

# Разработка системы поиска аномалий при видеонаблюдении

Выполнил: А.Н. Шарганов

Научный руководитель: ст.преп. Я.А. Кириленко

Рецензент: старший консультант ООО "САП Лабс СНГ" В.М. Палкин

# Введение

- Обеспечение безопасности
- Камеры видеонаблюдения
- Проблемы при использовании камер видеонаблюдения
- Поиск аномалий как решение



# Постановка задачи

**Целью** данной работы является разработка системы для поиска аномалий при видеонаблюдении.

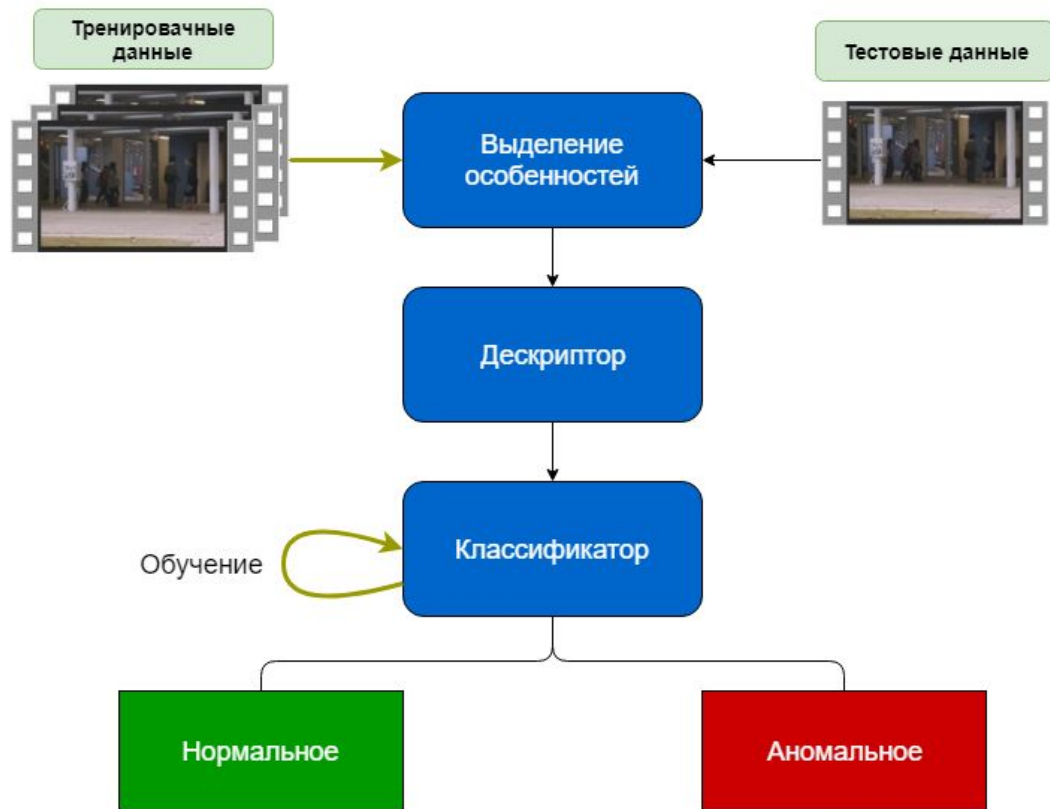
1. Разработать требования
2. Разработать архитектуру системы
3. Разработать прототип требуемой системы
4. Провести апробацию

## Требования к системе

- Детектировать аномалии
- Графический интерфейс
  - Выделение аномалии на кадре
  - Просмотр видео с камер видеонаблюдения
  - Просмотр истории выявленных аномалий
- Вспомогательный инструмент

# Подходы к детекции аномалий

- Hand-crafted features
  - HOF
  - HOG
  - Траектории
- Sparse coding
- Глубокое обучение

