду Количество используемых кадров в секунду
Косвенные данные	нет
Ожидаемый результат	Сервис не может подключиться к камерам, работа сервисов завершается

Тест	1005
Цель теста (описание)	Проверить работоспособность сервиса нахождения кода ошибки
Тип теста	Позитивный
Объект тестирования (модуль, интерфейс или функциональность)	Сервис or-error-fixation
Входные параметры	<p>В базу данных занесены следующие параметры:</p> <p>Угол поворота камеры</p> <p>Ip адрес, мак адрес и порт камеры</p> <p>Тип видеокамеры</p> <p>Протокол камеры</p> <p>Логин камеры</p> <p>Пароль камеры</p> <p>Количество кадров в секунду</p> <p>Количество используемых кадров в секунду (указывается значение больше количества кадров в секунду)</p> <p>Директория для сохранения видео потоков</p>
Косвенные данные	Созданный сервисом стриминга ZeroMQ поток
Ожидаемый результат	Сервис распознает коды ошибки и сохраняет их в базу данных

Тест	1006
Цель теста (описание)	Проверить работоспособность сервиса нахождения кода ошибки, если угол поворота камеры указан неправильно
Тип теста	Негативный
Объект тестирования (модуль, интерфейс или функциональность)	Сервис or-error-fixation

Входные параметры	<p>В базу данных занесены следующие параметры:</p> <p>Угол поворота камеры(указывается значение 7777)</p> <p>Ip адрес, мак адрес и порт камеры</p> <p>Тип видеокамеры</p> <p>Протокол камеры</p> <p>Логин камеры</p> <p>Пароль камеры</p> <p>Количество кадров в секунду</p> <p>Количество используемых кадров в секунду (указывается значение больше количества кадров в секунду)</p> <p>Директория для сохранения видео потоков</p>
Косвенные данные	Созданный сервисом стриминга ZeroMQ поток
Ожидаемый результат	Ошибка, неверный угол поворота, сервис заканчивает работу

2.4. Нагрузочное тестирование

Тест	10001
Цель теста (описание)	Проверить работоспособность сервиса стриминга с большим количеством камер
Объект тестирования (модуль, интерфейс или функциональность)	Сервис or-video-streaming
Входные параметры	<p>В базу данных занесены следующие параметры:</p> <p>Количество используемых видео камер(используются 2 камер)</p> <p>Ip адрес, мак адрес и порт трансляции каждой из них</p> <p>Тип видеокамеры</p> <p>Источник видео</p> <p>Протокол камеры</p> <p>Логин камеры</p> <p>Пароль камеры</p> <p>Количество кадров в секунду</p> <p>Количество используемых кадров в секунду (указывается значение</p>

	больше количества кадров в секунду)
Косвенные данные	нет
Ожидаемый результат	Сервис отработает в стандартном режиме

Тест	10002
Цель теста (описание)	Проверить работоспособность сервиса стриминга с большим количеством камер
Объект тестирования (модуль, интерфейс или функциональность)	Сервис or-video-streaming
Входные параметры	<p>В базу данных занесены следующие параметры:</p> <p>Количество используемых видео камер(используются 10 камер)</p> <p>Ip адрес, мак адрес и порт трансляции каждой из них</p> <p>Тип видеокамеры</p> <p>Источник видео</p> <p>Протокол камеры</p> <p>Логин камеры</p> <p>Пароль камеры</p> <p>Количество кадров в секунду</p> <p>Количество используемых кадров в секунду (указывается значение больше количества кадров в секунду)</p>
Косвенные данные	нет
Ожидаемый результат	Сервис отработает в стандартном режиме

Тест	10003
Цель теста (описание)	Проверить работоспособность сервиса стриминга с большим количеством камер

Объект тестирования (модуль, интерфейс или функциональность)	Сервис or-video-streaming
Входные параметры	<p>В базу данных занесены следующие параметры:</p> <p>Количество используемых видео камер(используются 50 камер)</p> <p>Ip адрес, мак адрес и порт трансляции каждой из них</p> <p>Тип видеокамеры</p> <p>Источник видео</p> <p>Протокол камеры</p> <p>Логин камеры</p> <p>Пароль камеры</p> <p>Количество кадров в секунду</p> <p>Количество используемых кадров в секунду (указывается значение больше количества кадров в секунду)</p>
Косвенные данные	нет
Ожидаемый результат	Сервис отработает в стандартном режиме

Тест	10004
Цель теста (описание)	Проверить работоспособность сервиса нахождения кода ошибки, если на вход подается изображение большого разрешения
Объект тестирования (модуль, интерфейс или функциональность)	Сервис or-error-fixation
Входные параметры	<p>В базу данных занесены следующие параметры:</p> <p>Угол поворота камеры</p> <p>Ip адрес, мак адрес и порт камеры</p> <p>Тип видеокамеры</p> <p>Протокол камеры</p> <p>Логин камеры</p> <p>Пароль камеры</p> <p>Количество кадров в секунду</p> <p>Количество используемых кадров в секунду (указывается значение</p>

	<p>больше количества кадров в секунду) Директория для сохранения видео потоков</p>
Косвенные данные	Созданный сервисом стриминга ZeroMQ поток с изображениями 1000 на 1000 пикселей
Ожидаемый результат	Сервис распознает коды ошибки и сохраняет их в базу данных

