Интеллектуальная система отслеживания активностей на основе больших нейросетевых моделей

Р. Р. Миннеахметов

Казанский федеральный университет

razil0071999@gmail.com

Аннотация

В статье представлен подход построения интеллектуальной системы отслеживания активностей с использованием больших нейросетевых моделей. Основное внимание уделено применению языковых моделей и моделей компьютерного зрения для анализа данных видеонаблюдения, сенсорных сигналов и логов. В качестве инструмента реализации использован локальный фреймворк Ollama, обеспечивающий безопасную и автономную работу больших языковых моделей. Предложен прототип системы, описана её архитектура, методика обработки данных и экспериментальная часть. Показано, что использование больших моделей позволяет автоматизировать процесс анализа и повысить точность обнаружения аномалий.

Ключевые слова: Ollama, большие языковые модели, отслеживание активностей, видеоаналитика, искусственный интеллект

Библиографическая ссылка: Миннеахметов Р. Р. Интеллектуальная система отслеживания активностей на основе больших нейросетевых моделей // Информационное общество: образование, наука, культура и технологии будущего. Выпуск 9 (Труды XXVIII Международной объединенной научной конференции «Интернет и современное общество», IMS-2025, Санкт-Петербург, 23 – 25 июня 2025 г. Сборник научных статей). — СПб: Университет ИТМО, 2025. С. 127-137. DOI: 10.17586/3033-5574-2025-9-127-137.

1. Введение

Современные большие языковые модели (Large Language Models — LLM) демонстрируют высокую эффективность в разнообразных приложениях — от обработки естественного языка до задач анализа данных и поддержки принятия решений [1, 2]. Развитие глубокого обучения в области компьютерного зрения также позволило достигнуть значительных успехов в распознавании объектов и действий на видеоданных [3, 4]. Однако многие практические задачи мониторинга и обеспечения безопасности требуют одновременного анализа нескольких типов данных (видео, показания датчиков, системные логи), что традиционно осуществлялось разрозненными специализированными методами [5, 6]. Например, для отслеживания активности людей и выявления инцидентов на промышленном объекте может потребоваться совмещение данных видеонаблюдения, показаний носимых сенсоров и записей систем контроля доступа. Существующие подходы часто сосредоточены на одном типе данных: либо на сенсорных данных [7, 8], либо на видеопотоке [3, 4, 9], либо на анализе логов [10]. Это создаёт разрыв в интегрированной оценке обстановки и осложняет своевременное обнаружение комплексных аномалий.

Недавно появилось множество работ, предлагающих применение больших моделей для отдельных видов данных: так, в [1] исследуется использование LLM для распознавания активности по данным носимых устройств, а в [10] предложен подход LogGPT для обнаружения аномалий в журнальных записях с помощью языковой модели. Тем

не менее, проблема единого анализа мультимодальных данных с помощью больших моделей остаётся малоизученной. Большие языковые модели обладают способностью обобщать знания и гибко интерпретировать текстовые описания сложных ситуаций, а мультимодальные версии таких моделей могут обрабатывать визуальную информацию [11, 12, 13, 14]. Данная работа направлена на заполнение указанного пробела: исследуется, насколько эффективно большие языковые модели могут анализировать и объединять разнородные данные (изображения, параметры сенсоров, текстовые события) для выявления нештатных ситуаций. В статье представлена архитектура прототипа интеллектуальной системы мониторинга, реализованного на основе LLM, а также результаты экспериментальной оценки её точности и производительности. В последующих разделах приведены постановка задачи, обзор существующих решений, описание предлагаемого метода, детали эксперимента, обсуждение полученных результатов и заключение.

2. Обзор существующих решений

Для эффективного отслеживания активностей необходимо анализировать разнородные данные: видеопотоки, логи систем, показания носимых сенсоров, а также контент, создаваемый пользователями (User Generated Content — UGC). Предлагаемая методология базируется на использовании предобученных больших языковых моделей, способных обрабатывать эти данные и извлекать из них значимые паттерны [9]. В части компьютерного зрения применяются глубокие сверточные сети и Vision Transformer-модели для распознавания действий на видео и в реальном времени классифицируют поведения людей [8]. В проведенном анализе последовательностей сенсорных сигналов (ускорение, гироскоп и др.) используются архитектуры на базе LSTM/GRU или Transformer, обученные на больших наборах данных о движениях. Это позволяет моделям выявлять характерные последовательности, соответствующие различным видам активности (ходьба, бег, падение и т. п.) и отклонения от нормы [15]. Для текстовых и логированных данных (журналы событий, отчёты о действиях) задействуются большие языковые модели: они рассматривают последовательности записей как естественный язык и способны выявлять аномалии или критические события по контексту [3, 10].