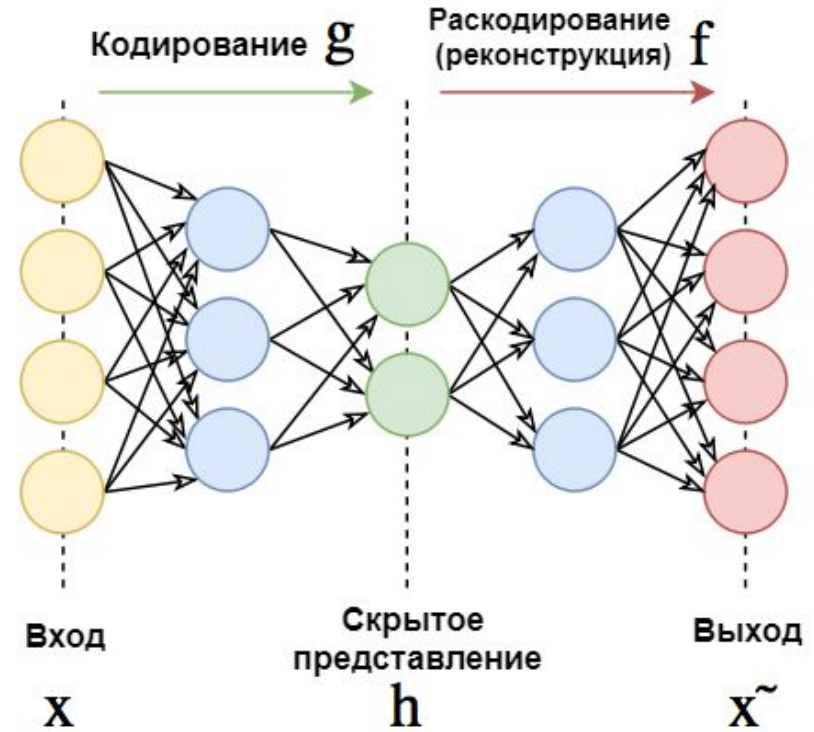


# Глубокое обучение. Автокодировщик

Архитектура нейронных сетей, копирующая свой вход на выход через скрытое представление.

**Цель:** минимизировать разницу между входом и выходом.

**Результат:** скрытое представление содержит в себе значимые “особенности” входных данных.



# Регулярность видео

**Идея:** обычные события будут иметь низкую ошибку реконструкции, аномальные – высокую.

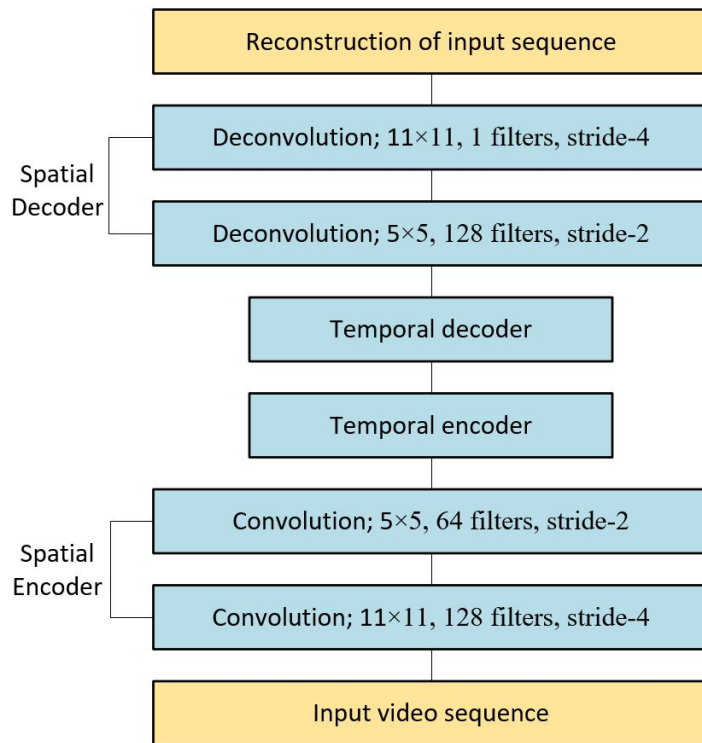
- Использование автокодировщиков на основе сверточных сетей
  - “Learning Temporal Regularity in Video Sequences” (2015)

