ИЗВЕСТИЯ РАН. ТЕОРИЯ И СИСТЕМЫ УПРАВЛЕНИЯ, 2020, № 5, с. 73–88

РАСПОЗНАВАНИЕ ОБРАЗОВ И ОБРАБОТКА ИЗОБРАЖЕНИЙ

УДК 004.932

ВЫЧИСЛИТЕЛЬНЫЙ МЕТОД РАСПОЗНАВАНИЯ СИТУАЦИЙ И ОБЪЕКТОВ В КАДРАХ НЕПРЕРЫВНОГО ВИДЕОПОТОКА С ИСПОЛЬЗОВАНИЕМ ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ СИСТЕМ КОНТРОЛЯ И УПРАВЛЕНИЯ ДОСТУПОМ[1]

© 2020 г. О. С. Амосов[a],*, С. Г. Амосова[a], С. В. Жиганов[b], Ю. С. Иванов[b], Ф. Ф. Пащенко[a]

[a] *ФГБУН ИПУ им. В.А. Трапезникова РАН, Москва, Россия*

[b] *ФГБОУ ВО Комсомольский-на-Амуре государственный ун-т, Комсомольск-на-Амуре, Россия*

**e-mail: osa18@yandex.ru*

Поступила в редакцию 13.01.2019 г.
После доработки 01.05.2020 г.
Принята к публикации 25.05.2020 г.

Предлагается эффективный, с точки зрения точности и быстродействия, вычислительный метод распознавания образов в непрерывном видеопотоке с использованием глубоких нейронных сетей для систем контроля и управления доступом. Выделен класс решаемых методом задач распознавания: как самого автомобиля, так и символов его номерного знака; лиц людей; нештатных ситуаций с помощью последовательности кадров видеопотока. В отличие от известных решений применяется классификация с последующим подкреплением на основе нескольких кадров видеопотока и алгоритмом автоматического аннотирования изображений. Предложены адаптированные для решаемых задач архитектуры нейронных сетей с независимыми рекуррентными слоями для классификации видеофрагментов, дуальная сеть для распознавания лиц, глубокая нейронная сеть для распознавания символов транспортного средства. Созданы оригинальные базы данных для обучения нейронных сетей. Предложена схема интеллектуальной системы контроля и управления доступом для обеспечения безопасности предприятия, отличительной особенностью которой является использование мультироторного беспилотного летательного аппарата с вычислительным модулем. Проведены натурные эксперименты и оценены характеристики точности и скорости вычислительного метода при решении каждой задачи. Разработаны программные модули на языке Python для решения задач интеллектуальной системы контроля и управления доступом.

DOI: 10.31857/S0002338820050029

Введение. В настоящее время бурный рост переживает внедрение систем компьютерного зрения (СКЗ) в различных предметных областях. На кадрах видеопотока могут фиксироваться образы физических и технических объектов, а также ситуации, происходящие с их участием. Для каждого объекта или ситуации набор свойств различный. Примеры свойств, которыми обладают технические объекты – автомобили: тип, номер, цвет и т.д.; физические объекты – люди: пол, возраст и т.д. Для объектов их свойства можно оценить по отдельным кадрам, даже по одному из них. Существенным отличием от объектов для ситуаций характерны протяженность во времени и взаимосвязи между динамическими объектами. Поэтому для оценки свойств ситуаций необходимо использовать именно последовательность кадров непрерывного видеопотока.

Основной задачей алгоритмов компьютерного зрения является поиск образов на изображении и выделение их ключевых признаков, характеризующих свойства объектов и ситуации, их распознавание для последующего принятия решений или управления.

[1] Работа выполнена при поддержке Минобрнауки России научного проекта – госзадания в рамках проектной части № 2.1898.2017/ПЧ “Создание математического и алгоритмического обеспечения интеллектуальной информационно-телекоммуникационной системы безопасности вуза”.

В последнее время одной из заметных тенденций является применение в СКЗ глубоких нейронных сетей (НС) при решении задач распознавания образов. Значительные результаты были получены с помощью сверточных и рекуррентных НС.