Рассмотренные нейросетевые подходы уже находят применение в ряде областей. Промышленное производство: большие модели компьютерного зрения используются для контроля действий работников на конвейерных линиях и обеспечения соблюдения техники безопасности. Например, системы на основе глубоких сетей в реальном времени выявляют отсутствие каски или другого средства защиты у сотрудника [16], что позволяет мгновенно реагировать и предотвращать инциденты. Кроме того, анализ вибраций и других сенсорных данных станков с помощью рекуррентных нейросетей помогает реализовать предиктивное обслуживание оборудования — обнаруживать отклонения в работе механизмов и предупреждать аварии [17]. Умные дома: в бытовой среде крупные модели помогают мониторить повседневную активность жителей для повышения удобства и безопасности. Так, анализ данных камер и датчиков движения позволяет определить, что пожилой человек упал, и автоматически вызвать помощь. Носимые устройства (фитнес-браслеты, смарт-часы), оснащённые моделями распознавании человеческой деятельности (Human Activity Recognition — HAR), отслеживают показатели активности и здоровья пользователя, отправляя уведомления при выявлении аномального поведения (длительная неподвижность, аритмия и пр.) [18]. Безопасность и предотвращение рисков: нейросетевые модели активно внедряются в системы видеонаблюдения для распознавания подозрительных действий и ситуаций. С их помощью можно обнаружить на улице оставленный без присмотра предмет или агрессивное поведение в толпе, и предупредить правонарушение. В кибербезопасности модели NLP анализируют сетевые логи и сообщения на наличие характерных паттернов, предшествующих атакам, позволяя оперативно

реагировать на киберинциденты [4]. Ещё одним направлением применения крупных моделей является медицина и здоровье: обработка потоков данных от носимых сенсоров и даже анализ речи/текста пациентов (записи сессий, соцсети) с помощью LLM дают возможность выявлять признаки стресса, депрессии или ухудшения физического состояния на ранних стадиях [1, 20]. Таким образом, индустриальные кейсы демонстрируют универсальность больших нейросетевых моделей: они успешно работают от заводских цехов до домашних условий, повышая эффективность мониторинга и снижая фактор человеческой ошибки.

3. Постановка задачи

Целью исследования является разработка прототипа интеллектуальной системы отслеживания активности на основе больших языковых моделей, способной анализировать мультимодальные данные (видеоизображения, показания сенсоров, текстовые логи) для обнаружения нештатных или аномальных ситуаций. В рамках данной цели решаются следующие задачи:

- спроектировать архитектуру системы, обеспечивающую объединение разнородных источников данных и последовательную обработку их с помощью больших моделей;
- настроить и применить несколько предобученных больших моделей (языковых и мультимодальных) для анализа сгенерированных сценариев активности;
- оценить точность распознавания событий и аномалий по каждому сценарию (с использованием метрик вроде F1-меры) и исследовать производительность моделей (время отклика) при локальном развёртывании.

Основные исследовательские вопросы включают выяснение пригодности и точности современных LLM для анализа нестандартных входных данных, их способности обрабатывать сочетание визуальных и числовых описаний, а также определение ограничений по быстродействию и потребляемым ресурсам при практическом применении.

4. Архитектура системы

Разработанная система состоит из нескольких модулей, обеспечивающих сбор данных, их предварительную обработку и анализ с помощью больших моделей. Общая схема функционирования включает следующие этапы:

1. Сбор данных. Система получает синхронизированные данные из трёх источников: видеокамера (отдельные кадры или короткие видеосегменты), набор датчиков (например, температуры и влажности) и система контроля доступа (журнальные записи событий). Каждый источник фиксирует состояние объекта мониторинга со своей стороны.

2. Предобработка и формирование описаний. На данном этапе «сырые» данные преобразуются в удобный для языка моделей вид. Визуальные данные (видео) либо поступают в модель как изображение (если модель поддерживает картинки на входе), либо описываются текстом с помощью модели-описателя (генерация текстового описания содержимого кадра). Для сенсорных показателей формируется структурированный фрагмент (например, JSON с полями *temperature* и *humidity*), отражающий текущие значения. Текстовые логи событий (например, СКУД) при необходимости фильтруются по релевантности и форматируются (в поле *timestamp* время приводится к стандарту ISO 8601 [20], поле *event* содержит описание самого события). В результате предобработки получается комплект текстовых описаний, представляющих одновременно и содержание видеокадра, и состояние сенсоров, и недавние события.

