

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра компьютерных технологий и систем

ЁДА Никита Дмитриевич

ОБНАРУЖЕНИЕ АНОМАЛИЙ ДВИЖЕНИЯ НА
ВИДЕОПОСЛЕДОВАТЕЛЬНОСТИ ПРИ ПОМОЩИ МЕТОДОВ
ОБРАБОТКИ ВРЕМЕННЫХ РЯДОВ

Дипломная работа

Научный руководитель:
старший преподаватель
С. В. Шолтанюк

Допущена к защите

«__» _____ 2025 г.

Зав. кафедрой компьютерных технологий и систем

Доктор педагогических наук, профессор В.В. Казаченок

Минск, 2025

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	6
ГЛАВА 1. ОБЗОР ОСНОВНЫХ МЕТОДОВ И АЛГОРИТМОВ АНАЛИЗА ДВИЖЕНИЯ ОБЪЕКТОВ ВИДЕОПОСЛЕДОВАТЕЛЬНОСТЕЙ	7
1.1 Оптический поток	7
1.2 Трекинг объектов	11
1.3 Сегментация объектов	14
ГЛАВА 2. ИССЛЕДОВАНИЕ АНОМАЛИЙ ДВИЖЕНИЯ ОБЪЕКТОВ ВИДЕОПОСЛЕДОВАТЕЛЬНОСТИ	18
2.1 Определение аномалий. Подходы к выявлению аномалий	18
2.2 Обработка видеопоследовательностей	21
2.3 Методы анализа временных рядов	22
2.4 Классификация аномалий. Применение классификации аномалий	26
ГЛАВА 3. ПОСТАНОВКА ЗАДАЧИ ОБНАРУЖЕНИЯ АНОМАЛИЙ НА ВИДЕОПОСЛЕДОВАТЕЛЬНОСТИ. ПРИМЕНЕНИЕ МЕТОДОВ И АЛГОРИТМОВ АНАЛИЗА ВРЕМЕННЫХ РЯДОВ	28
3.1 Постановка задачи.....	28
3.2 Структура программы. Применение методов и алгоритмов анализа временных рядов для решения поставленной задачи	29
3.3 Визуализация временных рядов и их сегментация	38
3.4 Реализация модели на другой видеопоследовательности	42
ЗАКЛЮЧЕНИЕ	46
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	47

РЕФЕРАТ

Дипломная работа, 47 с., 33 рис., 13 источников.

Ключевые слова: АНОМАЛИИ, АНОМАЛИИ В ДВИЖЕНИИ, ОБНАРУЖЕНИЕ АНОМАЛИЙ, ВРЕМЕННОЙ РЯД, МЕТОДЫ ОБРАБОТКИ ВРЕМЕННЫХ РЯДОВ, КОМПЬЮТЕРНОЕ ЗРЕНИЕ, ВИДЕОПОСЛЕДОВАТЕЛЬНОСТЬ, МЕТОДЫ ОБРАБОТКИ ВИДЕОПОСЛЕДОВАТЕЛЬНОСТЕЙ.

Объект исследования: видеопоследовательности, содержащие движущиеся объекты, и соответствующие временные ряды, полученные на основе их движения.

Цель работы: обнаружение аномалий движения на видеопоследовательности при помощи методов обработки временных рядов.

Методы исследования: методы компьютерного зрения, методы формирования временных рядов, методы статистического и алгоритмического анализа временных рядов, методы визуализации результатов, алгоритм обнаружения аномалий.

Результаты: реализован метод детекции и трекинга объектов, метод анализа видеопоследовательности, метод формирования и анализа временного ряда, метод обработки временного ряда, методы построения графиков аномалий, алгоритм обнаружения аномалий, проведен анализ полученных решений.

РЭФЕРАТ

Дыпломная праца, 47 с., 33 рыс., 13 крыніц.

Ключавыя словы: АНАМАЛІІ, АНАМАЛІІ Ў РУХУ, ВЫЯЎЛЕННЕ АНАМАЛІЙ, ЧАСОВЫ ШЛЯД, МЕТАДЫ АПРАЦОЎКІ ЧАСОВЫХ РЯДАЎ, КАМП'ЮТАРНЫ ЗРОК, ВІДЗАПАСЛЕДНАСЦЬ, МЕТАДЫ ВІДЗАПАСЛЕДНАСЦЯЎ.

Аб'ект даследавання: відзаспаслядоўнасці, якія змяшчаюць аб'екты, якія рухаюцца, і адпаведныя часовыя шэрагі, атрыманыя на аснове іх руху.

Мэта працы: выяўленне анамалій руху на відзаспаслядоўнасці пры дапамозе метадаў апрацоўкі часавых шэрагаў.

Метады даследавання: метады камп'ютарнага зроку, метады фарміравання часовых радоў, метады статыстычнага і алгарытмічнага аналізу часовых радоў, метады візуалізацыі вынікаў, алгарытм выяўлення анамалій.

Вынікі: рэалізаваны метады дэтэкцыі і трэкінгу аб'ектаў, метады аналізу відзаспаслядоўнасці, метады фарміравання і аналізу часовага рада, метады апрацоўкі часовага рада, метады пабудовы графікаў анамалій, алгарытм выяўлення анамалій, праведзены аналіз атрыманых рашэнняў.

ABSTRACT

Thies, 47 pages, 33 figures, 13 sources.

Keywords: ANOMALIES, ANOMALIES IN MOTION, ANOMALY DETECTION, TIME SERIES, TIME SERIES PROCESSING METHODS, COMPUTER VISION, VIDEO SEQUENCE, VIDEO SEQUENCE PROCESSING METHODS.

Object of study: video sequences containing moving objects and corresponding time series obtained based on their movement.

Purpose of work: detection of motion anomalies in video sequences using time series processing methods.

Research methods: computer vision methods, time series generation methods, statistical and algorithmic time series analysis methods, results visualization methods, anomaly detection algorithm.

Results: the method of detection and tracking of objects, the method of video sequence analysis, the method of formation and analysis of time series, the method of time series processing, the methods of constructing anomaly graphs, the algorithm for detecting anomalies were implemented, and the analysis of the obtained solutions was carried out.

ВВЕДЕНИЕ

Обработка и анализ информации представляют собой ключевую задачу для различных научных направлений. Одна из актуальных проблем — это обнаружение аномалий на видеопоследовательности при помощи методов обработки временных рядов. Аномалии могут сигнализировать о нештатной ситуации, нарушении безопасности или необычном поведении объектов.

Методы обработки временных рядов предназначены для анализа динамических процессов. Они позволяют выявлять закономерности, предсказывать поведение системы, обнаруживать отклонения от нормального функционирования. Использование методов открывает новые возможности в области компьютерного зрения и анализа данных.

Актуальность темы определяется ростом интереса к проблемам компьютерного зрения и необходимостью обеспечения безопасности в сферах транспорта, охраны, спорта. В условиях увеличения объема видеоданных становится важным автоматизировать процессы их анализа, что позволяет повышать эффективность систем наблюдения и оперативно реагировать на потенциальные угрозы.

Цель данной дипломной работы заключается в исследовании методов обнаружения аномалий на видеопоследовательностях с использованием подходов обработки временных рядов.

Для достижения данной цели была поставлена задача исследовать современные методы анализа движения объектов, методы выявления аномалий, алгоритмы обработки временных рядов, разработать подход к выявлению аномалий.

Настоящее исследование, направленное на освоение теоретической базы и реализации практических методов анализа временных рядов для выявления аномальных ситуаций в движении объектов.

ГЛАВА 1

ОБЗОР ОСНОВНЫХ МЕТОДОВ И АЛГОРИТМОВ АНАЛИЗА ДВИЖЕНИЯ ОБЪЕКТОВ ВИДЕОПОСЛЕДОВАТЕЛЬНОСТЕЙ

Анализ движения объектов в видеопоследовательностях включает в себя различные методы, каждый из которых имеет свои особенности и области применения. Эти методы могут варьироваться от простых подходов, таких как оптический поток, до более сложных, которые используют машинное обучение и глубокие нейронные сети.

1.1 Оптический поток

Оптический поток — это визуализация перемещения объектов, отображающая смещение отдельных точек между двумя кадрами. Данный поток можно обозначить как поле скоростей, так как сдвиг с точностью до масштаба равен мгновенной скорости. Основная идея метода состоит в том, что для каждой точки (x, y) изображения I_1 находится такой сдвиг (dx, dy) , чтобы исходной точке соответствовала точка с координатами $(x+dx, y+dy)$ на изображении I_2 . Чтобы установить соответствие между точками, используется **инвариантная функция**, то есть необходимо взять функцию точки, которая не изменяется в результате смещения. Считается, что у точки сохраняется яркость или цвет для цветных изображений, но можно считать одинаковыми точки, у которых сохраняется величина градиента или определитель. Сохранение интенсивности может нарушаться при изменении освещения или угла падения света в видеопотоке. Исходя из того, что временной интервал между кадрами мал, и условие освещенности остаются практически неизменными, то это не является критичной проблемой. Поэтому интенсивность часто используют в качестве основного критерия соответствия.

Способы вычисления оптического потока:

- 1) Плотный поток (dense) – рассчитывает смещение для всех точек изображения.
- 2) Выборочный поток (sparse) – определяет сдвиг только для ключевых точек, что значительно ускоряет вычисления.

Ключевые точки — это особые участки изображения, обладающие высокой информативностью и устойчивостью к преобразованиям (например, углы, контрастные области).

Если все точки изображения смещаются одинаково, то применяются более простые методы определения сдвига. Один из таких методов — метод фазовой корреляции.

Фазовая корреляция — это алгоритм оценки трансляционного смещения между двумя схожими изображениями (цифровая корреляция изображения). Основные шаги метода фазовой корреляции:

1) Вычисление двумерного преобразование Фурье для изображений I_1, I_2 :

$$F_1 = \mathcal{F}(I_1), F_2 = \mathcal{F}(I_2), \quad (1.1.1)$$

$$F_1 = \mathcal{F}(I_1) = \iint_{-\infty}^{\infty} I_1(x, y) e^{-i2\pi(ux+vy)} dx dy, \quad (1.1.2)$$

$$F_2 = \mathcal{F}(I_2) = \iint_{-\infty}^{\infty} I_2(x, y) e^{-i2\pi(ux+vy)} dx dy, \quad (1.1.3)$$

где $I_1(x, y)$ - значение пикселя изображения в координатах (x, y) , (u, v) - частотные координаты в пространстве Фурье, i - мнимая единица, $e^{-i2\pi(ux+vy)}$ - комплексная экспонента, представляющая колебания в пространстве частот.

2) Нормализация преобразования Фурье:

$$R = \frac{F_1 \bar{F}_2}{|F_1 \bar{F}_2|}, \quad (1.1.4)$$

$$\bar{F}_2 = |F_2| e^{-j\angle F_2}, \quad (1.1.5)$$

$$\angle F_2 = \tan^{-1} \left(\frac{\text{Im}(F_2)}{\text{Re}(F_2)} \right). \quad (1.1.6)$$

3) Вычисление обратного преобразования Фурье для полученной функции:

$$r(x, y) = \mathcal{F}^{-1}(R) = \iint_{-\infty}^{\infty} R(u, v) e^{i2\pi(ux+vy)} dx dy. \quad (1.1.7)$$

Метод Лукаса-Канаде. Рассмотрим интенсивность пикселя $I_1 = (x, y, t_1)$ на первом изображении. На втором изображении точка смещается на (dx, dy) за время dt . Разложив функцию интенсивности в ряд Тейлора с точностью до первого порядка, получим:

$$I_2 = I(x + dx, y + dy, t_1 + dt) = I_1 + I_x dx + I_y dy + I_t dt, \quad (1.1.8)$$

где I_x, I_y, I_t — частные производные по пространственным координатам и времени, отражающие изменения яркости в точке (x, y) между кадрами.

Условие постоянства интенсивности предполагает, что $I_1 = I_2$. Отсюда следует, что $I_x dx + I_y dy + I_t dt = 0$. Исходя из этого, имеем уравнение с двумя неизвестными (dx, dy) . Рассмотрим окно вокруг точки и минимизируем суммарную погрешность с весовыми коэффициентами, распределенными по Гауссу, чтобы пиксели, которые были ближе всего к исследуемому, имели наибольший вес. Тогда, после всех преобразований, получим систему следующего вида:

$$\begin{aligned} \sum_{i,j} g(i,j) I_x(x_i, y_j)^2 dx + \sum_{i,j} g(i,j) I_x(x_i, y_i) I_y(x_i, y_j) dy = \\ = - \sum_{i,j} g(i,j) I_x(x_i, y_j) I_t(x_i, y_j) dt, \end{aligned} \quad (1.1.9)$$

$$\begin{aligned} \sum_{i,j} g(i,j) I_x(x_i, y_j) I_y(x_i, y_j) dx + \sum_{i,j} g(i,j) I_y(x_i, y_j)^2 dy = \\ = - \sum_{i,j} g(i,j) I_y(x_i, y_j) I_t(x_i, y_j) dt. \end{aligned} \quad (1.1.10)$$

Данная система имеет единственное решение, если ее определитель отличен от нуля. В противном случае, когда определитель не равен нулю, решений либо нет, либо их бесконечно много [5].