**Возможные улучшения:** использование моделей нейронных сетей, предназначенных для исследования временных зависимостей.

# Использование RNN-LSTM

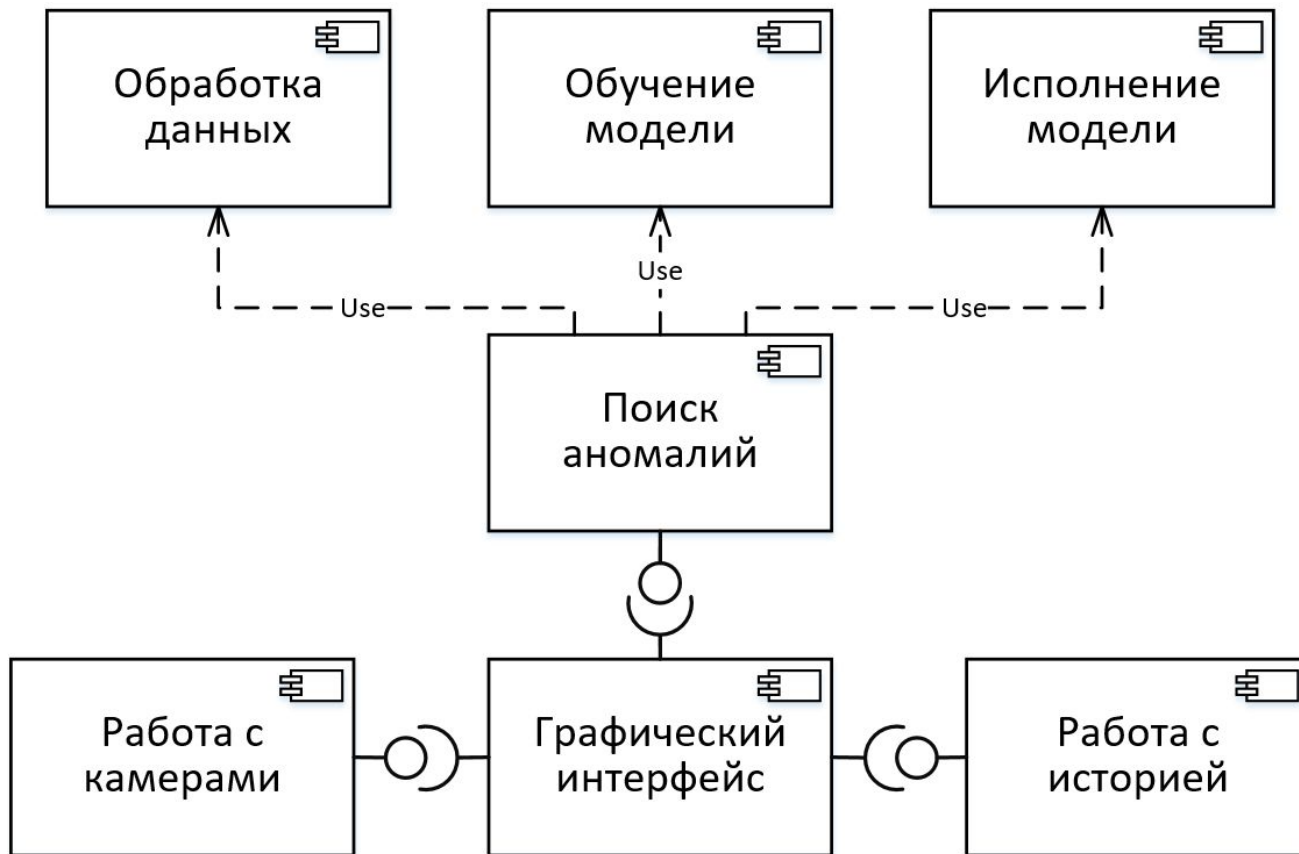
“Abnormal Event Detection in Videos using Spatiotemporal Autoencoder” (2017)

- Кодировывает пространственные особенности за счет свертки
- Учитывает временные зависимости благодаря LSTM
- Имеет хорошую скорость при тестировании на GPU: ~ 70 fps Nvidia Geforce GTX 1060