3. Формирование и подача промпта. Подготовленные текстовые фрагменты объединяются в единый запрос (*prompt*) к языковой модели. Промпт строится таким образом, чтобы модель получила максимум контекста по ситуации. Например, промпт

может содержать описание сцены (либо указание на сам изображение, если модель его принимает), показания датчиков со временем, а также последнее событие из лога. Завершается промпт инструкцией, побуждающей модель сделать вывод о том, является ли ситуация нормальной или содержит признаки аномалии, и пояснить свой вывод. При необходимости у модели запрашивается выдача ответа в формате JSON (используя параметр *format* API Ollama [22]) для облегчения автоматического разбора результата.

4. Анализ модели. Выбранная нейросетевая модель (LLM) обрабатывает переданный промпт, используя своё знание и способности обобщения, и генерирует ответ. В случае мультимодальных моделей изображение анализируется непосредственно внутренним визуальным модулем, иначе модель опирается только на текстовое описание кадра. В ответе модель может либо сформулировать описание ситуации, либо выдать структурированный вывод (например, «*anomaly»: true* вместе с пояснением). В прототипе системы реализован вызов моделей через API фреймворка Ollama [22, 23] (с использованием официальной Python-библиотеки [23]) — запрос содержит идентификатор модели (*model*), текст сформированного промпта (*prompt*) и набор параметров генерации.

5. Интерпретация результата. Сгенерированный ответ анализируется системой. Если модель выдала структурированный JSON, из него извлекаются ключевые поля (например, обнаружено ли отклонение). Если ответ дан в свободной форме, он подвергается разбору с помощью правил или дополнительного запроса, после чего система фиксирует распознанное событие и его статус (норма/аномалия). Данные этого этапа могут быть использованы для оповещения оператора или для последующего обучения системы.

Аппаратно-программная реализация прототипа выполнена на платформе Ollama — это позволяет загружать и запускать большие модели локально, без передачи данных внешним сервисам [22]. В качестве механизма анализа использовались несколько готовых больших моделей из каталога Ollama: gemma3:12b [11], llava:13b [12], llama3.2-vision:11b [13] и minicpm-v:8b [14]. Первые две модели (LLaVA и llama3.2-vision) обладают встроенной возможностью обрабатывать визуальные данные благодаря обучению на паре «изображение–текст», в то время как gemma3 и minicpm-v являются текстовыми LLM общего назначения. Чтобы последние могли работать с видеокадрами, для них на этапе предобработки автоматически генерировалось текстовое описание изображения. Все модели представляют собой нейросети с количеством параметров от 8 до 13 миллиардов, предобученные на больших корпусах данных. Дополнительного дообучения под нашу задачу не выполнялось — модели использовались в имеющемся виде, что позволяет оценить их изначальную способность к мультимодальной интерпретации. Важной частью настройки системы является подбор параметров генерации ответа. В API Ollama доступны температура (temperature), определяющая степень случайности и креативности ответа, и параметр выборки top_p (nucleus sampling), задающий порог охвата вероятностной массы [22]. Также можно ограничить максимальную длину генерируемого отклика и применить штраф за повторения (repeat_penalty) для предотвращения многократного вывода одинаковых фраз. В проведённом эксперименте в целях воспроизводимости основное внимание уделялось качеству содержательного ответа, поэтому параметры генерации выбирались консервативно: температура близкая к 0 (стремясь к детерминированному ответу), top_p = 1 (учитывать всё распределение вероятностей), фиксированная максимальная длина ответа и отключённый потоковый вывод. Такие настройки позволили минимизировать случайные вариации и упростить сравнение моделей по точности и времени.

5. Экспериментальная часть

Для проверки работоспособности предложенного подхода и оценки его эффективности был проведён эксперимент на наборе тестовых сценариев. Поскольку реальный размеченный датасет, содержащий синхронные видео, сенсоры и логи, отсутствует, все

данные были сгенерированы искусственно. Цель генерации заключалась в том, чтобы воспроизвести типичные ситуации, встречающиеся при мониторинге безопасности, включая как штатные, так и аномальные события. Было смоделировано шесть различных сценариев. Каждый сценарий описывал ситуацию в коридоре, отслеживаемом камерой и датчиками. Первые пять сценариев касались видеоданных:

Сценарий 1: человек упал на пол (имитация несчастного случая);

Сценарий 2: человек крадётся вдоль стены (возможное несанкционированное проникновение);

Сценарий 3: двое людей, один нападает на другого (инцидент безопасности);

Сценарий 4: человек стоит непосредственно перед объективом камеры (возможно, пытается её отключить или закрыть);

Сценарий 5: пустой коридор (нормальная ситуация, отклонений нет).