Метод Хорн-Шунк (Horn-Schunck). Метод опирается на предположение о том, что на всем изображении оптический поток будет достаточно гладким (не содержит резких изменений). От уравнения $I_x dx + I_y dy + I_t dt = 0$ перейдем к функционалу $(I_x dx + I_y dy + I_t dt = 0)^2 + \alpha^2(|\partial u|^2 + |\partial v|^2)$, то есть добавляем требование на отсутствие резкого изменения сдвигов с весовым коэффициентом α . Минимизация функционала приводит к системе из двух уравнений:

$$I_x^2 u + I_x I_y v = \alpha^2 \Delta u - I_x I_t, \quad (1.1.11)$$

$$I_x I_y u + I_y^2 v = \alpha^2 \Delta u - I_y I_t. \quad (1.1.12)$$

В этих уравнениях считаем разницу со средним значением $\Delta u = \langle u \rangle - u$, где u – значение пикселя, $\langle u \rangle$ – среднее значение всех пикселей в заданной области. Получаем систему уравнений, которую записываем для каждого пикселя и решаем общую систему итеративно:

$$u_{n+1} = u_n - I_x(I_x u_n + I_y v_n + I_t)/(I_x^2 + I_y^2 + \alpha^2), \quad (1.1.13)$$

$$v_{n+1} = v_n - I_y(I_x u_n + I_y v_n + I_t)/(I_x^2 + I_y^2 + \alpha^2). \quad (1.1.14)$$

Метод Фарнбека (Farneback). Фарнбек предположил алгоритм оценки оптического потока, основанный на аппроксимации локальной интенсивности изображения квадратичной формой: $I = xAx + bx + c$ с симметричной матрицей A . В случае, когда изображение перемещается в пределах окрестности, то $I_2(x) = I_1(x - d)$, подставляем квадратичное разложение и раскрываем скобки, тогда получаем следующие выражения:

$$A_2 = A_1, \quad (1.1.15)$$

$$b_2 = b_1 - 2A_1 d, \quad (1.1.16)$$

$$c_2 = d^T A_1 d = b_1^T d + c_1. \quad (1.1.17)$$

После вычисления A , b и c для обоих изображений система уравнений становится переопределенной относительно вектора смещения d . Для нахождения оптимального решения выразим d следующим образом:

$$d = -\frac{1}{2}A^{-1}(b_2 - b_1), \quad (1.1.18)$$

$$A = \frac{1}{2}(A_1 + A_2), \quad (1.1.19)$$

$$\Delta b = -\frac{1}{2}(d_2 - d_1), \quad (1.1.20)$$

$$Ad = \Delta b. \quad (1.1.21)$$

Предположим, что все точки имеют одинаковый сдвиг, тогда проинтегрируем погрешность $|Ad = \Delta b|^2$ по окну с гауссовскими весовыми коэффициентами w , затем, учитывая прошлые значения, найдем такой вектор d , который минимизирует эту суммарную погрешность. Получим оптимальное значение $d = (\Sigma w A^T A)^{-1} \Sigma w A^T \Delta b$ и соответствующую минимальную ошибку $e = (\Sigma w b^T \Delta b) - d^T \Sigma w A^T \Delta b$. Для каждой точки сначала считаем произведения $A^T A$, $A^T \Delta b$, $\Delta b^T \Delta b$. После данные значения усредняем по окну с гауссовыми весами. Затем обращаем усредненную матрицу $A^T A$ и умножаем ее на усредненный вектор $A^T \Delta b$, чтобы получить искомый вектор смещения d . Ошибку считаем через скалярное произведение усредненных величин. Все операции можно выполнять сразу для всего изображения, что ускоряет расчеты. Матричные

произведения считаются в каждой точке независимо, а усреднение делается общим фильтром, что позволяет эффективно обрабатывать большие изображения.

Метод простого потока (SimpleFlow). Для каждой точки в рассматриваемой области вычисляется «энергетическая функция», которая логарифмически обратно пропорциональна вероятности перехода данной точки в соответствующую точку на следующем изображении. Формула функции энергии представлена следующим образом:

$$e(x, y, dx, dy) = |F_1(x, y) - F_2(x + dx, y + dy)|^2. \quad (1.1.22)$$

Далее считается свертка с гауссовым окном:

$$E(x_0, y_0, dx, dy) = \quad (1.1.23)$$

$$= \sum (e(x, y, dx, dy) \exp\left(-\frac{|(x, y) - (x_0, y_0)|^2}{2s_d}\right) \exp\left(-\frac{|F_1(x, y) - F_1(x_0, y_0)|^2}{2s_c}\right)).$$

и находятся значения (dx, dy) , минимизирующие эту функцию. Чтобы получить субпиксельную точность, рассматривается малая окрестность найденной оптимальной точки (dx, dy) и в ней ищется пик «функции энергии» как пик параболоида.

Математическое описание методов и алгоритмов анализа движения объектов видеопоследовательности взято из статьи [5].

1.2 Трекинг объектов

Трекинг представляет собой процесс отслеживания координат движущегося объекта в видеопотоке. Современные алгоритмы трекинга анализируют последовательность кадров, определяя положения объекта в каждом временном отрезке [1].

Ключевая сложность трекинга заключается в необходимости точного позиционирования объекта при его быстром перемещении относительно частоты кадров видео. Решение задачи трекинга объектов в видеопоследовательности основано на использовании математических моделей движения, которые учитывают различные внешние воздействия. Архитектура систем трекинга объектов

включает две компоненты: модуль локализации объектов и блок фильтрации данных.

Для локализации объектов применяются следующие подходы:

- 1) Детекторы объектов на основе машинного обучения.
- 2) Методы семантической сегментации.
- 3) Алгоритмы выделения ключевых признаков.
- 4) Глубокие нейронные сети.

В процессе обработки данных возникают проблемы, связанные с точностью. Они решаются посредством алгоритмов фильтрации, которые объединяют априорную информацию о модели движения с фактическими измерениями. Среди методов фильтрации наиболее распространенный — **фильтр Калмана**. Фильтр применяется для оценки систем по зашумленным данным.

Состояние системы описывается вектором параметром. Благодаря некоторым воздействиям система переходит из одного состояния в другое. В свою очередь, множество всех состояний системы и переходов образует некоторую модель. Математическая модель системы описывается следующим образом:

- 1) Вектором состояния (совокупность параметров системы).
- 2) Оператором перехода между состояниями.
- 3) Вектором наблюдений (измеряемыми параметрами).

Особенностью фильтра Калмана является способность оценивать полное состояние системы даже при неполной наблюдаемости параметров. Метод основан на представлении системы в виде дискретной линейной цепи Маркова, где переходы между состояниями описываются линейными операторами с добавлениями гауссовского шума и, при наличии, управляющих воздействий.

Математическая модель данного процесса:

$$x_k = F_k x_{k-1} + B_k u_k + w_k, \quad (1.2.1)$$

$$z_k = H_k x_k + v_k. \quad (1.2.2)$$

где F_k – матрица перехода состояния (размерностью $n \times n$), определяющая эволюцию системы между моментами $k-1$ и k без внешнего управления, B_k – матрица управления (размерностью $n \times l$), связывающая управляющий сигнал u_k (размерности l) с изменениями состояния системы, H_k – матрица наблюдений (размерностью $s \times n$), преобразующая внутреннее состояние x_k в измеряемый вектор z_k (размерности s), w_k, v_k – произвольные значения, представляющие нормально распределенный шум при измерении состояния с соответствующими матрицами ковариации Q_k, R_k , $w_k \sim N(0, Q_k)$, $v_k \sim N(0, R_k)$.

Фильтр Калмана работает циклически, постоянно переключаясь между двумя основными этапами. Сначала идет фаза предсказания, где система пытается предсказать новое значение, опираясь на внутреннюю математическую модель и предыдущие данные. В этот момент рассчитываются так называемые **априорные оценки** – неточные приближения, которые не учитывают новые измерения. Следующий этапом является фаза коррекции, где предсказания уточняются в зависимости от реальных наблюдений, но с учетом возможной погрешности измерений. В результате получается **апостериорные оценки** — более точные, учитывающие модель и фактические данные. В некоторых случаях фаза коррекции может быть пропущена и предсказание будет происходить на основе неуточненной оценки.

Пусть x_k - действительное состояние системы на момент времени k , $\hat{x}_k$ - оцененное состояние на момент времени k , $\hat{x}_k^-$ - предсказанное состояние системы на момент времени k , P_k - оцененная матрица ковариации ошибок измерения состояния, P_k^- - предсказанная матрица ковариации ошибок измерения состояния. Схематично алгоритм фильтра Калмана представлен на рисунке 1.1.

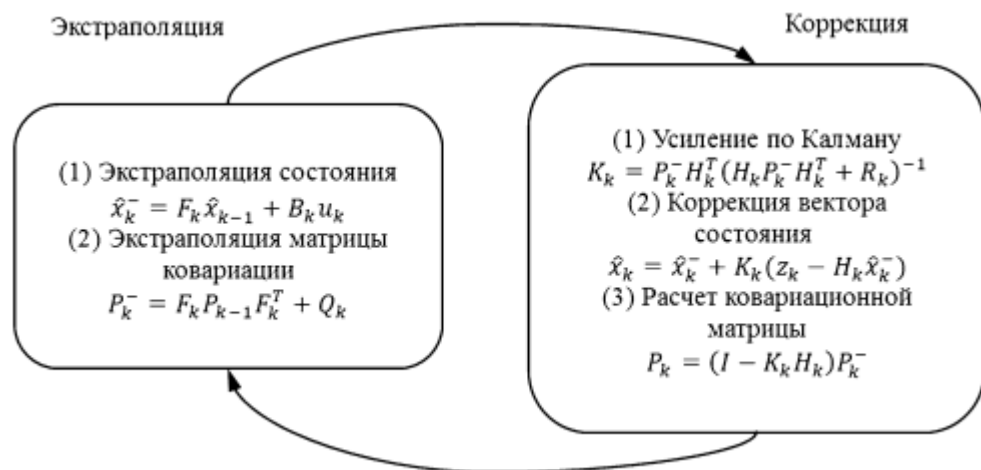


Рисунок 1.1 - Алгоритм фильтра Калмана

Алгоритм фильтра Калмана приведен в статье [1].

Фильтр Калмана эффективно решает проблему, когда объект временно скрыт, так как фильтр предсказывает будущее положение объекта. Это достигается благодаря экстраполяции текущего состояния через математическую модель движения. Алгоритм значительно снижает влияние шумов и ошибок детектирования, которые возникают при работе с видеоданными.

Метод сдвига средних (Mean Shift) — это метод нечеткой кластеризации, который находит плотностные области в данных. Он работает путем перемещения центроидов к областям высокой плотности данных. Основные шаги:

1) Выбираются начальные позиции центроидов. Это могут быть случайные точки из выборки.

2) Для центроида определяется окно (в форме круга), в пределах которого будет происходить вычисление.

3) Вычисляется среднее значение всех точек, которые находятся в окне. Вычисление происходит включая каждый центроид.

$$m(x) = \frac{\sum_{x_i \in N(x)} K(|x - x_i|) x_i}{\sum_{x_i \in N(x)} K(|x - x_i|)}, \quad (1.2.3)$$

где $m(x)$ – новое положение центроида, $N(x)$ – набор точек в окне, K – функция ядра.

4) Перемещаем центроид в новое положение $m(x)$.

5) Повторяем шаги 2-4 до тех пор, пока центроиды не перестанут изменять свои позиции.

6) После сходимости центроидов объединяются точки, взаимное расположение которых незначительно относительно друг друга.

Функция ядра:

$$K(u) = \frac{1}{(2\pi)^{d/2}} e^{-\frac{1}{2}u^2}. \quad (1.2.4)$$

Обновление центроида:

$$x_{new} = x_{old} + \frac{1}{N} \sum_{i=1}^N K(|x_{old} - x_i|)(x_i - x_{old}). \quad (1.2.5)$$

1.3 Сегментация объектов

Сегментация изображений — это разбиение цифрового снимка на области для последующего анализа. Данный подход позволяет выделить и анализировать отдельные объекты на снимке. Наиболее распространенный алгоритм для сегментации — **алгоритм К-средних (K-means)**, который относится к алгоритмам кластеризации.

Алгоритм K-means, описанный в работе [7], реализуется следующим образом. Сначала случайным образом выбираются K точек, которые становятся центрами будущих кластеров. Затем для каждого пикселя изображения вычисляется расстояние до всех центров кластеров, после чего пиксель присваивается к ближайшему кластеру. На следующем этапе происходит перерасчет центроидов — новые центры кластеров устанавливаются в средние точки соответствующих

групп пикселей. Процесс повторяется циклически до тех пор, пока центры кластеров не перестанут значительно смещаться при последующих итерациях, что свидетельствует о стабилизации решения.

Особенность алгоритма заключается в том, что на каждом шаге уточняется как принадлежность пикселей кластерам, так и положение самих центров. В результате изображение делится на K однородных областей, где пиксели внутри каждого кластера схожи по характеристикам. Данный подход эффективен для предварительной обработки изображений.

Обновление центров кластеров задается формулой:

$$\mu_j = \frac{1}{N_j} \sum_{i \in C_j} x_i, \quad (1.2.6)$$

где μ_j - центр j -го кластера, N_j - количество точек в кластере j , C_j - набор точек в кластере j , x_i - точки данных.

Кластеризация изображений алгоритмом К-средних работает с цветом, координатами и яркостью пикселей, объединяя их в группы. Чтобы сравнить разные характеристики, данные сначала приводятся к общему масштабу, а затем считают расстояние между пикселем и центром кластера как сумму квадратов разниц по всем параметрам. Результат сильно зависит от начального выбора центров и их количества – можно получить как неестественное разделение объекта на части, так и слипание разных объектов. Несмотря на возможные отклонения, данный алгоритм прост в реализации и в скорости обработки, что делает его популярным для базовой сегментации.