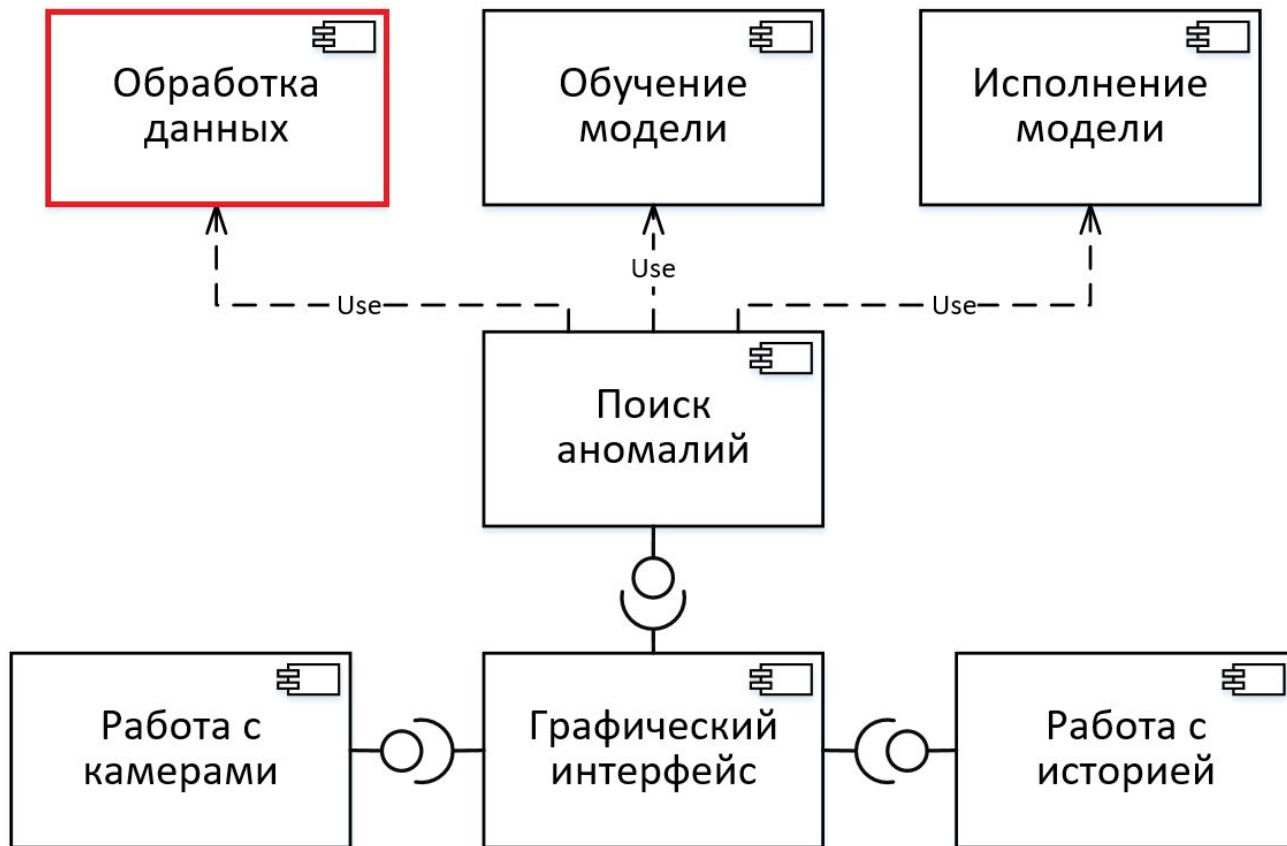
# Архитектура системы



# Реализация обучения и исполнения модели

- Обучение модели реализовано с помощью библиотеки Keras на языке Python
- Исполнение обученной модели реализовано с помощью библиотеки TensorFlow на языке C++

# Архитектура системы



# Обработка данных

- Фильтрация
  - Удаление последовательностей кадров без значительных изменений
- Нормализация (стандартизация)
  - Сжатие кадров к размеру 224 x 224
  - Вычитание среднего
- Аугментация
  - Последовательности вида: каждый кадр, каждый второй, каждый третий

# Применение в офисе. Апробация алгоритма

- Размер обучающей выборки: 17000 изображений
- Размер тестовой выборки : 10000 изображений