Для каждого из этих случаев с помощью инструментов генерации изображений были получены соответствующие изображения, стилизованные под кадры с камер видеонаблюдения (с умеренным шумом, низким разрешением и чёрно-белой палитрой).

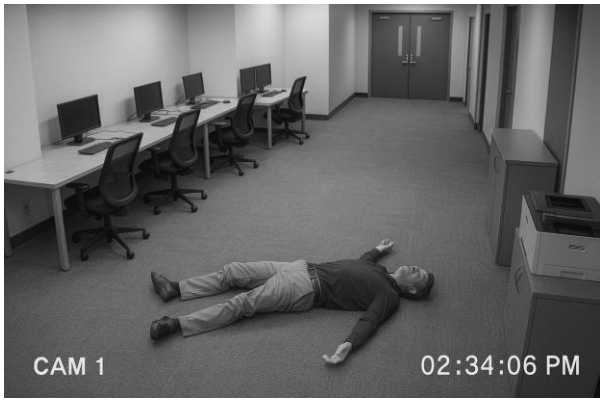

Рис 1. Пример сгенерированного кадра видеонаблюдения для сценария падения человека.

На изображении выше (рис. 1) человек лежит неподвижно на полу пустого коридора, имитируя несчастный случай. Кадр стилизован под запись с реальной камеры: черно-белое изображение с отметкой времени (в правом нижнем углу) и идентификатором камеры (в левом нижнем углу). Такие визуальные данные служат входной информацией для модели, дополняя показания сенсоров и логов контекстом происходящего.

Шестой сценарий был посвящён анализу показаний датчика: имитировалась ситуация пожара, при которой температура резко повысилась, а влажность воздуха упала до ненормально низкого уровнААя. Были сгенерированы пары значений температуры и влажности, соответствующие нормальному состоянию (например, 22°C и 45 % для обычных условий) и экстремальному при пожаре (например, 85°C и 10 %). Дополнительно для контекста формировались текстовые логи системы контроля доступа: в эксперименте использовалась одна примерная запись, отражающая обычное событие (вход сотрудника в определённое время). Лог представлен в формате JSON с полями timestamp (время события в стандарте ISO 8601 [20]) и event (описание действия). Таким образом, тестовые данные включали визуальный контекст, числовые параметры и текстовое событие для каждой ситуации.

Проведение эксперимента. Каждый из подготовленных сценариев подавался на вход всем выбранным моделям по очереди. Для моделей, способных принимать изображение (llava:13b, llama3.2-vision:11b), в запрос напрямую включался сгенерированный кадр и сопутствующий текст (показания датчиков, лог). Для текстовых моделей (gemma3:12b,

minicpm-v:8b) вместо изображения в промпт добавлялось описание сцены (сформированное вручную или отдельным модельным генератором на основе содержимого кадра). Структура промпта оставалась единой: сначала представлялась информация о текущей обстановке (например, «На камере видно: человек лежит на полу; Датчики: temperature = 85, humidity = 10; Событие: Дверь открыта сотрудником №1234»), затем задавался вопрос модели о наличии отклонений или необходимости тревоги. Такой формат взаимодействия выбран потому, что он максимально использует возможности LLM по пониманию описательного текста и установлению связей между разнородными фактами [24]. Все запросы и ответы автоматизированно отправлялись и получались с помощью скрипта на Python через API Ollama [25].

Для оценки результатов вручную была проведена разметка эталонного ответа для каждого сценария: отмечалось, является ли ситуация нормальной или аномальной, и какой именно инцидент происходит (если есть). Например, для сценария падения эталон: «аномалия (падение человека)», для пустого коридора: «норма». Ответы моделей приводились к упрощённому виду (модель либо сигнализирует об аномалии, либо считает ситуацию нормальной исходя из своего описания). На этой основе для каждой модели вычислялись показатели точности.

6. Результаты и обсуждение

По итогам эксперимента для каждой модели оценивалось качество обнаружения аномалий. В качестве основной метрики была выбрана F1-мера, сочетающая полноту и точность обнаружения аномального класса [26]. Расчёт выполнялся с помощью стандартной функции из библиотеки Scikit-learn [27]. В табл. 1 представлены итоговые значения F1-Score для каждой модели по совокупности всех сценариев.