В случаях, когда необходимо найти границы объектов, используется — **детектор Кэнни**. Изображение переводится в черно-белый формат. Это можно осуществить путем использования различных фильтров, таких как YUV, HSL, HSV. Затем размывают гауссовским фильтром с маской 5x5 пикселей, чтобы убрать шумы. После, алгоритм Кэнни ищет резкие перепады яркости, определяя, где проходят настоящие границы, а где помехи. Данный алгоритм требует больше вычислений по сравнению с К-средним, так как позволяет находить даже слабые границы. Таким образом, алгоритм К-теап лучше выделяет цветовые зоны, а Кэнни точнее находит контуры.

Маска Гауссианы представлена на рисунке 1.2.

	2	4	5	4	2
	4	9	12	9	4
1	5	12	15	12	5
115	4	9	12	9	4
$\sigma=1.4$	2	4	5	4	2

Рисунок 1.2 - Маска гауссианы

После сглаживания изображения гауссовым фильтром начинается поиск границ с помощью вычисления градиентов. Для этого применяют оператор Собеля — специальные сверточные ядра 3x3, которые выделяют перепады яркости по горизонтали и вертикали.

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Рисунок 1.3 - Фильтры Собеля

Фильтры помогают вычислить для каждого пикселя угол наклона градиента. При этом углы градиента квантуются до четырех основных направлений: 0°, 45°, 90°, 135°.

Затем определяются граничные пиксели — те, у которых значение градиента достигает локального максимума в направлении его действия, то есть пиксель должен иметь более сильный градиент по сравнению с соседними пикселями вдоль линии границы.

Основным недостатком данного подхода является высокая чувствительность к шумам на изображении. Поэтому крайне важна качественная предварительная обработка исходных данных. Математические основы данного алгоритма описаны в исследовании [7].

Нейронные сети. Лучших результатов по сегментации изображения достигают при помощи нейронных сетей. Для задач сегментации лучше всего подходят полносверточные нейронные сети

С развитием технологий глубокого обучения наибольшую популярность приобрели модели, такие как **You Only Look Once (YOLO)**. Этот алгоритм позволяет одновременно детектировать и классифицировать объекты в реальном времени, что делает его наиболее оптимальным для применения в условиях потоковой видеопередачи. Функция потерь для YOLO задается следующим образом:

$$L = \sum_{i=0}^{S^2} \sum_{j=0}^B loss_{coord} + \sum_{i=0}^{S^2} loss_{class}, \quad (1.2.7)$$

где S - размер сетки, B - количество боксов, $loss_{coord}$ - ошибка координат, $loss_{class}$ - ошибка классификации.

ГЛАВА 2

ИССЛЕДОВАНИЕ АНОМАЛИЙ ДВИЖЕНИЯ ОБЪЕКТОВ ВИДЕОПОСЛЕДОВАТЕЛЬНОСТИ

Исследование аномалий движения объектов на видеопоследовательностях является одной из целевых задач компьютерного зрения и обработки видеоданных. Аномалии, как правило, определяются как события, отклоняющиеся от ожидаемого или нормального поведения в видеопоследовательности.

2.1 Определение аномалий. Подходы к выявлению аномалий

Аномалия в контексте видеопоследовательностей — это отклонение от нормального шаблона движения, которое может проявляться в:

- 1) Неправильном перемещении объектов (например, человек, внезапно бегущий в толпе).
- 2) Нарушении привычного потока движения (машина, едущая по тротуару).
- 3) Появлении необычных объектов или исчезновении привычных объектов.

Определение аномалии сильно зависит от контекста: то, что считается нормой в одном случае, может быть аномалией в другом.

Существующие подходы к выявлению аномалий можно разделить на три основные группы:

- 1) Статистические методы.
- 2) Методы машинного обучения.
- 3) Методы глубокого обучения.

Статистические методы. Эти методы основаны на анализе данных через распределения и вероятности. В таком случае предполагается, что нормальное поведение может быть описано статистической моделью, а аномалии представляют собой отклонения от заданной модели.

Методы порогового анализа. Задаются фиксированные пороги для определенных характеристик (например, скорости, размера объекта или направления движения). В случае, если значение превышает заданный порог, то такое событие считается аномалией. Например, если x - скорость объекта, то:

$$anomaly \Leftrightarrow x > T, \tag{2.1.1}$$

где T - пороговое значение.

Параметрические модели. Гауссовы модели, которые описывают данные как нормальное распределение. Аномалиями считаются точки, лежащие за пределами определенного доверительного интервала. Если $X \sim N(\mu, \sigma^2)$, то аномалии могут быть определены следующим образом:

$$anomaly \Leftrightarrow |X - \mu| > z \cdot \sigma, \quad (2.1.2)$$

где z - значение из стандартного нормального распределения.

Непараметрические модели. Гистограммы или ядерная оценка плотности, используются для описания распределения данных без предположений о форме распределения. Ядерная оценка плотности может быть представлена так:

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n K_h(x - x_i), \quad (2.1.3)$$

где K_h - ядерная функция, h - ширина окна, а x_i - наблюдаемые данные.

Методы машинного обучения. Алгоритм машинного обучения предоставляет гибкий инструментарий, позволяющий реализовать выявление аномалий при наличии большой базы данных, полученных в процессе обучения модели.

Обучение с учителем. Предполагает наличие размеченного набора данных, где нормальные и аномальные события уже выделены. Модели классификации, такие как деревья решений, Support Vector Machine (SVM) или ансамблевые методы (например, Random Forest), могут быть использованы для обучения классификатора. Для SVM задача формулируется как:

$$\text{minimize } \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^n \xi_i, \quad (2.1.4)$$

где ω - вектор весов (опорных векторов), C - параметр регуляризации, ξ_i - переменные, представляющие собой штрафы за ошибки в классификации.

При ограничениях:

$$y_i(\omega^T x_i + b) \geq 1 - \xi_i, \quad (2.1.5)$$

где y_i - метка класса, x_i - вектор признаков для i -го образца, b - смещение, которое добавляется к линейной комбинации весов и входных данных, ω^T - транспонированный вектор весов ω .

Обучение без учителя. Применяется, когда размеченные данные отсутствуют. Методы кластеризации, такие как K-means Clustering, группируют данные:

$$\text{minimize } \sum_{i=1}^k \sum_{x_j \in C_i} \|x_j - \mu_i\|^2, \quad (2.1.6)$$

где μ_i - центр кластера C_i , k - число кластеров, x_j - наблюдение в данных, C_i - набор объектов, принадлежащих i -му кластеру.

Методы глубокого обучения. Современные подходы, основанные на глубоких нейронных сетях, позволяют эффективно выявлять аномалии в сложных видеопоследовательностях. Метод является частным случаем основных алгоритмов машинного обучения, в основе которого существует множество скрытых слоев, извлекающих признаки из входных данных и передающих их на последующий слой.

Автоэнкодеры (Autoencoder) используются для сжатия и восстановления данных. Если объект или событие плохо восстанавливается, это может указывать на аномалию. Ошибка восстановления может быть определена как:

$$Loss = \|x - \hat{x}\|^2, \quad (2.1.7)$$

где $\hat{x}$ - восстановленное значение, x - текущее значение.

Генеративные состязательные сети (Generative Adversarial Networks) обучаются генерировать нормальные данные. Отклонения от модели выявляются как аномалии. Функция потерь может быть представлена как:

$$Loss = E[\log(D(x))] + E[\log(1 - D(G(z)))], \quad (2.1.8)$$

где $D(x)$ - функция дискриминатора, которая оценивает, насколько вероятно, что данные x принадлежат реальному распределению, $G(x)$ - генератор, который принимает случайный шум z и генерирует данные, пытаясь сделать их похожими на реальные, $D(G(z))$ - оценка дискриминатора для данных, сгенерированных генератором, E - математическое ожидание.

Сверточные и рекуррентные нейронные сети (Convolutional Neural Network) используются для анализа пространственных особенностей, таких как формы или текстуры, а Recurrent Neural Network или Long Short-Term Memory для временных последовательностей. Для Long Short-Term Memory можно использовать следующие уравнения:

Формула для забывания:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \quad (2.1.9)$$

где σ - сигмоидная функция, W_f - матрица весов для забывания, b_f - смещение для забывания, $[h_{t-1}, x_t]$ - конкатенация предыдущего состояния h_{t-1} и входных данных x_t .

Формула для состояния ячейки:

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c), \quad (2.1.10)$$

Обновление состояния ячейки:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t, \quad (2.1.11)$$

Формула для входного состояния:

$$h_t = O_t \cdot \tanh(C_t), \quad (2.1.12)$$

где O_t – выходные веса, h_t – окончательное состояние.

2.2 Обработка видеопоследовательностей

Анализ временных рядов позволяет понять поведение временных данных и выявить скрытые тенденции и закономерности, которые в дальнейшем используются для прогнозирования.

Устранение шума. Видеоданные могут содержать разнообразные виды помех, такие как цифровые артефакты, шумы камеры или внешние факторы (например, дождь, снег, пыль). Для их устранения используются фильтры сглаживания, например, медианный или гауссовский фильтр:

$$I'(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k \omega(i, j) * I(x + i, y + j), \quad (2.2.1)$$

где $I(x, y)$ - исходное изображение, $I'(x, y)$ - отфильтрованное изображение, $\omega(i, j)$ - весовая функция фильтра, k - радиус окна фильтра.

Стабилизация видео. Для устранения колебаний камеры применяется компенсация движения, которая основана на определении матрицы преобразования между кадрами:

$$T = \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.2.2)$$

где T - матрица аффинного преобразования, (t_x, t_y) - сдвиг по осям, (a, b, c, d) - параметры масштабирования и вращения.

Преобразование данных. Для упрощения анализа видеоданные разбиваются на последовательности кадров. Пусть $F = \{f_1, f_2, \dots, f_n\}$ последовательность

кадров, где f_i - i -й кадр. При выборе ключевых кадров часто используется метод на основе разности кадров:

$$D(f_i, f_{i-1}) = \sum_{x,y} |f_i(x, y) - f_{i-1}(x, y)|, \quad (2.2.3)$$

Выделение ключевых характеристик:

- 1) Использование оптического потока позволяет определить направление и скорость движения объектов в сцене.
- 2) Сегментация изображения позволяет выделить движущиеся объекты на фоне динамических элементов.
- 3) Необходимость выделения формы и размера объектов для дальнейшей идентификации по окончании процесса сегментации.
- 4) Осуществление трекинга (отслеживания) динамических объектов в кадрах.
- 5) Извлечение временных характеристик временного ряда (ускорение или замедление).
- 6) Использование алгоритмов машинного обучения для получения классификаций на основе характеристик (форма, размер, скорость).

2.3 Методы анализа временных рядов

Изучение временных рядов дает возможность выявить закономерности и обнаружить скрытые динамические процессы. Среди основных подходов к анализу можно выделить три фундаментальных метода:

- 1) Проверка стационарности.
- 2) Исследование автокорреляции.
- 3) Спектральный анализ.

Стационарность — свойство временного ряда, означающая постоянство его статистических параметров (среднего значения и дисперсии) на всем временном интервале. Ряды, которые обладают данным свойством, значительно проще анализировать и прогнозировать. **Нестационарные временные ряды** — это такие ряды, которым характерно динамическое изменение: наличие выраженного тренда, периодических колебаний или повторяющихся сезонных паттернов. Повторение определенных событий в течении года называется **сезонностью**.

Для диагностики стационарности применяются **графические методы**:

- 1) **Визуализация временного ряда** предназначена для оценки динамического изменения.

2) **График автокорреляции** отражает взаимосвязь между наблюдениями на разных временных лагах.

3) **График частной автокорреляции** учитывает прямую зависимость между двумя точками, исключая влияние промежуточных значений.

Автокорреляция — это статистическая взаимосвязь между значениями ряда, разделенными определенным временным интервалом, рисунок 2.1. Высокий уровень автокорреляции указывает на устойчивую зависимость между данными в разные моменты времени. Для ее расчета используется **коэффициент корреляции Пирсона**, который измеряет силу линейной связи между переменными. Если автокорреляция статистически значима, это свидетельствует о наличии тренда, цикличности или сезонных колебаний в данных.

Формула для расчета коэффициента Пирсона:

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n \sum x^2 - (\sum x)^2][n \sum y^2 - (\sum y)^2]}} \quad (2.3.1)$$

где n – количество пар наблюдений, x , y – значения переменных.

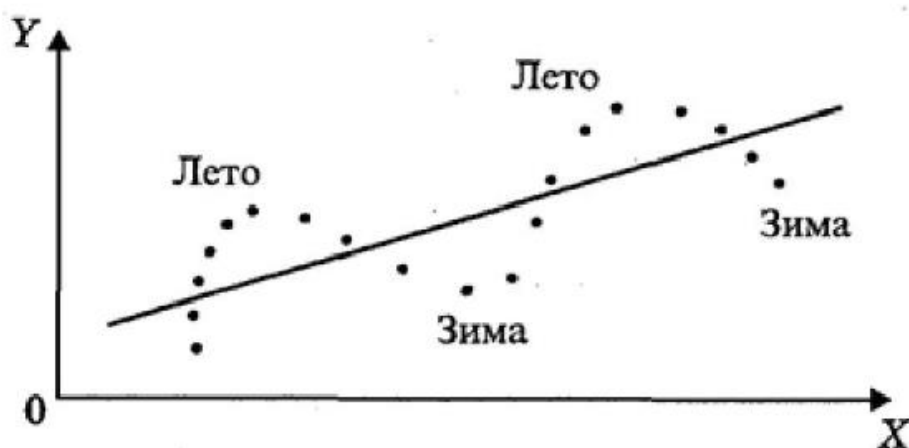


Рисунок 2.1 – Пример автокорреляции

Спектральный анализ — это подход, направленный на изучение частотных характеристик временного ряда. Анализ предназначен для обнаружения скрытых периодических закономерностей, которые трудно выявить стандартными методами. Суть метода заключается в декомпозиции временного ряда на отдельные частотные компоненты с последующим изучением их спектральной плотности.