Precision: 70

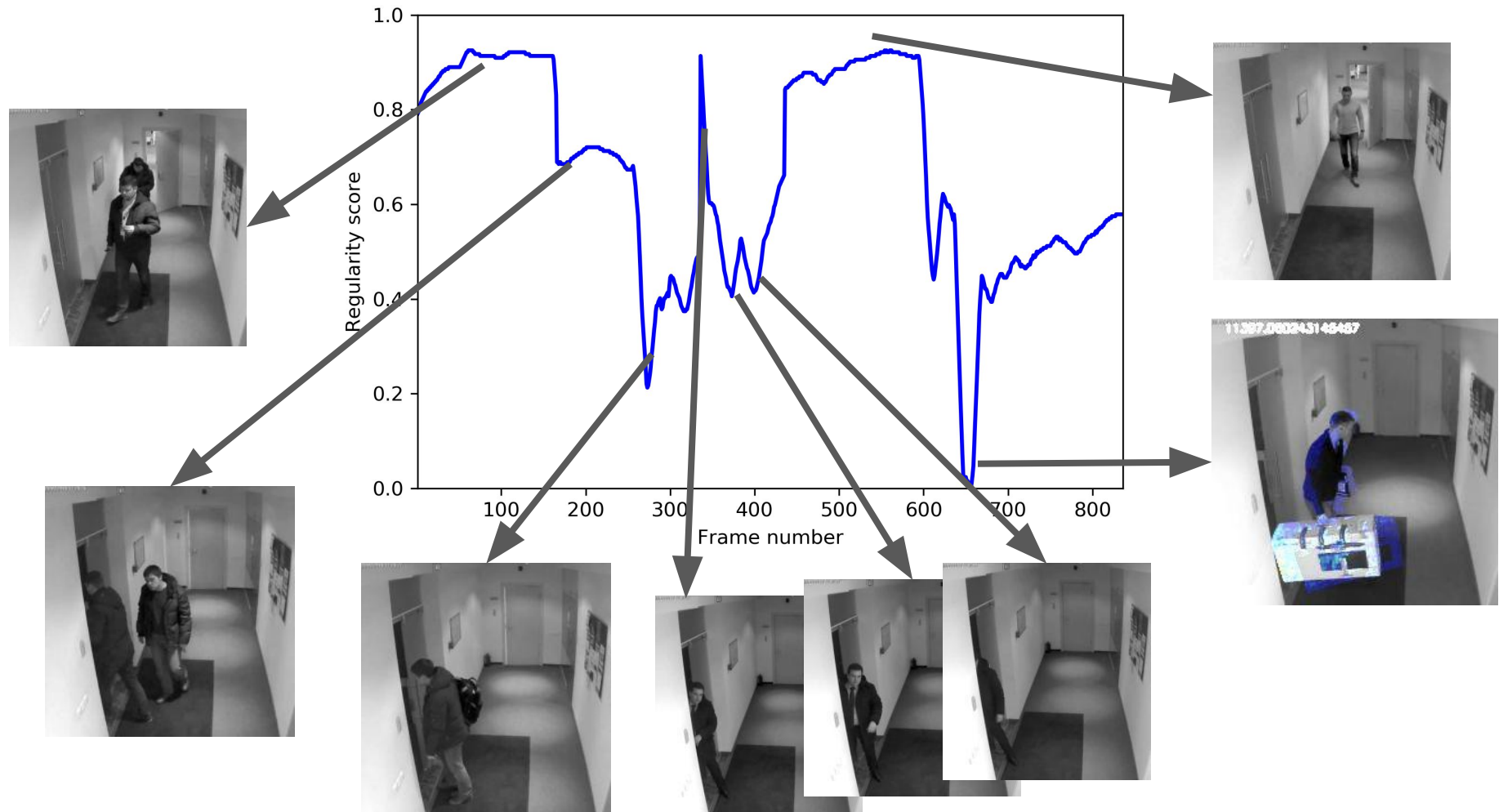
Recall: 97

$$e(t) = \left\| x(t) - f(x(t)) \right\|_2$$

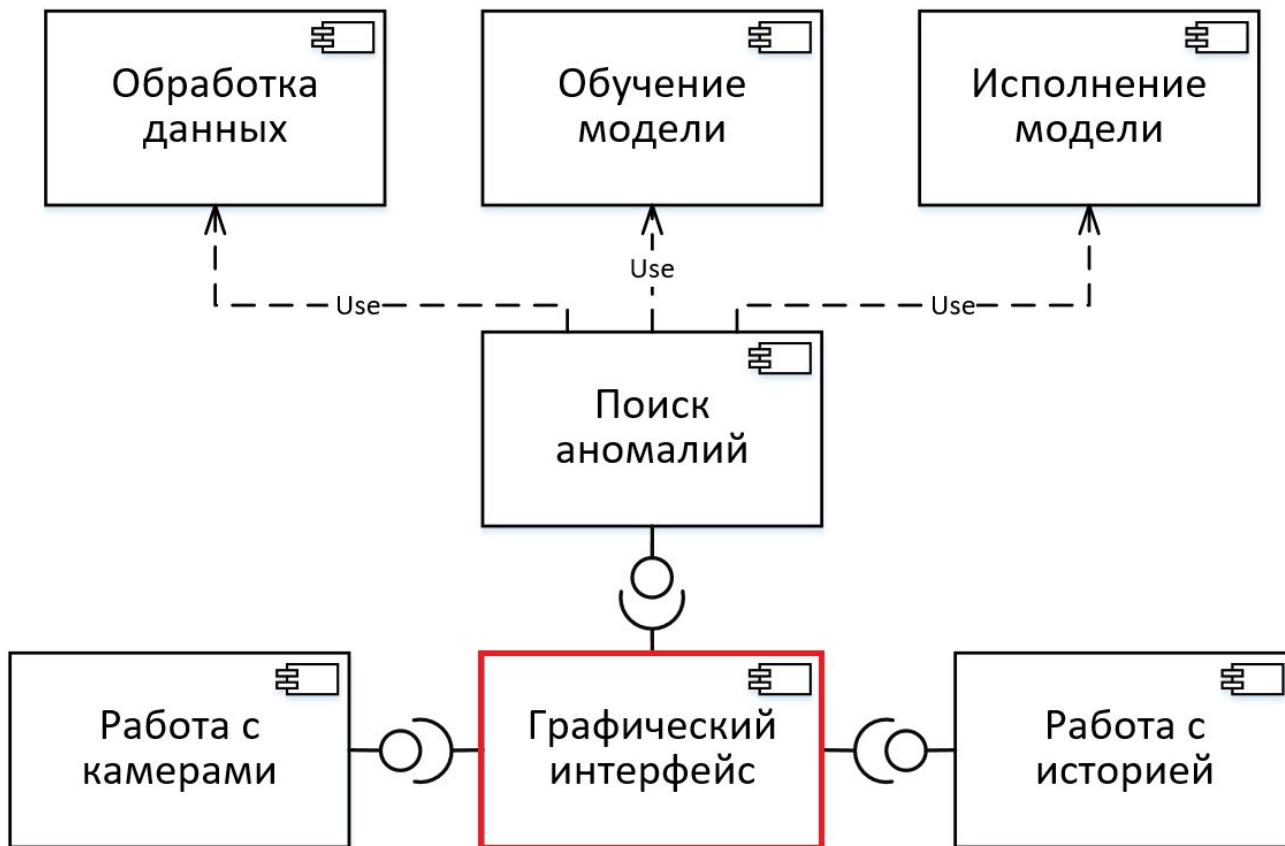
$$s(t) = \frac{e(t) - e(t)_{min}}{e(t)_{max}}$$

$$s_r(t) = 1 - s(t)$$





# Архитектура системы



# Графический интерфейс

- Реализован с помощью библиотеки Qt
- Использует интерфейсы, предоставленные остальными модулями
- Позволяет просматривать в онлайн-режиме видео с камер видеонаблюдения
- В режиме поиска аномалий работает до 3 камер (NVidia GeForce GTX 1060)



# Удобство развертывания. Docker

- Большое количество зависимостей
- Простота развертывания
- Минимальные потери в производительности
- nvidia-docker

# Результаты

- Разработаны требования к системе поиска аномалий при видеонаблюдении.
- Сделан обзор существующих алгоритмов поиска аномалий в видео.
- Разработана архитектура системы, основанная на паттерне MVC
- Реализован прототип системы, включающий следующие компоненты:
  - графический интерфейс
  - компонент для поиска аномалий
  - компонент для работы с историей найденных аномалий
  - компонент для работы с камерами видеонаблюдения
  - компонент для обучения модели на пользовательских данных.
- Проведена апробация алгоритма для офисного помещения.