Таблица 1. Модели и их F1-Score

Модель	F1-Score
llama3.2-vision:11b	0.5714285714285714
gemma3:12b	0.8888888888888888
llava:13b	0.0
minicpm-v:8b	0.8

Как видно, результаты заметно разнятся между моделями. Лучший показатель продемонстрировала модель gemma3:12b (F1 ≈ 0,89), правильно распознавшая аномальные ситуации почти во всех случаях. Чуть хуже выступила облегчённая модель minicpm-v:8b (F1 = 0,80), уступившая из-за нескольких ошибок. Модель llama3.2-vision:11b верно определила лишь около половины аномалий (F1 ≈ 0,57), а мультимодальная модель LLaVA (13b) не справилась с заданием (F1 = 0), фактически не обнаружив ни одного отклонения. Вероятно, такие различия связаны с тем, как данные модели обучены и настроены: gemma3, хоть и не имела прямого «зрения», сумела корректно интерпретировать описания, тогда как LLaVA, несмотря на способность анализировать изображения, могла неправильно понять контекст сценариев или не была достаточно обучена на подобных ситуациях. Возможной причиной неудачи LLaVA является то, что её обучение на паре «изображение – текст» могло не включать специфичные сцены видеонаблюдения, из-за чего модель не распознала признаки аномалий (например, падение человека) на уровне здравого смысла.

Помимо точности распознавания, измерялось время отклика каждой модели на каждый сценарий. В табл. 2 приведены задержки генерации ответа (в секундах) по всем шести сценариям, а также суммарное время на обработку всего набора.

Таблица 2. Время ответа моделей на каждый сценарий

Модель	Сценарий 1	Сценарий 2	Сценарий 3	Сценарий 4	Сценарий 5	Сценарий 6	Общее время
llama3.2-vision:11b	3,36 с	3,12 с	3,06 с	2,57 с	22,82 с	2,89 с	37,83 с
gemma3:12b	14,65 с	11,15 с	10,73 с	10,26 с	10,06 с	9,86 с	66,70 с
llava:13b	12,83 с	7,46 с	4,64 с	5,06 с	5,82 с	4,60 с	40,40 с
minicpm-v:8b	12,62 с	10,59 с	10,95 с	10,31 с	9,83 с	1,88 с	56,18 с

Общее время обработки всех сценариев варьируется от ~38 секунд (для llama3.2-vision) до ~67 секунд (для gemma3), что ожидаемо отражает как различия в объёме модели (большее число параметров требует больше вычислений), так и особенности реализации. Так, модель gemma3:12b оказалась самой медленной, хотя и наиболее точной: её суммарное время ~66,7 секунд — почти вдвое больше, чем у сравнимой по размеру LLaVA. Возможно, gemma3 менее оптимизирована для быстрого вывода в выбранной среде или генерирует более развёрнутые ответы. Наоборот, модель llama3.2-vision:11b показала лучшую скорость, выдав результаты по всем сценариям примерно за 38 секунд. Однако стоит отметить, что в её случае наблюдалась наибольшая дисперсия времени между разными сценариями: если большинство изображений обрабатывались за 2–3 секунды, то сценарий 5 (пустой коридор) занял у неё почти 23 секунды. Вероятно, на пустой сцене модель сгенерировала более длинное рассуждение, пытаясь описать или осмыслить обстановку без очевидных объектов, что увеличило время вывода. У модели minicpm-v:8b, напротив, самый быстрый отклик был на сценарии 6 (всего ~1,9 секунд), поскольку этот сценарий не содержал изображение и представлял собой только небольшой текстовый фрагмент (числовые показатели), с обработкой которого компактная модель справилась очень быстро. В других же случаях minicpm-v уступала по скорости более крупным моделям, вероятно, из-за менее эффективной архитектуры или отсутствия специализированных оптимизаций для работы с изображениями. Таким образом, разные модели демонстрируют различное поведение во времени: большие модели способны быть относительно быстрыми благодаря оптимизациям, а меньшие могут проигрывать в скорости, если задача выходит за пределы их узкой специализации.