Авторегрессионная модель (AR-модель) — это статистическая модель, прогнозирующая будущие значения ряда на основе его предыдущих наблюдений. Основной принцип AR-модели заключается в том, что текущее значение

линейно зависит от одного или нескольких прошлых значений. Наглядный пример работы авторегрессионной модели представлен на рисунке 2.2.

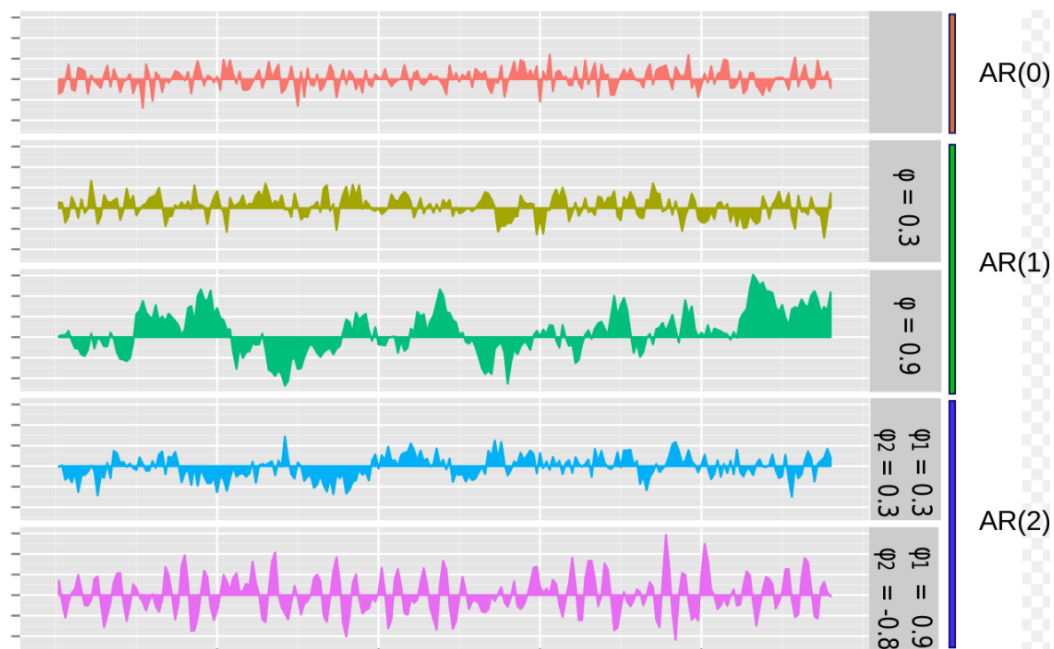


Рисунок 2.2 - Пример AR-модели

Модель скользящего среднего (МА-модель) основывается на использовании прошлых ошибок прогнозирования (разница между фактическими и предсказанными значениями) для построения будущих прогнозов. Самая простая форма модели – MA(1), где текущая ошибка зависит исключительно от ошибки предыдущего периода.

Модель ARIMA (Autoregressive Integrated Moving Average) представляет собой комбинированный подход, объединяющий принципы авторегрессии (AR) и скользящего среднего (MA). Ключевое преимущество модели заключается в возможности работы с нестационарными данными, что отличает ее от простых AR и MA моделей. Модель характеризуется тремя параметрами: p – порядок авторегрессионной составляющей, q – порядок компоненты скользящего среднего, d – степень дифференцирования, необходимая для достижения стационарности.

Вид модели ARIMA:

$$\Delta^d Y_t = \sum_{i=1}^p \alpha_i \Delta^d Y_{t-i} + \sum_{j=1}^q \beta_j \varepsilon_{t-j} + \varepsilon_t. \quad (2.3.2)$$

Формула для вычисления порядка авторегрессии:

$$Y_t = \sum_{i=1}^p \alpha_i Y_{t-i} + \varepsilon_t, \quad (2.3.3)$$

где $\alpha_1, \alpha_2, \dots, \alpha_p$ – коэффициенты авторегрессии, ε_t – случайная ошибка.

Формула для скользящего среднего:

$$Y_t = \sum_{j=1}^q \beta_j \varepsilon_{t-j}, \quad (2.3.4)$$

где $\beta_1, \beta_2, \dots, \beta_q$ - параметры модели.

Математическое описание модели ARIMA взято из статьи [8].

Модель SARIMA (Seasonal ARIMA) представляет собой расширенную версию классической ARIMA-модели. Модель SARIMA была разработана для работы с временными рядами, имеющими выраженные сезонные колебания. Данный подход включает в себя все возможности стандартной ARIMA-модели, дополненные специальными сезонными компонентами. На рисунке 2.3 представлена визуализация работы SARIMA-модели.

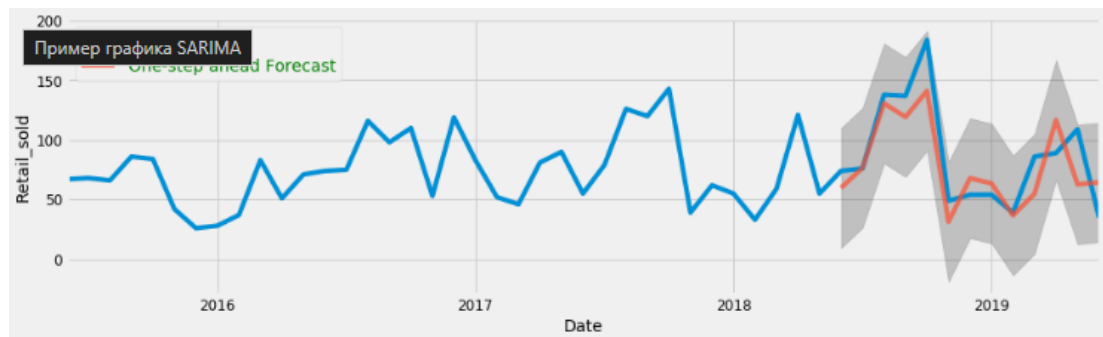


Рисунок 2.3 - Пример графика SARIMA

Описание модели SARIMA взято из статьи [6].

Экстраполяционные методы представляют собой класс подходов к прогнозированию, основанных на продлении в будущее наблюдаемых закономерностей прошлых периодов. Данные методы особенно эффективны для краткосрочного прогнозирования. К этим методам относятся: скользящее среднее, экспоненциальное сглаживание, метод Хольта-Винтерса. Метод Хольта-Винтерса добавляет тренд и сезонность к модели экспоненциального сглаживания.

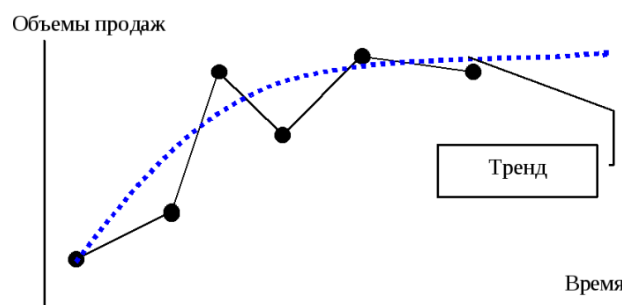


Рисунок 2.4 - Пример графика экстраполяции

Нейронные модели работают по принципу самообучения на исторических данных с последующей генерацией прогноза для новых наблюдений. **Регрессионные алгоритмы** машинного обучения устанавливают взаимосвязь между предыдущими значениями ряда и будущими значениями. По сравнению с экстраполяционными подходами, нейросети демонстрируют более высокую прогнозную точность.

2.4 Классификация аномалий. Применение классификации аномалий

Были рассмотрены следующие **классификации аномалий**: люди, транспортные средства, животные и природные явления. Каждый из этих классов является уникальным и требует описания.

1) Аномалии в поведении людей часто выражаются в нарушении привычных нам паттернов движения: резкое изменение траектории движения, ускорение, остановка, движение в обратном направлении относительно основного потока.

2) Аномалии в движении транспортных средств проявляются как отклонение от привычных маршрутов, нарушение скоростного режима, движение по встречной полосе, резкие торможения, движение в запрещенных зонах, выезд на тротуар, создание пробок.

3) Движение животных могут создавать аномальные ситуации, требующие внимания. К необычным повадкам животных можно отнести появление в нехарактерных местах.

4) Природные явления: оползень, лавина, торнадо.

Классификация по уровням анализа:

1) Уровень траектории анализирует отклонение в маршрутах движения объектов. Оно включает в себя неожиданное пересечение траекторий или резкое изменение направления.

2) Потокное движение анализирует общую динамику, например, нарушение плотности, скорости, однородности движения толпы, транспортных средств или групп других объектов.

3) Индивидуальные объекты. Внимание сосредоточено на характеристиках объекта: изменение формы, размера, характера движения. Такой ряд признаков может оценивать нормальность поведения.

Применение классификаций:

1) Классификации применяются в сфере безопасности и наблюдении. Анализ позволяет выявлять факты проявления подозрительного поведения, агрессии, нарушение правил дорожного движения.

2) Транспорт и логистика. Методы используются для анализа траекторий беспилотных автомобилей и дальнейшего предотвращения их аварий, для контроля нестандартной перевозки груза.

3) Медицина. Задачи, связанные с выявлением нарушений движения у пациентов, например, при расстройствах походки, внезапных падениях.

4) Спорт. Методы позволяют анализировать движение игроков, выявляя тактические отклонения, ошибки, отслеживание траектории мяча, разрешение спорных ситуаций.

5) Промышленность и робототехника. Задачи применяются для мониторинга работы конвейера, анализ движения роботов. Используя инструментарий видеоаналитики, становится возможным определять явные нарушения в стабильности производственных процессов предприятий в конкретных местах производственной цепочки. Отслеживание статистики выявления подобных нарушений способствует своевременной локализации возможных технических сбоев в работе автоматизированного как оборудования, так и эксплуатируемого программного обеспечения. По мере увеличения точности такого анализа становится возможным получать сведения об износе автоматизированных систем.

Современный подход к проектированию роботизированных систем, который заключается в замене человеческого ресурса, сводится к имитации как органов восприятия информации, так и реализации конкретных механизмов (подвижных частей) для реализации поставленных задач в имеющихся условиях. Методы анализа видеопоследовательностей обеспечивают разработчиков и конструкторов широким перечнем инструментов для реализации высокотехнологичных технических процессов, которые недоступны человеку ввиду его естественных ограничений, либо с целью заменить человеческий ресурс для увеличения общей производительности труда автоматизированного производства.

ГЛАВА 3

ПОСТАНОВКА ЗАДАЧИ ОБНАРУЖЕНИЯ АНОМАЛИЙ НА ВИДЕОПОСЛЕДОВАТЕЛЬНОСТИ. ПРИМЕНЕНИЕ МЕТОДОВ И АЛГОРИТМОВ АНАЛИЗА ВРЕМЕННЫХ РЯДОВ

3.1 Постановка задачи

Для реализации методов и алгоритмов был рассмотрен видеофайл, содержащий запись дорожного движения. В данной работе поставлена задача выявления аномалий в скорости автомобилей с использованием методов обработки временных рядов и компьютерного зрения. Задача включает несколько этапов:

1) Анализ видеофайла. Этап состоит из оценки качества видео, определения разрешения и частоты кадров, анализа условий съемки. На этом шаге важно проверить наличие помех и искажений, а также определить область интереса, чтобы сосредоточиться на определенном участке.

2) Детекция автомобилей в каждом кадре. Этап подразумевает идентификацию автомобиля и сохранение его координат для дальнейшей обработки.

3) Трекинг объектов. После детекции объектов следует обеспечить их отслеживание между кадрами. Этап включает в себя связывание объектов между кадрами. Это поможет отслеживать автомобиль даже в случае, когда он скроется за препятствующим объектом в кадре.

4) Извлечение временного ряда. Исходя из прошлых методов следует сформировать временной ряд, на основе которого будет происходить дальнейшее выявление аномалий.

5) Обработка временного ряда. Необходимо провести обработку временного ряда. Обработка может включать в себя сглаживание данных, нормализацию, исключение пустых значений.

6) Критерий для аномалий. Следует установить пороговые значения для нормальной скорости на участке дороги. Аномалии могут включать скорости, выходящие за пределы заданного доверительного интервала минимального и максимального порога.

7) Визуализация аномалий. Построение графика временного ряда скоростей с выделением аномальных значений. Генерация отчетов с графиками позволяет удобно анализировать результаты.

3.2 Структура программы. Применение методов и алгоритмов анализа временных рядов для решения поставленной задачи

Создание четкой структуры проекта является важным моментом, направленным на предотвращение путаницы и упрощение разработки. Программа состоит из нескольких модулей, каждый модуль выполняет свои функции. Структура позволяет создать многофункциональную архитектуру, которая легко адаптируется под разные задачи.

Рассмотрим реализацию четкой архитектуры и соответствующей ей структуры проекта. Качество проведения работ на текущем этапе позволяет обеспечить дальнейшую масштабируемость проекта и гибкость при внесении различного масштаба изменений. Качественная структура проекта также позволяет обеспечить низкий порог входа сторонних разработчиков в проект, способствуя поддержанию кодовой базы в актуальном состоянии.

Каждый модуль рассматриваемого проекта отвечает за соответствующий функционал, конечной целью которого является обработка видеофайла. Следование принятым стандартам и подходам к организации четкой структуры позволяет проводить дальнейшее тестирование и отладку каждого процесса независимо.

Архитектура представлена на рисунке 3.1.