Для получения свойств технического объекта (автомобиля) авторами [1] предлагается использовать сложную структуру на базе сверточной НС, особенностью которой является возможность обнаружения номерных знаков в сложных сценах (различная ориентация изображения, шумы и т.д.). Однако архитектура Mask-RCNN [1] достаточно требовательна к ресурсам при работе в режиме реального времени. В [2] предлагается использовать глубокие нейросети (англ. deep neural network) для детекции номерного знака (НЗ), локализации символов и их распознавания даже при условии различных геометрических искажений номерной пластины. Недостатком подхода является обязательное выполнение предварительной калибровки камеры для каждого случая детектирования. Авторами [3] предлагается метод распознавания цветов транспортного средства на базе глубоких архитектур НС на смазанных и затуманенных изображениях. Несмотря на высокие результаты данного подхода, цвет автомобиля не является исчерпывающим идентификационным признаком и может использоваться только в качестве подкрепления.

Подобные нейросетевые решения применялись и для физических объектов. В [4] для идентификации человека выполняется классификация вектора признаков методом опорных векторов. Однако для решения задачи локализации применяется вычислительно затратный перевод изображения в гистограмму направленных градиентов (англ. histogram of oriented gradients, HOG). В [5] предлагается модифицированная модель глубокой НС MobileNet для верификации лиц. Несмотря на высокую точность – 99.55% на базе LFW (англ. labeled faces in the wild) [6], данный подход не позволяет изменять количество классов во время работы. В [7] достигнута высокая точность распознавания на общедоступных тестах благодаря триплетной функции потерь [8] при обучении. Однако вычислительные затраты не позволяют использовать данный подход в режиме реального времени (РВ).

Для распознавания ситуаций, происходящих с участием объектов, необходимо анализировать последовательность кадров. Авторами [9] предлагается обнаруживать нештатные ситуации, анализируя характеристики оптического потока в переполненных людьми сценах, что влечет ограничение на количество распознаваемых ситуаций. В [10] представлена система, которая может обнаруживать аномальное поведение людей на видео с помощью глубоких архитектур НС. Несмотря на высокую точность – 85%, используемые авторами слои долгой краткосрочной памяти (англ. long short-term memory, LSTM) в архитектуре НС требуют огромных вычислительных ресурсов и с трудом анализируют видео в РВ.

При всем разнообразии различных подходов не существует универсального вычислительного метода, отражающего все этапы решения задачи распознавания объектов и ситуаций в единой системе контроля и управления доступом объектов для обеспечения безопасности предприятия.

Поэтому в статье ставится цель разработать вычислительный метод распознавания образов в непрерывном видеопотоке с помощью глубоких НС. В разд. 1 постановки и решения задачи приведена математическая формулировка задачи распознавания образов, а также предлагается вычислительный метод распознавания образов. В разд. 2 приведены иллюстрирующие примеры решения следующих задач с использованием предложенного вычислительного метода: распознавание нештатных ситуаций, символов НЗ и лиц людей. Продемонстрирована реализация на языке Python и проведены вычислительные эксперименты на различных конфигурациях оборудования. В заключение приводятся основные научные результаты статьи.

1. Постановка и решение задачи. Под образами будем понимать технические, физические объекты, штатные и нештатные ситуации, происходящие при взаимодействии объектов.

По поступающему видеопотоку необходимо обнаружить образы объектов или ситуаций, выделив ключевые признаки, и отнести их к одному из классов. Для реализации распознавания образов нужно разработать вычислительный метод на базе сверточных и рекуррентных НС и продемонстрировать возможность его применения в разных предметных областях, создать программное обеспечение на языке программирования Python и провести натурный эксперимент с видеопотоком, поступающим с камер наблюдения системы контроля и управления доступом (СКУД) в университете ФГБОУ ВО “КнАГУ”.