В целом эксперимент подтверждает, что применение больших языковых моделей для анализа мультимодальных данных является перспективным, однако выбор конкретной модели сильно влияет на качество и скорость. Без дополнительной адаптации предобученные модели показывают неоднозначные результаты: одни (как gemma3) практически «из коробки» справляются с выявлением нештатных ситуаций, в то время как другие (как LLaVA) требуют, возможно, дообучения или более продуманного промпта, чтобы давать полезные выводы. Разброс метрик качества от 0 до ~0,9 указывает на необходимость тщательного подхода к выбору и настройке модели под конкретную задачу. Аналогично, колебания времени работы — от единиц секунд до десятков — показывают, что внедрение таких систем на практике потребует учёта требований к быстродействию и, возможно, компромисса между точностью и скоростью.

7. Заключение

Разработанный прототип интеллектуальной системы мониторинга продемонстрировал принципиальную возможность использования больших нейросетевых моделей для одновременного анализа разнородных данных (видео, сенсоры, логи) в задаче отслеживания активности. Показано, что даже без специального обучения на целевой выборке некоторые предобученные модели способны обнаруживать аномалии, объединяя информацию из описаний различных модальностей. Это подтверждает актуальность выбранного направления: современные LLM могут служить своеобразным «мозгом» для систем наблюдения, обобщающим разноплановые сигналы и помогающим оператору быстрее получать инсайты о происходящем. Практическая ценность такого подхода заключается в снижении необходимости ручного мониторинга и разработки множества узкоспециализированных детекторов — единая модель или небольшое их множество может выполнять сразу несколько задач анализа. Кроме того, использование мультимодальных больших моделей позволяет учитывать контекст: например, сопоставлять событие в логе с картиной на видео и показаниями сенсора, что повышает надёжность выявления комплексных инцидентов.

Научная значимость работы в том, что она заполняет пробел между разрозненными направлениями (видеоаналитика, анализ сенсорных данных, обработка текстовых логов) посредством единого подхода на базе LLM. Проведённое исследование выявило как преимущества, так и текущие ограничения такого подхода. К преимуществам относится универсальность и гибкость больших моделей — они могут интерпретировать нестандартизированные описания и делать выводы, близкие к человеческим. Однако обнаружены и проблемы: разные модели демонстрируют очень неоднородные результаты, что указывает на необходимость их адаптации под предметную область. В будущем планируется расширить набор сценариев и данных, а также исследовать возможность дообучения (fine-tuning) моделей на специальных мультимодальных датасетах, чтобы улучшить их понимание специфических ситуаций (например, сцен с падениями или другими инцидентами). Также представляет интерес оптимизация производительности: сокращение времени отклика, использование модельных ансамблей или иерархий (например, быстрая фильтрация событий компактной моделью с последующим уточнением большой моделью). В заключение необходимо отметить, что интеграция больших нейросетевых моделей в системы анализа активности является перспективным направлением, способным повысить интеллектуальность и автономность средств мониторинга в различных прикладных областях — от промышленной безопасности до умных домов и кибербезопасности.

Весь код скриптов, изображения, а также результаты моделей опубликованы на GitHub: https://github.com/minneakhmetov/llm-activity.

Литература

[1] Ferrara E. Large Language Models for Wearable Sensor-Based Human Activity Recognition, Health Monitoring, and Behavioral Modeling // Sensors. 2024. Vol. 24, No. 15. P. 5045.
[2] OpenAI ChatGPT-4o-mini. URL: https://chatgpt.com/ (дата обращения: 30.03.2025).
[3] Пятаева А.В., Мерко М.А., Жуковская В.А., Казакевич А.А. Распознавание активности человека по видеоданным // International Journal of Advanced Studies. 2022. Т. 12. № 4. С. 96-110.
[4] Sharma R., Patel N. Deep learning-based anomaly detection in surveillance videos // Journal of Visual Communication and Image Representation. 2022. Vol. 86. 103624.
[5] Котенко И. В., Полубелова О. В., Саенко И. Б., Чечулин А. А. Применение онтологий и логического вывода для управления информацией и событиями безопасности // Системы высокой доступности. 2012. Т. 8. № 2. С. 100-108.