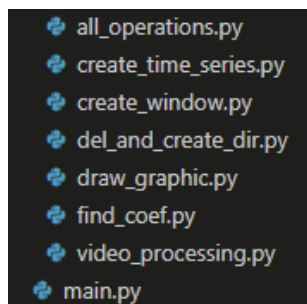


Рисунок 3.1 - Строение проекта

Модуль main.py. Он представляет собой точку входа для запуска программы. В нем инициализируются все необходимые компоненты и производится настройка среды. По завершению описанных процессов программа переходит на этап отрисовки графического интерфейса.

Модуль create_window.py. В этом модуле создается графический интерфейс, который представлен на рисунке 3.2. Интерфейс обеспечивает эффективное взаимодействие пользователя с программой. Рассматриваемый модуль отвечает за настройку проекта.

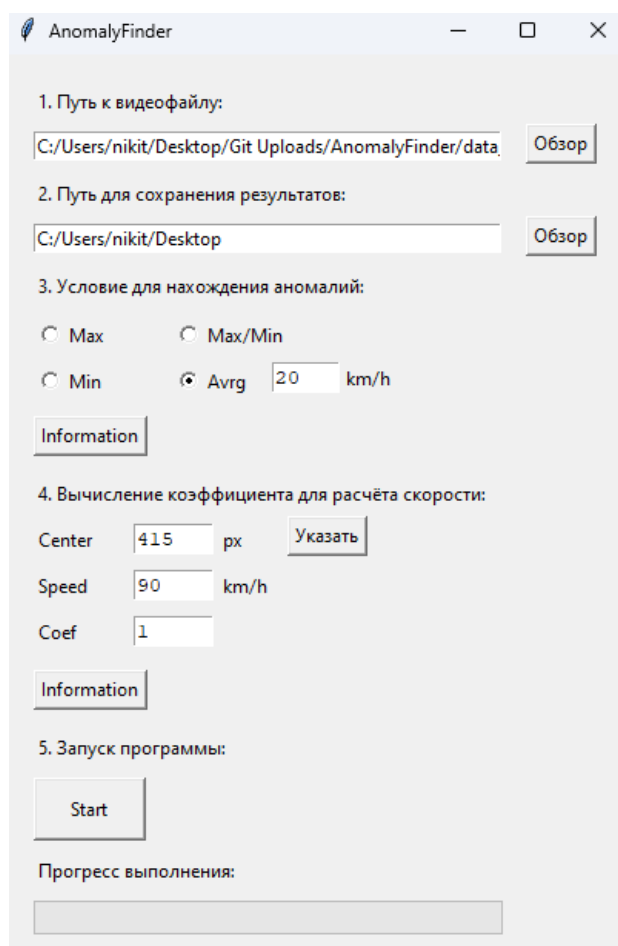


Рисунок 3.2 - Графический интерфейс

Содержание интерфейса:

1. **Путь к видеофайлу.** Элемент интерфейса позволяет пользователю задавать путь к видеофайлу, который задействуется в программе.

2. **Путь для сохранения результатов.** В этом разделе пользователь обязан задать директорию с местоположением конечных результатов. Под результатами подразумевается: создание графика аномалий, создание финального видео, формирование временного ряда, сохранение фотографии каждого объекта, сохранение фотографии аномального объекта.

3. **Условие для нахождения аномалий.** В этом разделе пользователь настраивает параметры для нахождения аномалий.

1) **Условие максимума (Max).** Аномалия определяется как максимальное значение, а также те значения, которые составляют не менее 80% от значения максимума. Это условие для выявления аномалии, связанной с превышением установленных норм.

2) **Условие минимума (Min).** В этом случае аномалия определяется как минимальное значение и значения, которые составляют не более 120% от минимума.

3) Объединение двух предыдущих условий (Min/max). Условие позволяет объединить параметры Min и Max.

4) Условие среднего (Avg). Аномалия определяется как значение, которое превышает среднее значение на величину N , что позволяет выявить аномалии, отклоняющиеся от среднестатистических показателей.

4. Вычисление коэффициента для расчета скорости. Вычисление необходимо для корректного определения скорости в связи с учетом масштаба дороги на изображении.

1) Параметр Center. Пользователь задает центр пути. Начало пути – момент, когда камера впервые начинает отслеживать объект. Конец пути – момент, когда камера прекращает отслеживать объект. Параметр позволяет установить границу, которая определяет момент начала демонстрации текущей скорости.

2) Параметр Speed. Дополнительный параметр, который используется для вычисления коэффициента. Например, если скорость ограничена дорожным знаком, то значение Speed будет этим ограничением.

5. Запуск программы. На данном этапе инициализируются все процессы. Пользователь может наблюдать за прогрессом с помощью шкалы прогресса.

Модуль `all_operations.py`. Модуль отвечает за последовательный запуск всех процессов. Структура модуля явно отображает преимущества описываемого подхода в этой главе по организации четкой архитектуры проекта, способствующей модульной отладке и тестированию.

Модуль `del_and_create_dir.py`. Задача модуля заключается в менеджменте рабочего пространства: своевременный запуск процессов очистки рабочего пространства, удаление старых директорий и создание новых, поддержание целостности папочной структуры.

Модуль `video_processing.py`. Является основным модулем, отвечающим за вычисление скорости, произведение детекции, классификации и трекинга.

YOLOv8 — используемая в проекте современная модель глубокого обучения, предназначенная для реализации задачи детекции и классификации объектов на изображении. Цель модуля – выделение объектов в реальном времени с высокой точностью.

YOLOv8 использует уникальный подход к детекции объектов, который позволяет эффективно анализировать изображение. Принцип работы сводится к разбиению входного изображения на ряд изображений с последующим наложением сетки фиксированного размера. Визуализация процесса представлена на рисунке 3.3. Каждая ячейка отвечает за обнаружение объектов, находящихся в ее пределах. Разделение позволяет модели сосредоточиться на различных участках изображения и эффективно отлавливать объекты, даже в случае, если они находятся близко друг к другу.

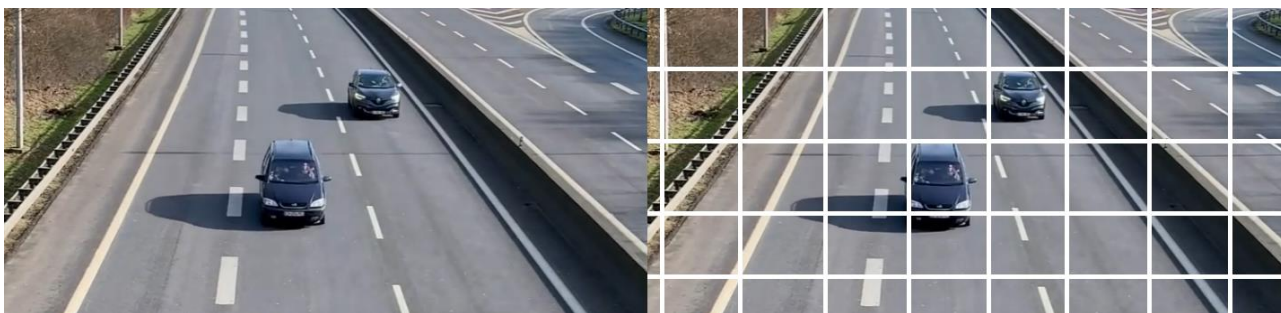


Рисунок 3.3 - Разбиение изображения сеткой

Для каждой ячейки сетки модель применяет сверточные нейронные сети. Сверточная нейронная сеть «видит» изображение в особом представлении - в виде матриц. В RGB цвет закодирован в виде трех целых чисел от 0 до 255. Каждый пиксель матрицы представлен в виде числа, которое показывает яркость в этом диапазоне, представление изображения в виде матрицы на рисунке 3.4.

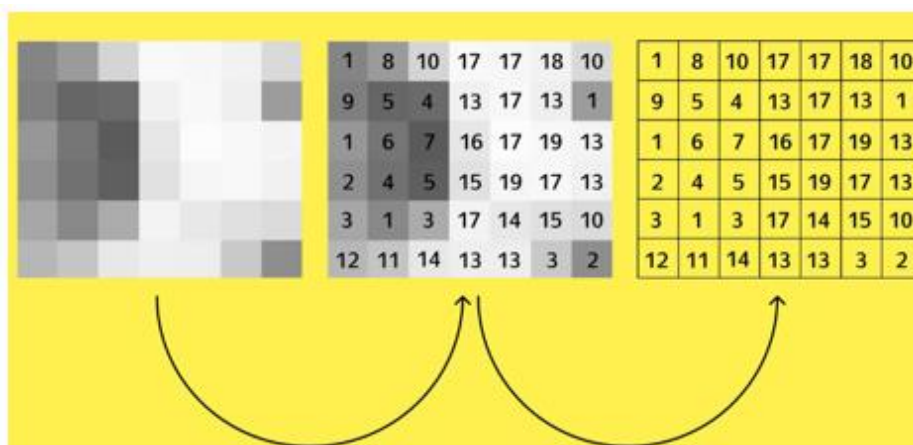


Рисунок 3.4 - Представление изображения в виде матрицы

Чтобы нейросеть распознала машину, необходимо проделать с изображением несколько типовых операций на каждом слое. Пример фотографии машины на рисунке 3.5.



Рисунок 3.5 - Фотография машины

Главной операцией в данном процессе является **свертка**. При выполнении свертки нейронная сеть выполняет фильтрацию, сохраняя только наиболее важные для последующего анализа элементы изображения – такие как контуры объектов и границы. Для различных типов признаков могут быть созданы отдельные свертки. В процессе работы сети на каждом сверточном слое происходит автоматический подбор оптимальных параметров для распознавания и классификации. Наглядное представление архитектуры такой сверточной нейронной сети приведено на рисунке 3.6.

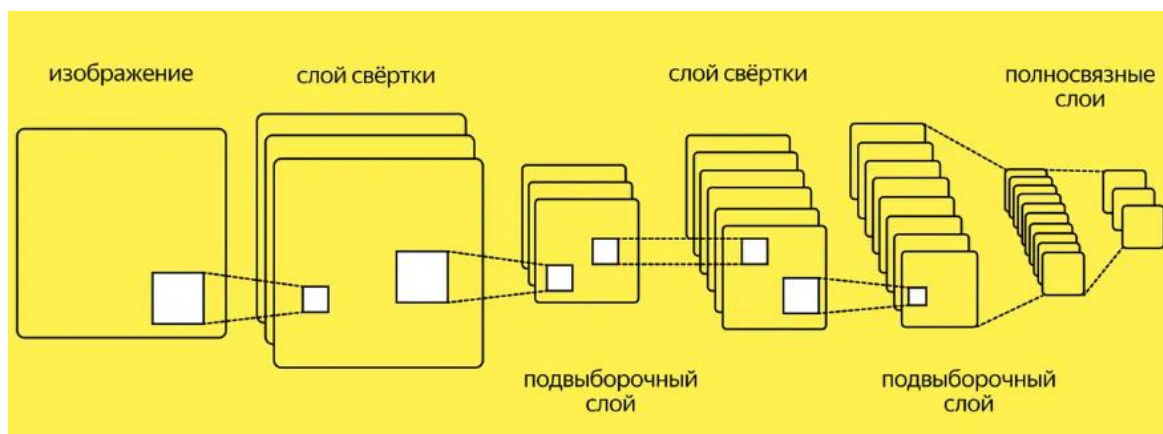


Рисунок 3.6 - Структура сверточной нейронной сети

В архитектуре сверточных нейронных сетей за слоем свертки следует слой подвыборки (пулинга). Данный слой отбирает наиболее информативные признаки, извлеченные предыдущим сверточным слоем, одновременно уменьшая размерность данных за счет устранения менее значимых элементов.

Результат, полученный после операции пулинга может снова поступать на вход следующего сверточного слоя. Такая последовательность операций может повторяться многократно, формируя тем самым глубокую иерархию признаков.

Работа нейросети организована по принципу постепенного усложнения признаков. Первоначальные слои анализируют минимальные фрагменты изображения (размером 2x2 или 3x3 пикселя), что позволяет выявлять лишь элементарные особенности. По мере углубления в архитектуру сети, слои начинают оперировать все более сложными признаками, комбинируя простые элементы в сложные структуры. Финальные слои сети выполняют задачу классификации, сопоставляя выявленные признаки с конкретными объектами.

Основным инструментом сверточных слоев являются специальные фильтры (ядра свертки), представленные в виде числовых матриц. Процесс свертки можно рассмотреть на конкретном примере: для пикселя со значением 2 (как показано на рисунке 3.7) выполняется поэлементное умножение его значения и соседних пикселей на соответствующие элементы матрицы фильтра с

последующим суммированием результатов. Полученное значение (в данном случае 13) заменяет исходное значение центрального пикселя, в то время как соседние пиксели остаются без изменений. Данная операция последовательно применяется ко всем пикселям обрабатываемого изображения.