1.1. Математическая формулировка задачи распознавания образов. Пусть имеются: множество образов $\omega \in \mathbf{\Omega}$, заданных признаками x_i, $i = \overline{1,n}$, совокупность которых для образа ω представлена векторными описаниями $\mathbf{\Phi}(\omega) = (x_1(\omega), x_2(\omega), \ldots, x_n(\omega)) = \mathbf{x}$; множество классов $\mathbf{B} = \{\beta_1, \ldots, \beta_k, \ldots, \beta_c\}$, c – количество классов. Априорная информация представ-

лена обучающим множеством (датасетом) $\boldsymbol{D} = \{(\mathbf{x}^j, \beta^j)\}, j = \overline{1, L}$, заданным таблицей, каждая строка j которой содержит векторное описание образа $\boldsymbol{\Phi}(\omega)$ и метку класса β_k, $k = \overline{1, c}$. Заметим, что обучающее множество характеризует неизвестное отображение $^*\mathbf{F}$: $\boldsymbol{\Omega} \to \mathbf{B}$.

Требуется по имеющимся кадрам $\mathbf{I}_t$ непрерывного видеопотока $\mathbf{V} = (\mathbf{I}_1.., \mathbf{I}_t, .., \mathbf{I}_\tau)$ и априорной информации, заданной обучающим множеством $\mathbf{D} = \{(\mathbf{x}^j, \beta^j)\}$, $j = \overline{1, L}$, для глубокого обучения НС с учителем, решить задачу распознавания образов: обнаружить образы ω в виде оценки признаков $\tilde{\mathbf{x}}$ с помощью НС, реализующих отображение [11] $\mathbf{F}_1$: $\mathbf{I}_t \to \tilde{\mathbf{x}}$, и классифицировать их с использованием отображения $\mathbf{F}_2$: $\tilde{\mathbf{x}} \to \beta_k$, $k = \overline{1, c}$, в соответствии с заданным критерием $\mathrm{P}(\tilde{\mathbf{x}})$, минимизирующим вероятность ошибки классификации.

Таким образом, необходимо найти отображение $\mathbf{F}$: $\mathbf{I}_t \to \beta_k$, $k = \overline{1, c}$, при котором $\mathbf{F}$ является набором функций и алгоритмов $\mathbf{f}_i$, $i = \overline{1, N_f}$.

1.2. Решение задачи обнаружения и классификации объектов и ситуаций. Для решения задачи обнаружения и классификации объектов и ситуаций предлагается вычислительный метод распознавания образов, выполняющий отображение $\mathbf{F}$: $\mathbf{I}_t \to \beta_k$, $t = \overline{1, \tau}$, $k = \overline{1, c}$, с его реализацией на основе композиции традиционных методов обработки изображений и глубоких НС.

1. Выделение из непрерывного видеопотока $\mathbf{V} = (\mathbf{I}_1.., \mathbf{I}_t, .., \mathbf{I}_\tau)$ кадра $\mathbf{I}_t$ размером $w^{\mathbf{I}_t} \times h^{\mathbf{I}_t}$, где t – номер текущего кадра.

2. Поиск образов объектов на кадре $\mathbf{f}_1$: $\mathbf{I}_t \to \mathbf{G}_t$, где $\mathbf{G}_t$ – массив элементов, содержащий параметры n объектов в кадре видеопотока. При наличии искомого объекта o выполняется переход к следующему этапу, в противном случае берется следующий кадр.

3. Выделение области интереса первого уровня $\mathbf{R}^{\langle 1 \rangle} = crop(\mathbf{I}_t, x^o, y^o, w^o, h^o)$, где x^o, y^o – координаты центра o-го объекта, w^o, h^o – его размеры, $crop$ – операция вырезания из $\mathbf{I}_t$ подматрицы по координатам $(x^o - w^o/2, y^o - h^o/2)$, $(x^o + w^o/2, y^o + h^o/2)$.

4. Уточнение области интереса для детализации информации об образе $\mathbf{f}_2$: $(\mathbf{R}^{\langle 1 \rangle}, t) \to \mathbf{R}^{\langle 2 \rangle}$.

5. Выполнение предобработки области интереса $\mathbf{R}^{\langle 2* \rangle} = \mathbf{f}_3(\mathbf{R}^{\langle 2 \rangle}, \mathbf{M}, \mathbf{g})$, где $\mathbf{M}$ – матрица геометрических линейных и аффинных преобразований $\mathbf{R}^{\langle 2 \rangle}$, $\mathbf{g}$ – набор матричных функций и их параметров для яркостных и контрастных преобразований $\mathbf{R}^{\langle 2 \rangle}$.