[6] Nour B., Pourzandi M., Debbabi M. A Survey on Threat Hunting in Enterprise Networks // IEEE Communications Surveys & Tutorials. 2023. T. 25. C. 2299-2324. DOI: 10.1109/COMST.2023.3299519.
[7] Suh S., Rey V.F., Lukowicz P. Tasked: Transformer-based adversarial learning for human activity recognition using wearable sensors // Knowledge-Based Systems. 2023. Vol. 260. 110143.
[8] Gupta S. Deep learning-based human activity recognition using wearable sensor data // Int. J. Inf. Manag. Data Insights. 2021. Vol. 1. 100046.
[9] Nath N. D., Behzadan A. H., Paal S. G. Deep learning for site safety: Real-time detection of personal protective equipment // Automation in Construction. 2020. Vol. 112. 103085.
[10] Han S., Yuan, S., Trabelsi M. LogGPT: Log Anomaly Detection via GPT // arXiv. 2023. DOI: 10.48550/arXiv.2309.14482.
[11] Ollama gemma3:12b Model. URL: https://ollama.com/library/gemma3:12b (дата обращения: 30.03.2025).
[12] Ollama llava:13b Model. URL: https://ollama.com/library/llava:13b (дата обращения: 30.03.2025).
[13] Ollama lllama3.2-vision:11b Model. URL: https://ollama.com/library/llama3.2-vision (дата обращения: 30.03.2025).
[14] Ollama minicpm-v:8b Model. URL: https://ollama.com/library/minicpm-v (дата обращения: 30.03.2025).
[15] Uçar A., Karakoşe M., Kırımça N. Artificial Intelligence for Predictive Maintenance Applications: Key Components, Trustworthiness, and Future Trends // Applied Sciences. 2024. Vol.14. No. 2. 898.
[16] Özüağ S., Ertuğrul Ö. Enhanced Occupational Safety in Agricultural Machinery Factories: Artificial Intelligence-Driven Helmet Detection Using Transfer Learning and Majority Voting // Applied Sciences. 2014. Vol. 14. No. 23. 11278. DOI: 10.3390/app142311278.
[17] Li X., Chen Y., Hu L. Real-time workplace activity recognition using deep learning models // IEEE Transactions on Industrial Informatics. 2023. Vol.19. No. 2. P. 1520–1532.
[18] Wu Z., Zhao J., Shen H. Smart home automation based on human activity recognition: A survey // Future Generation Computer Systems. 2023. Vol. 137. P. 41–57.
[19] Yadav S., Jha C.K., Kumar N. AI-powered fall detection systems for elderly care: Challenges and future directions // Computer Methods and Programs in Biomedicine. 2024. Vol. 230. 107416.
[20] ISO 8601-1:2019 Standard. URL: https://www.iso.org/obp/ui/#iso:std:iso:8601:-1:ed-1:v1:en (дата обращения: 30.03.2025).
[21] Ollama API Documentation. URL: https://github.com/ollama/ollama/blob/main/docs/api.md (дата обращения: 30.03.2025).
[22] Ollama. URL: https://ollama.com/ (дата обращения: 30.03.2025).
[23] Ollama Python Library. URL: https://github.com/ollama/ollama-python (дата обращения: 30.03.2025).
[24] Hand D.J., Christen P. F: an interpretable transformation of the F-measure // Journal of Classification. 2021. Vol. 38. No. 1. P. 3–17.
[25] Sahoo P., Singh A. K., Saha S., Jain V., Mondal S., Chadha A. A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications // arXiv. 2024. DOI: 10.48550/arXiv.2402.07927.
[26] Scikit Learn F1-Score. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html (дата обращения: 30.03.2025).

Intelligent Activity Tracking System Based on Large Neural Network Models

R. Minneakhmetov

Kazan Federal University

razil0071999@gmail.com

An approach to constructing an intelligent activity tracking system using large neural network models is presented. The main focus is on the use of language models and computer vision models for analyzing video surveillance data, sensor signals, and logs. The local Ollama framework is used as an implementation tool, ensuring safe and autonomous operation of large language models. A prototype of the system is proposed, its architecture, data processing methodology, and experimental part are described. It is shown that the use of large models allows automating the analysis process and increasing the accuracy of anomaly detection.

Keywords: Ollama, Large Language Models, activity tracking, video analytics, artificial intelligence

Reference for citation: R. Minneakhmetov Intelligent Activity Tracking System Based on Large Neural Network Models // Information Society: Education, Science, Culture and Technology of Future. Vol. 9 (Proceedings of the XXVIII International Joint Scientific Conference «Internet and Modern Society», IMS-2025, St. Petersburg, June 23–25, 2025). – St. Petersburg: ITMO University, 2025. P. 127-137. DOI: 10.17586/3033-5574-2025-9-127-137.