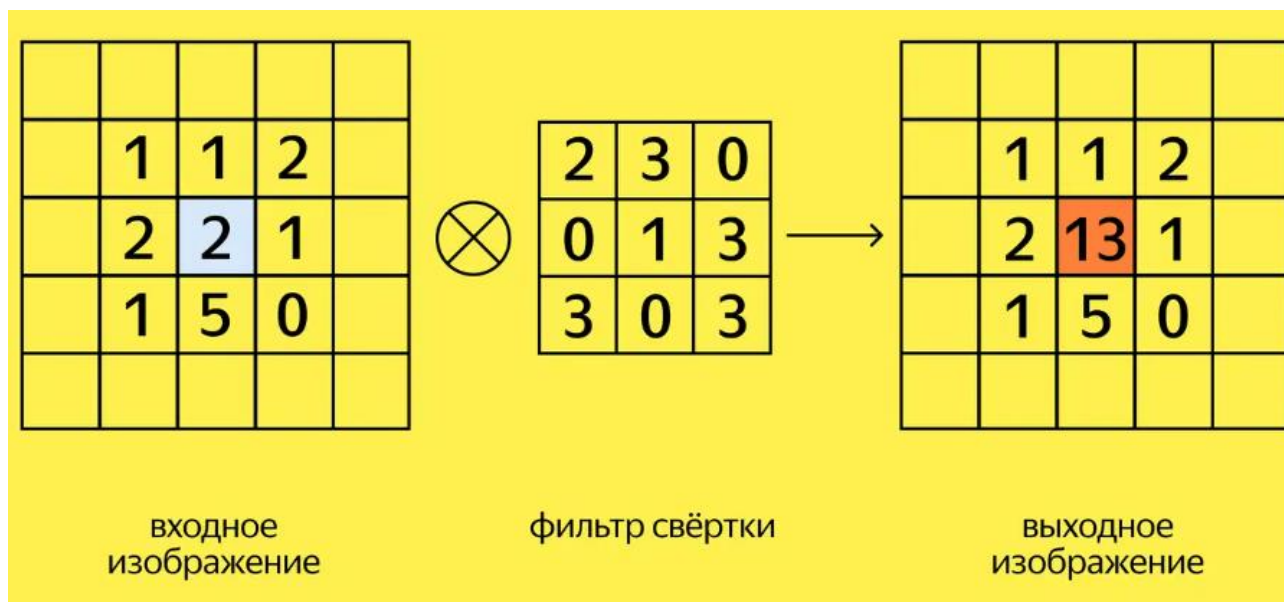


Рисунок 3.7 - Свертка на примере одного пикселя

Процесс свертки изображения осуществляется следующим образом:

1) Применение фильтра. Ядро свертки последовательно перемещается по всем пикселям входного изображения. На каждой позиции происходит поэлементное умножение значений фильтра на соответствующий участок изображения. Результаты умножения суммируются, формируя элементы новой выходной матрицы.

2) Формирование карты признаков. Все вычисленные значения объединяются в единую матрицу активаций. К каждому элементу добавляется смещение, соответствующее величине шага фильтра. Величина шага определяет расстояние между позициями фильтра. Каждая отдельная карта признаков представляет собой финальную матрицу для одного канала.

3) Построение выходного тензора. Все обработанные матрицы и каналы объединяются в трехмерный тензор. Итоговое изображение характеризуется как изменением пространственного размера, так и новым количеством каналов.

Состояние матрицы для RGB представлено на рисунке 3.8.

После применения фильтров на различных слоях сети формируется карта признаков, которая представляет собой матрицы, содержащие информацию о выявленных характеристиках. Используя данные, можно классифицировать объект, находящийся в пределах каждой ячейки сетки.

Модель сверточной нейронной сети приведена в статье [9].

Модель анализирует кадры в видеопоследовательности и согласно этому выделяет автомобили с помощью рамок **Bounding Boxes**, рисунок 3.9.

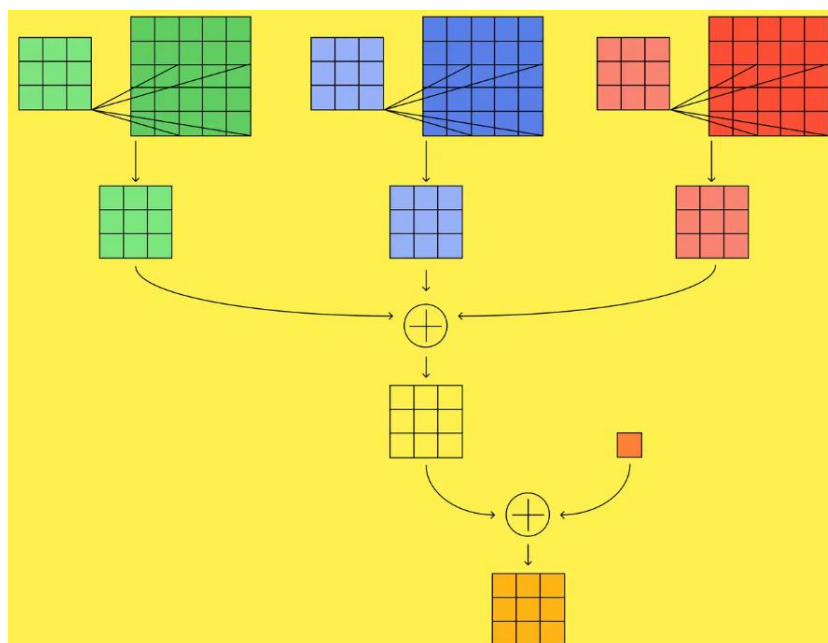


Рисунок 3.8 - Составление матрицы для RGB



Рисунок 3.9 - Детекция автомобиля

Для отслеживания автомобилей между кадрами используется **алгоритм SORT**. Он позволяет сопоставить объекты в последовательных кадрах, формируя траектории движения, рисунок 3.10. В основании метода лежит фильтр Калмана, который используется для предсказания положения объектов на следующем кадре.

Опираясь на полученные значения координат объектов и их временных меток, для каждого кадра вычисляется скорость автомобиля.



Рисунок 3.10 – Траектория движения

$$speed = \frac{|current_position[1] - previous_position[1]|}{time_difference} \cdot coef, \quad (3.2.1)$$

где *current_position* – текущая координата по оси *Y*, *previous_position* – предыдущая координата по оси *Y*, *time_difference* – разница во времени между текущим и предыдущим отслеживанием, *coef* – коэффициент для масштабирования.

Модуль **create_time_series.py** отвечает за создание временного ряда. Визуализация его представлена на рисунке 3.11.

Ряд состоит из нескольких столбцов: «id» - уникальные идентификатор, «coordinates(x;y)» – координаты автомобиля в момент времени *t*, «time(s)» – время, за которое автомобиль перешел из одного кадра в другой, «distance(px)» – расстояние, пройденное автомобилем, «speed(km/h)» – скорость автомобиля.

Таким образом, модуль формирования временного ряда обеспечивает структурированное представление данных, что является основой для дальнейшего анализа и визуализации.

Модуль **draw_graphic.py** создает график, на котором по оси *x* отложены номера объектов, а по оси *y* их скорость (рисунок 3.12). Горизонтальные, прерывистые, цветные линии обозначают границы для аномалий. Описание линий представлено на сноске в легенде.

Вертикальные столбцы отображают объекты, а точки на столбцах обозначают состояние каждого объекта. Зеленые точки указывают на нормальные значения скорости, в то время как красные точки обозначают аномалии. Красные точки подписываются меткой «Аномалия», что позволяет легко идентифицировать проблемные случаи.

Модуль **find_coef.py** обеспечивает корректное вычисление скорости автомобилей. Для адекватной интерпретации скорости необходимо учитывать масштаб дороги и конвертацию пикселей в метры.

id	coordinates(x;y)	time(s)	distance(px)	speed(km/h)
1	(1030.50, 398.60)	0.0	0	95.72
1	(1035.76, 400.48)	0.305	5.59	100.41
1	(1039.23, 404.85)	0.493	11.16	107.76
1	(1042.04, 409.51)	0.658	16.61	96.16
1	(1044.45, 415.60)	0.825	23.15	85.47
1	(1044.88, 420.22)	1.142	27.80	75.00
1	(1048.17, 425.86)	1.326	34.32	63.67
1	(1052.10, 432.74)	1.508	42.24	60.21
1	(1055.56, 439.56)	1.676	49.89	74.37
1	(1059.88, 446.29)	1.860	57.89	71.82
1	(1060.75, 452.77)	2.041	64.43	74.55
1	(1066.48, 458.76)	2.224	72.72	65.90
1	(1070.75, 464.90)	2.409	80.20	60.70
1	(1075.62, 473.12)	2.608	89.75	51.88
1	(1078.44, 481.70)	2.823	98.78	68.19
1	(1082.88, 491.44)	3.088	109.49	66.91
1	(1088.29, 500.76)	3.338	120.26	70.34
1	(1092.08, 509.98)	3.588	130.23	72.83
1	(1095.46, 518.41)	3.838	139.31	73.19
1	(1102.88, 530.03)	4.026	153.10	70.48
1	(1109.62, 541.95)	4.208	166.79	67.06
1	(1117.44, 554.97)	4.375	181.99	76.42
1	(1124.17, 569.03)	4.558	197.56	63.55

Рисунок 3.11 – Временной ряд

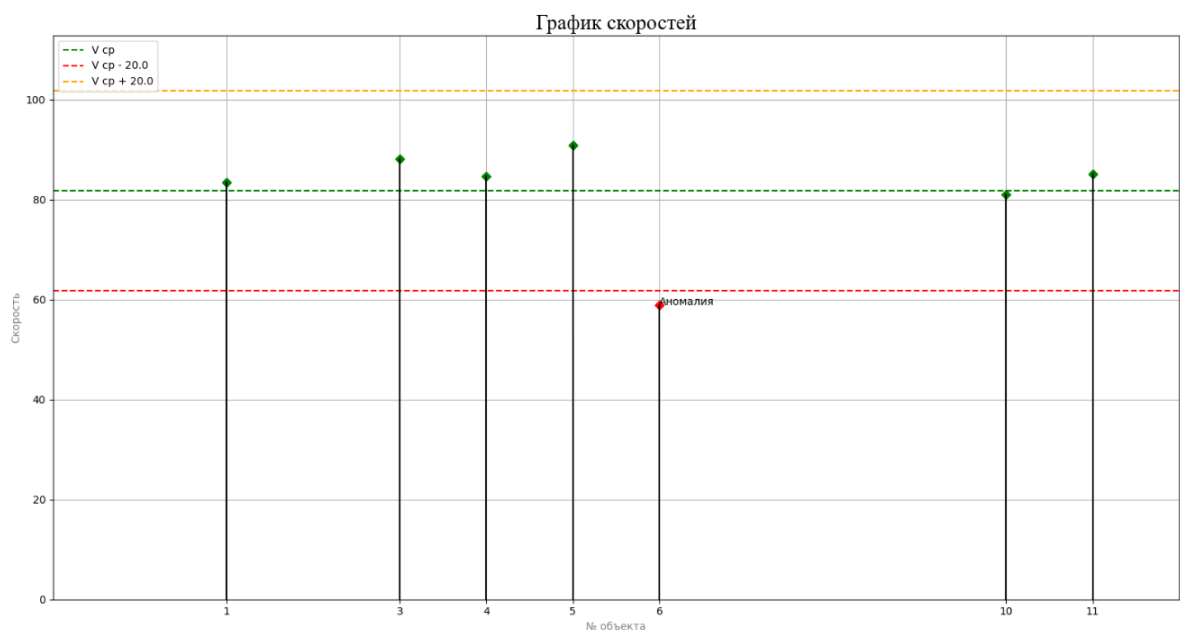


Рисунок 3.12 - График аномалий

Начальное значение коэффициента устанавливается равным единице, что служит базой для дальнейших вычислений. В процессе работы программы коэффициент может изменяться в зависимости от характеристик конкретного видеофайла и условий съемки.

Формула для вычисления коэффициента:

$$coef = \frac{speed \cdot len(speed_obj.spd)}{sum}, \quad (3.2.2)$$

где *speed* - значение ограничения (например, если есть знак ограничения скорости), *len(speed_obj.spd)* - количество всех объектов, *sum* - сумма всех скоростей объектов.

Коэффициент вычисляется в конце, так как программа должна обработать полученные данные.

Для написания программного кода и описания модели YOLOv8 использовалась статья [3].

3.3 Визуализация временных рядов и их сегментация

Графически представим визуализацию временных рядов без сегментации. Под сегментацией понимаем процесс обработки временного ряда, при котором удаляются все лишние значения: нулевые значения и значения, которые явно отличаются от нормальных данных. Также изобразим линию скользящего среднего.

Представим визуализацию временных рядов с сегментацией. К графику добавим линейный прогноз, полиномиальный прогноз (степени 3) и прогнозируемые значения согласно полиномиальному прогнозу.

В завершение покажем вывод из консоли, включая последнее значение, прогнозируемое значение и результат роста тренда. Графики приведены на рисунках 3.13-3.21.

На основе представленных данных можно сделать следующие выводы:

1. Влияние сегментации. Сегментация временного ряда позволяет более точно анализировать данные, удаляя шум и аномалии, что может улучшить качество прогнозов.

2. Сравнение прогнозов. Сравнение линейного и полиномиального прогнозов позволяет оценить, какой метод лучше подходит для данной временной серии.

3. Тенденция роста. Анализ прогнозируемых значений и последнего значения дает представление о текущем тренде. Если прогнозируемые значения показывают рост, это может указывать на устойчивую положительную динамику.

4. Скользящее среднее. Линия скользящего среднего помогает визуализировать долгосрочные тренды и сгладить краткосрочные колебания, что облегчает интерпретацию данных.