6. Выделение информативных признаков $\mathbf{R}^{\langle 2* \rangle}$ путем извлечения признаков из заданного слоя предобученной сверточной НС (англ. convolutional neural network, CNN): $\boldsymbol{\Phi}^{CNN}$: $\mathbf{R}^{\langle 2* \rangle} \to \tilde{\mathbf{x}}$, где $\tilde{\mathbf{x}}$ – область интереса, переведенная в признаковое пространство CNN (оценка карты признаков). Стоит отметить, что могут использоваться архитектуры сверточных НС, предобученных как на датасете ImageNet [12], так и на подготовленных датасетах $\mathbf{D}$.

7. Отнесение вектора признаков к одному из классов $\mathbf{f}_4$: $\tilde{\mathbf{x}} \to \mathbf{p}_{\tilde{\mathbf{x}}}$, где $\mathbf{p}_{\tilde{\mathbf{x}}}$ – вектор размером $c \times 1$, содержащий вероятности классификации, c – количество классов.

Для усиления классификации используются предложенные нами алгоритмы подкрепления на базе нечеткой логики [11]. Учитывая особенность предметной области, нами вводятся нечеткие функции доверия, зависящие от нескольких кадров и размеров объекта.

8. Критерий классификации определяется как $J(\mathbf{f}_4) = \max \mathbf{p}_{\tilde{\mathbf{x}}}$. Если $J(\mathbf{f}_4) \geq \varepsilon$, где ε – заданный порог, то $\beta_k = \arg\max\limits_{k \in 1..c}(\mathbf{p}_{\tilde{\mathbf{x}}_k})$, в противном случае классификация считается ошибочной.

Характеристика точности метода складывается из характеристик используемых алгоритмов и вычисляется с помощью традиционных метрик [13].

Результирующая точность и полнота метода рассчитывается как арифметическое среднее его метрик точности и полноты по всем классам.

Дополнительной характеристикой метода является быстродействие.