Reference

[1] Ferrara E. Large language models for wearable sensor-based human activity recognition, health monitoring, and behavioral modeling // Sensors. 2024. Vol. 24, No. 15. P. 5045.
[2] OpenAI ChatGPT-4o-mini. URL: https://chatgpt.com/ (accessed date: 30.03.2025).
[3] Pyataeva A. V., Merko M. A., Zhukovskaya V. A., Kazakevich A. A. Raspoznavanie aktivnosti cheloveka po videodannym // International Journal of Advanced Studies. 2022. Vol. 12, No. 4. P. 96-110. (In Russian)
[4] Sharma R., Patel N. Deep learning-based anomaly detection in surveillance videos // Journal of Visual Communication and Image Representation. 2022. Vol. 86. 103624.
[5] Kotenko I. V., Polubelova O. V., Saenko I. B., Chechulin A. A. Primenenie ontologij i logicheskogo vyvoda dlja upravlenija informaciej i sobytijami bezopasnosti // Sistemy vysokoj dostupnosti. 2012. Vol. 8, No. 2. P. 100-108. (In Russian)
[6] Nour B., Pourzandi M., Debbabi M. A survey on threat hunting in enterprise networks // IEEE Communications Surveys & Tutorials. 2023. Vol. 25. P. 2299-2324. DOI: 10.1109/COMST.2023.3299519.
[7] Suh S., Rey V. F., Lukowicz P. Tasked: Transformer-based adversarial learning for human activity recognition using wearable sensors // Knowledge-Based Systems. 2023. Vol. 260. 110143.
[8] Gupta S. Deep learning-based human activity recognition using wearable sensor data // International Journal of Information Management Data Insights. 2021. Vol. 1. 100046.
[9] Nath N. D., Behzadan A. H., Paal S. G. Deep learning for site safety: Real-time detection of personal protective equipment // Automation in Construction. 2020. Vol. 112. 103085.
[10] Han S., Yuan S., Trabelsi M. LogGPT: Log anomaly detection via GPT // arXiv. 2023. DOI: 10.48550/arXiv.2309.14482.
[11] Ollama gemma3:12b Model. URL: https://ollama.com/library/gemma3:12b (accessed date: 30.03.2025).

[12]Ollama llava:13b Model. URL: https://ollama.com/library/llava:13b (accessed date: 30.03.2025).
[13]Ollama llama3.2-vision:11b Model. URL: https://ollama.com/library/llama3.2-vision (accessed date: 30.03.2025).
[14]Ollama minicpm-v:8b Model. URL: https://ollama.com/library/minicpm-v (accessed date: 30.03.2025).
[15]Uçar A., Karakoşe M., Kırımça N. Artificial intelligence for predictive maintenance applications: Key components, trustworthiness, and future trends // Applied Sciences. 2024. Vol. 14, No. 2. 898.
[16]Özüağ S., Ertuğrul Ö. Enhanced occupational safety in agricultural machinery factories: Artificial intelligence-driven helmet detection using transfer learning and majority voting // Applied Sciences. 2024. Vol. 14, No. 23. 11278. DOI: 10.3390/app142311278.
[17]Li X., Chen Y., Hu L. Real-time workplace activity recognition using deep learning models // IEEE Transactions on Industrial Informatics. 2023. Vol. 19, No. 2. P. 1520-1532.
[18]Wu Z., Zhao J., Shen H. Smart home automation based on human activity recognition: A survey // Future Generation Computer Systems. 2023. Vol. 137. P. 41-57.
[19]Yadav S., Jha C. K., Kumar N. AI-powered fall detection systems for elderly care: Challenges and future directions // Computer Methods and Programs in Biomedicine. 2024. Vol. 230. 107416.
[20]ISO 8601-1:2019 Standard. URL: https://www.iso.org/obp/ui/#iso:std:iso:8601:-1:ed-1:v1:en (accessed date: 30.03.2025).
[21]Ollama API Documentation. URL: https://github.com/ollama/ollama/blob/main/docs/api.md (accessed date: 30.03.2025).
[22]Ollama. URL: https://ollama.com/ (accessed date: 30.03.2025).
[23]Ollama Python Library. URL: https://github.com/ollama/ollama-python (accessed date: 30.03.2025).
[24]Hand D.J., Christen P. F: An interpretable transformation of the F-measure // Journal of Classification. 2021. Vol. 38, No. 1. P. 3-17.
[25]Sahoo P., Singh A.K., Saha S., Jain V., Mondal S., Chadha A. A systematic survey of prompt engineering in large language models: Techniques and applications // arXiv. 2024. DOI: 10.48550/arXiv.2402.07927.
[26] Scikit Learn F1-Score. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html (accessed date: 30.03.2025).