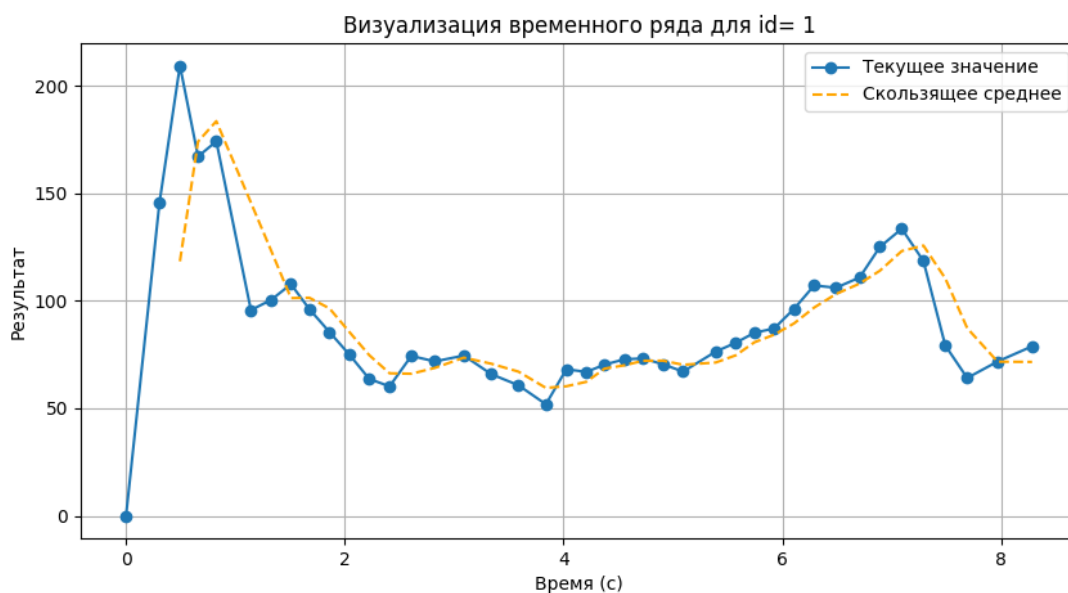


Рисунок 3.13 – Визуализация временного ряда для id=1

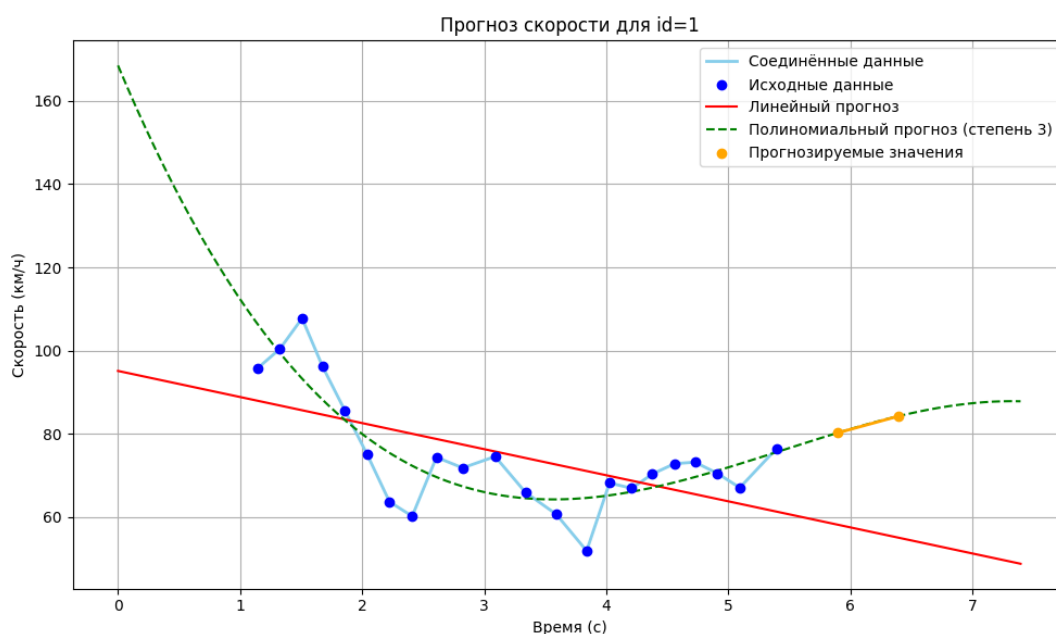


Рисунок 3.14 – Визуализация сегментированного временного ряда для id=1 с линейным прогнозом, полиномиальным прогнозом (степень 3) и прогнозируемым значением

id=1:
 Последнее время: 5.39 с, последняя скорость: 76.42 км/ч
 Прогноз на 5.89 с: 80.22 км/ч
 Прогноз на 6.39 с: 84.25 км/ч
 Тренд: ☒ Скорость будет расти

Рисунок 3.15 – Вывод прогноза для id=1 с консоли

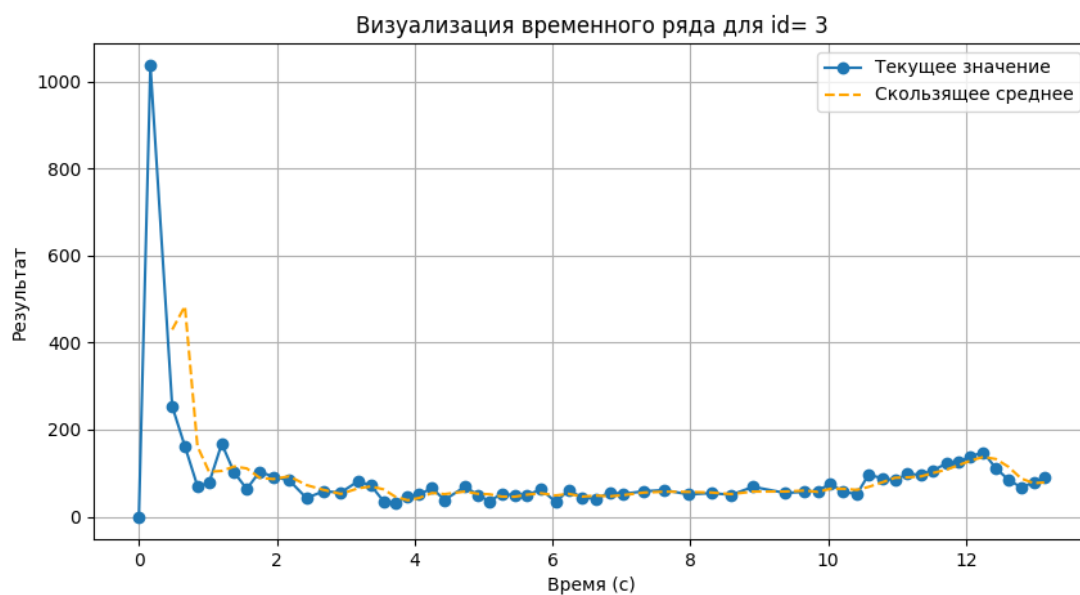


Рисунок 3.16 – Визуализация временного ряда для id=3

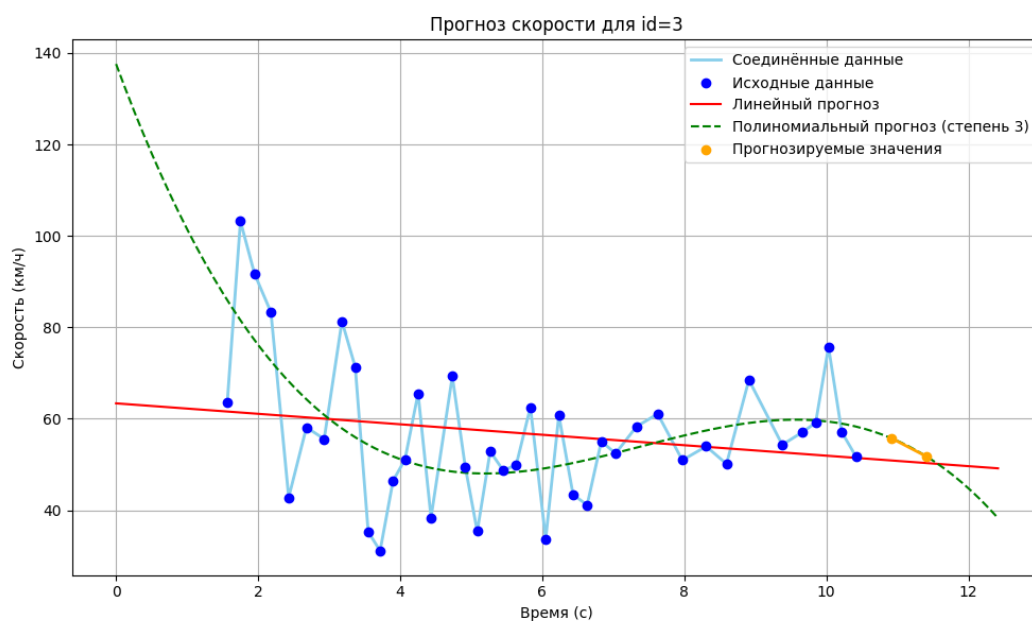


Рисунок 3.17 – Визуализация сегментированного временного ряда для id=3 с линейным прогнозом, полиномиальным прогнозом (степень 3) и прогнозируемым значением

id=3:
 Последнее время: 10.42 с, последняя скорость: 51.75 км/ч
 Прогноз на 10.92 с: 55.74 км/ч
 Прогноз на 11.42 с: 51.69 км/ч
 Тренд: ❌ Скорость будет падать или оставаться неизменной

Рисунок 3.18 – Вывод прогноза для id=3 с консоли

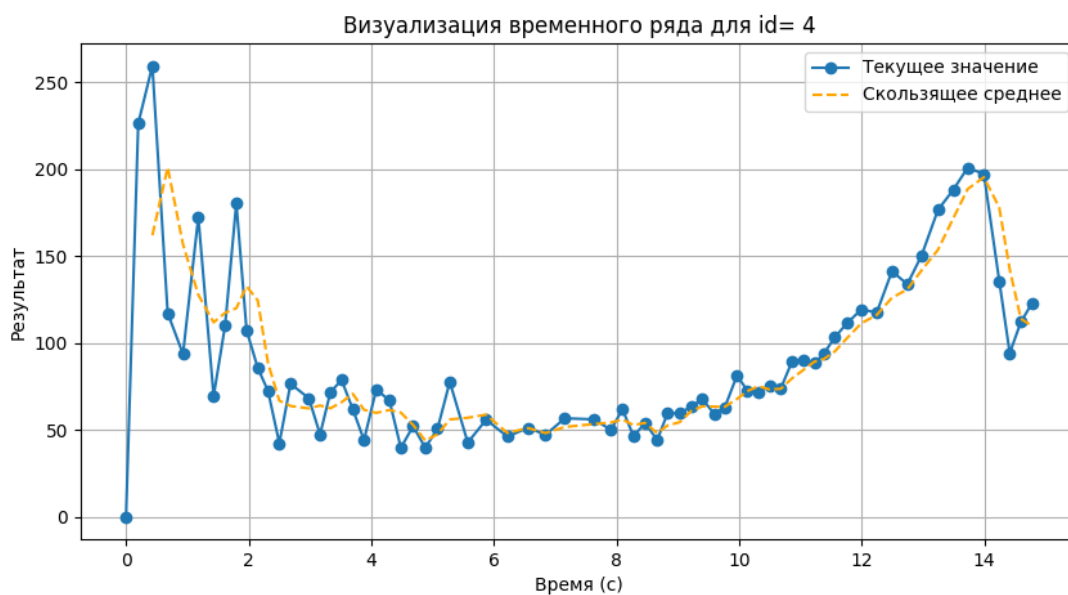


Рисунок 3.19 – Визуализация временного ряда для id=4

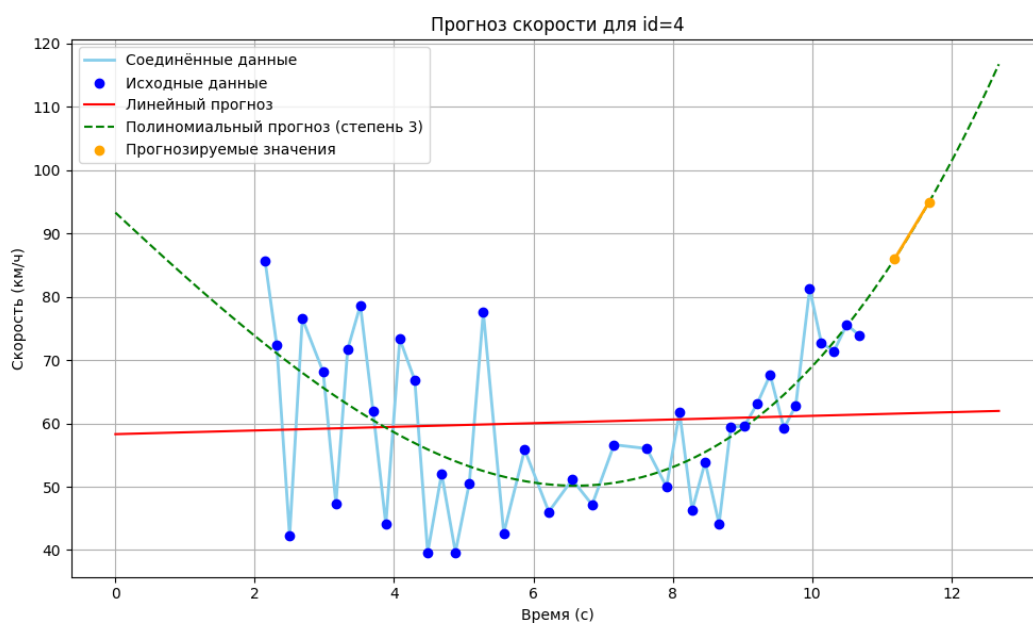


Рисунок 3.20 – Визуализация сегментированного временного ряда для id=4 с линейным прогнозом, полиномиальным прогнозом (степень 3) и прогнозируемым значением

id=4:
 Последнее время: 10.68 с, последняя скорость: 73.85 км/ч
 Прогноз на 11.18 с: 85.90 км/ч
 Прогноз на 11.68 с: 94.95 км/ч
 Тренд: ☒ Скорость будет расти

Рисунок 3.21 – Вывод прогноза для id=4 с консоли

3.4 Реализация модели на другой видеопоследовательности

С целью демонстрации работы разработанного программного обеспечения рассмотрим видеофайл, содержащий запись движения людей в торговом центре. К данному видеофайлу необходимо повторно применить готовый алгоритм для выявления аномалий на видеопоследовательности. Стоп кадр такого видеофайла представлен на рисунке 3.22.