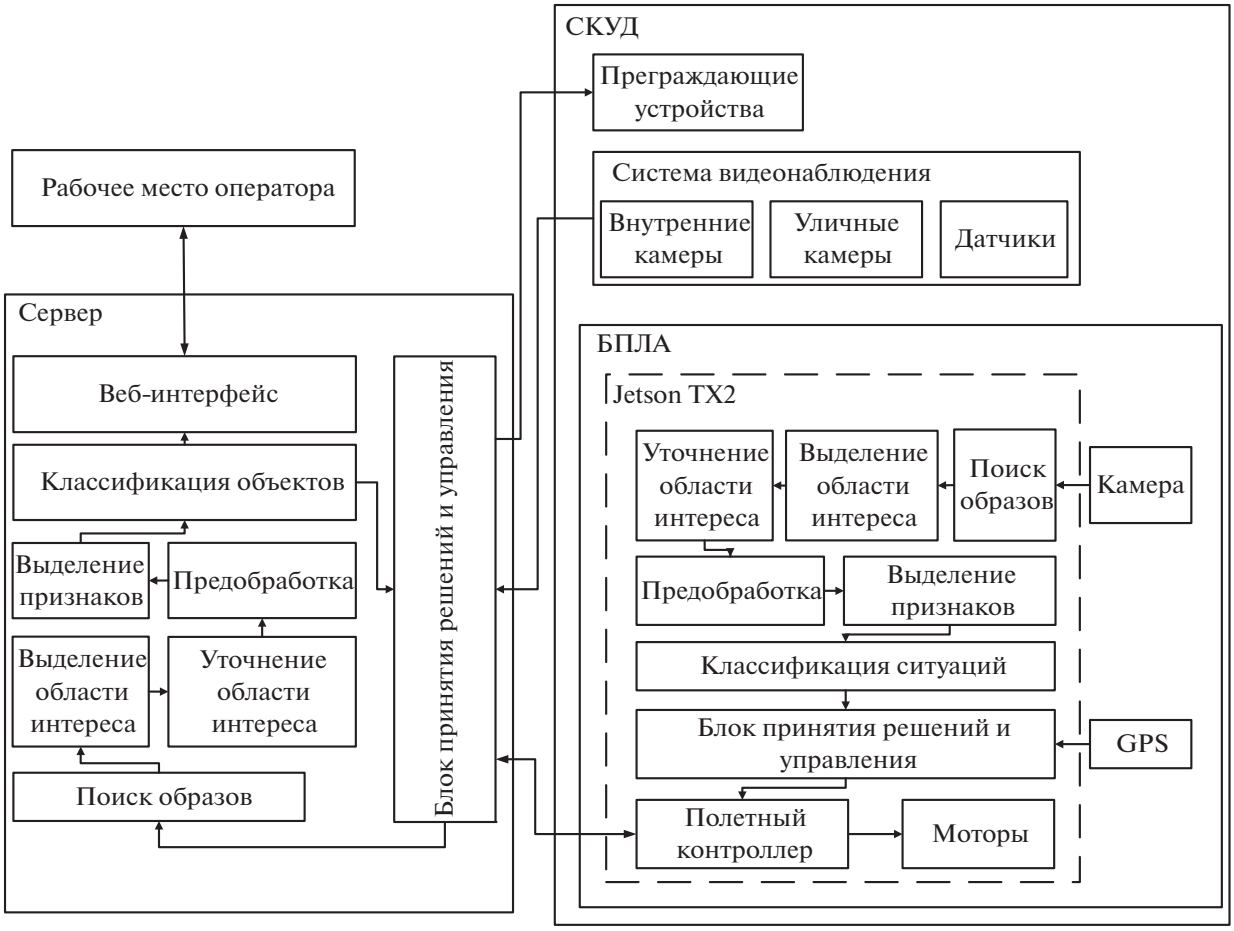

Рис. 1. Структурная схема СКУД

2. Иллюстрирующие примеры. Предлагаемый вычислительный метод может быть применен для распознавания разнотипных образов из разных предметных областей. Однако наиболее показательно и актуально применение в задачах комплексной безопасности предприятия.

Методы компьютерного зрения в таких системах способны ограничивать доступ физических лиц и технических объектов. Однако одной из важнейших нерешенных задач остается раннее обнаружение нештатных ситуаций для принятия решения и выработки управляющих воздействий.

На рис. 1 приведена схема интеллектуальной СКУД для обеспечения безопасности предприятия, содержащая множество внутренних и наружных камер, датчиков и управляемых устройств. Отличительной особенностью предлагаемой СКУД является использование мультироторного беспилотного летательного аппарата (БПЛА) Matrice 100 с вычислительным модулем Jetson TX2 на борту, патрулирующего охраняемую территорию.

Задачами приведенной системы является распознавание: символов НЗ транспортного средства (ТС) [14], лиц людей [15], штатных и нештатных ситуаций [16].

Алгоритмы для иллюстрирующих примеров разработаны согласно этапам предложенного вычислительного метода. Все алгоритмы были реализованы на языке Python с использованием библиотек Tenserflow и Keras. Вычислительные эксперименты выполняются на различных конфигурациях оборудования, имеющих следующие типы центрального (ЦПУ) и графического (ГПУ) процессорных устройств (табл. 1).

В данной статье более подробно рассмотрим распознавание штатных и нештатных ситуаций как наиболее сложную задачу.

2.1. Распознавание штатных и нештатных ситуаций. При решении задачи распознавания штатных и нештатных ситуаций использовалось оборудование:

– внутренняя и внешняя камеры, установленные на территории;

Таблица 1. Конфигурации компьютеров для оценки скорости алгоритмов

Конфигурация	1	2	3	4
ЦПУ	Intel Core i3-7100	Intel Core i5-7400	Intel Core i5-4690	Intel Core i7-5820K
ГПУ	GeForce 1030	GeForce 1050	GeForce 1050 Ti	GeForce 1080 Ti

– сервер, выполняющий обработку видеопотока, распознавание образов и вырабатывающий управляющие воздействия на основе правил нечеткой логики [15].