Тест	10005
Цель теста (описание)	Проверить работоспособность сервиса нахождения кода ошибки, если на вход подается изображение большого разрешения
Объект тестирования (модуль, интерфейс или функциональность)	Сервис or-error-fixation
Входные параметры	<p>В базу данных занесены следующие параметры: Угол поворота камеры Ip адрес, мак адрес и порт камеры Тип видеокамеры Протокол камеры Логин камеры Пароль камеры Количество кадров в секунду Количество используемых кадров в секунду (указывается значение больше количества кадров в секунду) Директория для сохранения видео потоков</p>
Косвенные данные	Созданный сервисом стриминга ZeroMQ поток с изображениями 5000 на 5000 пикселей
Ожидаемый результат	Сервис распознает коды ошибки и сохраняет их в базу данных(отстает от онлайн режима, из-за нехватки вычислительных ресурсов)

Тест	10006
------	-------

Цель теста (описание)	Проверить работоспособность сервиса нахождения кода ошибки, если на вход подается изображение большого разрешения
Объект тестирования (модуль, интерфейс или функциональность)	Сервис or-error-fixation
Входные параметры	В базу данных занесены следующие параметры: Угол поворота камеры Ip адрес, мак адрес и порт камеры Тип видеокамеры Протокол камеры Логин камеры Пароль камеры Количество кадров в секунду Количество используемых кадров в секунду (указывается значение больше количества кадров в секунду) Директория для сохранения видео потоков
Косвенные данные	Созданный сервисом стриминга ZegoMQ поток с изображениями 10000 на 10000 пикселей
Ожидаемый результат	Сервис распознает коды ошибки и сохраняет их в базу данных(значительно отстает от онлайн режима, из-за нехватки вычислительных ресурсов)

3. Журнал тестирования

Номер теста	Результат
1	Пройден
2	Ошибка(Отчет 1)
3	Пройден
4	Пройден
5	Пройден
6	Пройден
7	Пройден

8	Пройден
9	Ошибка(Отчет 2)
10	Пройден
11	Пройден
101	Пройден
102	Пройден
103	Пройден
104	Пройден
105	Пройден
106	Пройден
1001	Пройден
1002	Пройден
1003	Пройден
1004	Пройден
1005	Пройден
1006	Ошибка(Отчет 3)
10001	Пройден
10002	Пройден
10003	Пройден
10004	Пройден
10005	Пройден
10006	Пройден

3.1. Покрытие кода тестами

Всего строк кода – 890

Количество строк покрытыми тестами – 368

Тестовое покрытие – $(368/890)*100 = 41\%$

3.2 Пример тестов

```

import sys
sys.path.insert(0, '../liborvideoprocessing/connection/')
sys.path.insert(0, '../liborvideoprocessing/preparation/')
sys.path.insert(0, '../')
import selfdiagnosis
import pytest
import time
import cv2
import threading
import zmq

def test_testDevice():
    status = selfdiagnosis.testDevice(['people-walking.mp4'])
    assert status == True, "Поток создается и существует"

def test_testDevice_2():
    status = selfdiagnosis.testDevice('1234.mp4')
    assert status == False, "Потока не существует"

```

```

def test_arp_scan():
    macs, ips = scan.arp_scan()
    assert type(macs) == list, "Функция должна возвращать список мак адресов"
    assert type(ips) == list, "Функция должна возвращать список айпи адресов"

```

```

def test_mac2ip():
    ip = scan.mac2ip('de:da:2c:00:4d:00')
    assert ip == '192.168.1.182', "Функция должна возвращать верное айпи"

```

```

def test_mac2ip_2():
    ip = scan.mac2ip('de:da:2c:00:4d:0d')
    assert ip == 0, "Функция должна возвращать ноль в случае неверного айпи"

```

```

def test_read():
    image = cv2.imread('./test.png')

    cap = cv2.VideoCapture('people-walking.mp4')
    fps = cap.get(cv2.CAP_PROP_FPS)
    #stream.read('testfile.mp4', cap, './', fps, 1, './', '234sdhgh', 'rtyuifgh4t', 'cam1', 4567)
    my_thread = threading.Thread(target=stream.read, args=('testfile.mp4', cap, './', fps, 1, './', '234sdhgh', 'rtyuifgh4t', 'cam1', 4567))
    my_thread.start()

    context = zmq.Context()
    src = context.socket(zmq.PULL)
    tcpstring = "tcp://127.0.0.1:4567"
    src.connect(tcpstring)

    msg = src.recv_pyobj()
    frame = msg['frame']

    assert type(frame) == type(image), "Функция должна возвращать frame"

```



```
def test_makeDir_2():
    shutil.rmtree('10', ignore_errors=True)
    d = connect.makeDir('1', '0', 'cam1')
    assert os.path.isdir(d)
```

4. Отчет об ошибках

Отчет 1

Тест номер 2

Описание ошибки: нет проверки на наличия функции agr, поэтому сервис падает, а не завершается.

Ожидаемый результат: Требуется чтобы сервис уведомлял пользователя о не нахождении функции и корректно завершился

Отчет 2

Тест номер 9

Описание ошибки: нет проверки на то что передается изображение, поэтому сервис падает а не корректно завершается.

Ожидаемый результат: Требуется чтобы сервис уведомлял пользователя о том что переданное не является изображением и корректно завершился сервис

Отчет 3

Тест номер 1006

Описание ошибки: нет проверки на то, что пользователь указал одно из четырёх положений, поэтому сервис может найти угол и завершается с ошибкой.

Ожидаемый результат: Требуется уведомить пользователя о неверном введенном угле и корректно завершить программу.

5. Результаты

Данное тестирование показало, что код написан с учетом, что им будет пользоваться “умный пользователь” и не достаточно много проверок на входные данные и данные между внутренними функциями. Требуется устранить недостатки в ходе разработки сервисов.

6. Заключение

Во время тестирования приложения было выявлено три ошибки, представленных в отчётах об ошибках №1–3. Все найденных ошибок все являются критическими и должны быть исправлены. По результатам тестирования необходимо исправить найденные ошибки и повторно протестировать сервисы стриминга и нахождения ошибки.