Рисунок 3.22 – Исходный видеофайл

Отличительными особенностями этого исходного файла, используемого в качестве входных данных, являются:

- 1) Статичное положение камеры. Наибольшая эффективность алгоритма проявляется при анализе динамических объектов относительно статического фона.
- 2) Достаточная освещенность сцены и, как следствие, высокий уровень контраста изображения, позволяющая повышать эффективность работы алгоритма по определению границ подвижных объектов.
- 3) Допустимый уровень активности, позволяющий получать на регулярной основе примеры выявленной аномальной активности: наличие как активно передвигающихся людей в кадре, так и примеры относительно неподвижно стоящих людей.

Реализация методов классификации и детекции, рисунок 3.23. На приведенном скриншоте работы программы можно выявить следы процесса детекции подвижных элементов видеопоследовательности и постоянства идентификации отдельных людей.



Рисунок 3.23 – Результат детекции и классификации

Визуальное отображение процесса работы алгоритма трекинга наглядно показано на рисунке 3.24. Каждый объект, отображенный на части скриншота, помечен отдельным уникальным цветом, также можно рассмотреть следы подвижности относительно статичного положения кадра в виде характерных непрерывных линий.



Рисунок 3.24 – Результат трекинга

Дополненный вариант демонстрации работы алгоритма отслеживания динамических объектов в кадре представлен на рисунке 3.25. Отображена собранная статистика по каждому подвижному объекту в виде значений скоростей в момент пересечения заданных пользователем линии.

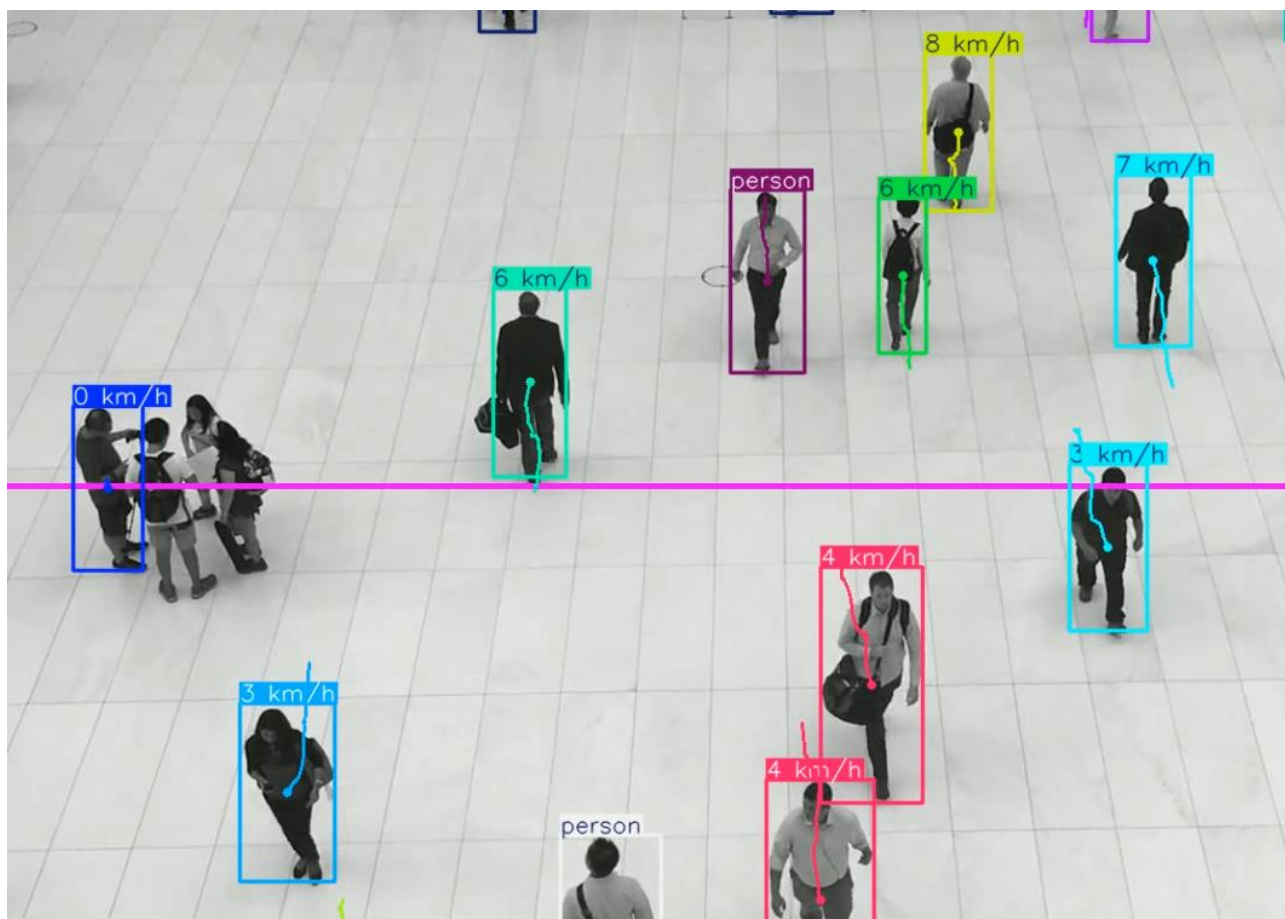


Рисунок 3.25 – Вычисление скорости

Демонстрацию результата работы метода выявления аномалий наглядно отображена на рисунке 3.26. На скриншоте представлена статистика по вычисленной скорости анализируемой сцены, при этом совокупность данных градуируется относительно трёх условных пунктирных уровней. Вертикальная ось графика содержит значение скоростей, при этом абсолютные значения трех условных уровней вычисляются относительно общей статистики анализируемых объектов: зеленая линия является отображением средней скорости, а две оставшиеся – значение дисперсии средней скорости.

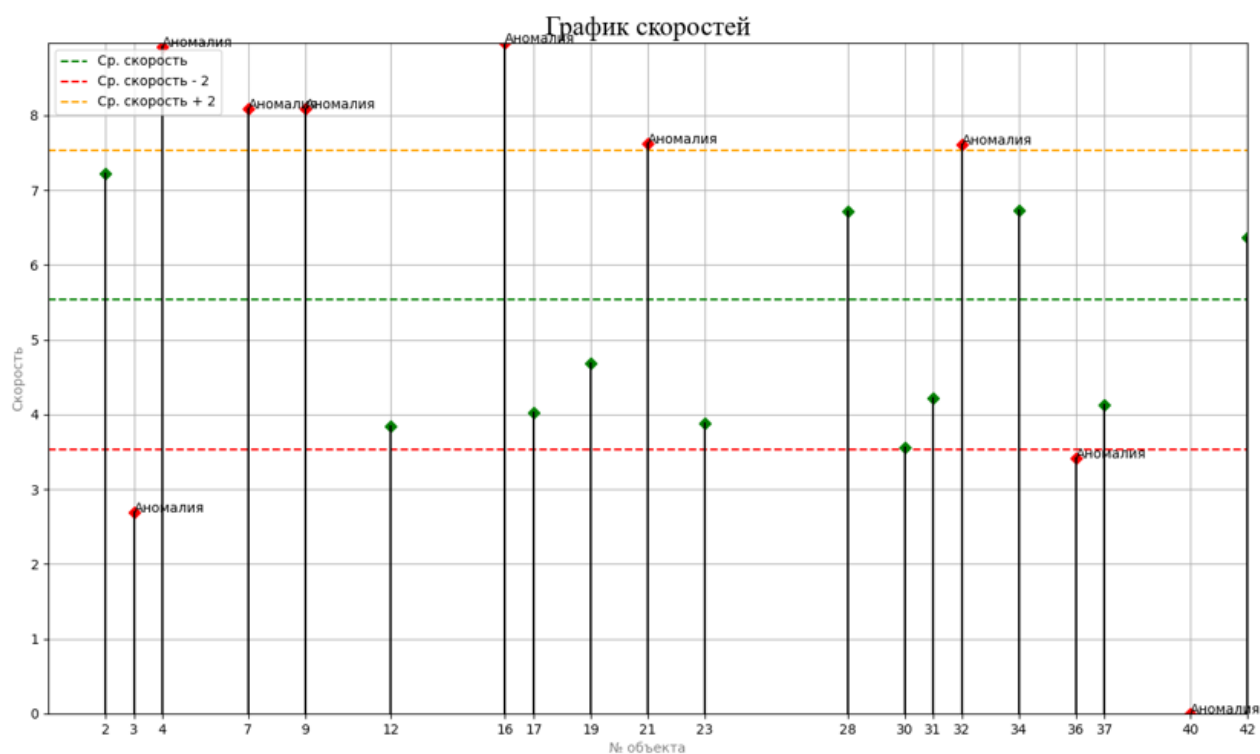


Рисунок 3.26 – Результат выявления аномалий

Итог работы главного анализирующего метода выявил восемь аномалий. Визуальный осмотр результатов позволяет однозначно оценить корректность и стабильность результатов обработки избранных последовательностей видеоизображений. Как следствие, алгоритм может быть эффективно использован в дальнейшем для реализации поставленных задач.

ЗАКЛЮЧЕНИЕ

В рамках данной дипломной работы были исследованы и реализованы основные подходы к решению задачи выявления аномалий на видеопоследовательности при помощи методов обработки временных рядов. В ходе дипломной работы значительное внимание уделено разработке соответствующего программного обеспечения, реализующее теоретическую часть работы. Разработанное программное обеспечение включает в себя использование современных алгоритмов анализа видеоизображений и демонстрирует высокую практическую значимость рассматриваемых подходов.

Анализ видеопоследовательности продемонстрировал высокую эффективность в задаче автоматического детектирования и восстановлении траекторий автомобилей. Видеопоток был успешно обработан.

В ходе анализа были вычислены ключевые параметры движения для каждого автомобиля. Полученный временной ряд позволил более глубоко исследовать динамику движения. Аномалии были автоматически обнаружены и визуализированы на видеокадрах и графиках. Графики позволяют быстро и наглядно оценить состояние дорожной обстановки. Выявленные аномалии указывают на потенциальные проблемы на дороге.

Наибольшую эффективность показали методы, основанные на скользящем среднем и экспоненциальном сглаживании при анализе скоростных характеристик. Для обнаружения резких изменений направления движения лучше проявили себя алгоритмы, основанные на преобразованиях Фурье и методах машинного обучения.

Модель YOLOv8 обеспечила точное распознавание автомобилей на каждом кадре, а в сочетании с алгоритмами трекинга, анализа временных рядов и выявлением аномалий, показала свою эффективность в реальном времени.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Агафонов, В.Ю. Статистический анализ временных рядов с использованием модели ARIMA / В.Ю. Агафонов, В.Л. Розалиев, А.В. Заболеева-Зотова // Труды Института статистики и анализа. - 2024. - № выпуска. - С. 2-5.
2. Гафаров, Ф.М. Искусственные нейронные сети и их приложения / Ф.М. Гафаров, А.Ф. Галимянов // Труды Казанского университета. - 2018. - № выпуска. - С. 43-47.
3. Документация и особенности модели. YOLOv8 [Электронный ресурс] Режим доступа: <https://docs.ultralytics.com/ru/models/yolov8/#citations-and-acknowledgements>. - Дата доступа: 20.04.2025.
4. Залесский Б. А. Алгоритм обнаружения движущихся объектов, наблюдаемых видеокамерой // Доклады Национальной академии наук Беларуси. 2023 Т.67, №1. С.20-26.
5. Математическое описание оптического потока [Электронный ресурс]. - Режим доступа: <https://habr.com/ru/articles/201406/>. - Дата доступа: 07.05.2025.
6. Модель Autoregressive Integrated Moving Average. Принципы работы и применение в анализе временных рядов [Электронный ресурс]. - Режим доступа: <https://habr.com/ru/companies/otus/articles/732080/>. - Дата доступа: 03.03.2025.
7. Тисецкий, А.Т. Обзор методов сегментации и обнаружения объектов на изображении в реальном времени / А.Т. Тисецкий, Т.П. Мансурова // Труды Красноярского государственного университета. - 2022. - № выпуска. - С. 5-10.
8. Яндекс Практикум. Autoregressive Integrated Moving Average [Электронный ресурс]. - Режим доступа: <https://education.yandex.ru/handbook/ml/article/modeli-vida-arima>. - Дата доступа: 03.03.2025.
9. Яндекс Практикум. Сверточные нейронные сети: принцип работы и применение [Электронный ресурс]. - Режим доступа: <https://practicum.yandex.ru/blog/svertochnye-neyronnye-seti/>. - Дата доступа: 06.05.2025.

10. GitHub [Электронный ресурс]. – Режим доступа: <https://www.kaggle.com/datasets/bladerunnerrachel/ucsd-anomaly-dataset>. – Дата доступа: 05.02.2025.
11. Duong, H.-T. Deep Learning-Based Anomaly Detection in Video Surveillance: A Survey / H.-T. Duong, V.-T. Le, V. T. Hoang // *Sensors*. – 2023. – Vol. 23, Iss. 11. – Article No. 5024. – 25 p. – DOI: 10.3390/s23115024.
12. FlowNet 2.0: Evolution of Optical Flow Estimation With Deep Networks / E. Ilg [et al.]. // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. – 2017. – P. 2462–2470. – URL: https://openaccess.thecvf.com/content_cvpr_2017/html/Ilg_FlowNet_2.0_Evolution_CVPR_2017_paper.html.
13. Hyndman, R.J. Principles and practice [Electronic resource]. – Principles and practice: <https://otexts.com/fpp3/>. - Access date: 20.03.2025