Также предлагается вариант мобильной камеры, установленной на мультироторном БПЛА Matrice 100, тогда обработка возможна на борту летательного аппарата с использованием встраиваемого вычислительного графического модуля Jetson XT2.

Рассмотрим особенности этапов вычислительного метода применительно к задаче распознавания штатных и нештатных ситуаций.

1. Выделение из непрерывного видеопотока кадра.

2. Поиск объектов на кадре.

В качестве алгоритма поиска объекта предлагается для сегментации изображения использовать предобученную глубокую НС, имеющую архитектуру как у модели НС SegNet [17]. Тогда $\mathbf{f}_1^{SegNet}: \mathbf{I}_t \to \mathbf{G}_t$. Результатом сегментации является разметка изображения на области определенного класса, т.е. мы получаем массив $\mathbf{G}_t$, содержащий области класса и их координаты. Если $\tilde{o} = \{1, 2, ..., 5\}$, т.е. детектированы объекты “человек”, “автомобиль”, “дым”, “огонь”, “вспышки”, то происходит переход к следующему этапу, иначе выполняется первый этап метода для получения следующего кадра.

Таким образом данный пункт служит триггером перехода к следующему шагу и началом отчета записи видеофрагмента для дальнейшего анализа. Без использования данного пункта необходимо постоянно выполнять классификации видеофрагментов, что приведет к повышению нагрузки на вычислительный сервер. При этом в качестве детектора также может применяться архитектура YOLO.

3. Выделение области интереса первого уровня.

При обнаружении ситуаций областью интереса выступает изображение в целом, тогда $x^o = i/2$, $y^o = j/2$, $w^o = i$, $h^o = j$, т.е. $\mathbf{R}^{\langle 1\rangle} = \mathbf{I}_t$.

4. Уточнение области интереса для детализации информации об образе.

Особенностью ситуаций является протяженность во времени, а значит, необходимо выполнить анализ последовательности кадров и обобщить полученную информацию за определенный временной интервал, тогда $\mathbf{R}^{\langle 2\rangle} = concat\left(\mathbf{I}_{t-i+1}, \mathbf{I}_{t-i+2}, \ldots, \mathbf{I}_t\right)$, где i – количество кадров, $concat$ – операция конкатенации нескольких подряд идущих кадров в многомерный массив.

Данная задача решается последовательным прохождением по видеопотоку $\mathbf{V}$ сканирующего окна размером $i = 690$ кадров и шагом смещения $d = 23$ кадра (1 с). Кадры, захваченные окном, создают массив $\mathbf{R}_k^{\langle 2\rangle}$, где k – номер окна (рис. 2). Для анализа нами формировалось 10 подряд идущих окон, каждое из которых классифицировалось отдельно.

5. Выполнение предобработки области интереса.

Изображения с камер видеонаблюдения подвержены влиянию негативных факторов внешней среды: изменение освещенности сцены, цифровой и аналоговый шум, потеря фокуса, погодные условия. Для снижения их влияния необходимо выполнить предобработку каждого кадра из $\mathbf{R}_k^{\langle 2\rangle} = \left[\mathbf{I}_{t-i+1}, \mathbf{I}_{t-i+2}, \ldots, \mathbf{I}_{t-i+d+1}, \ldots, \mathbf{I}_t\right]$ при помощи алгоритма устранения помех [18], и в том числе с использованием нечеткой логики [14], результатом которых будет предобработанная область интереса $\mathbf{R}_k^{\langle 2*\rangle} = [\mathbf{R}_{k_{t-i+1}}^{\langle 2*\rangle}, \mathbf{R}_{k_{t-i+2}}^{\langle 2*\rangle}, \text{K}, \mathbf{R}_{k_t}^{\langle 2*\rangle}]$. Аффинные преобразования в автоматическом режиме, как правило, не применяются.

6. Выделение информативных признаков.

Для кодирования видеофрагмента длительностью i кадров необходимо выполнить последовательное кодирование каждого кадра видеофрагмента и дальнейшее их “склеивание”.

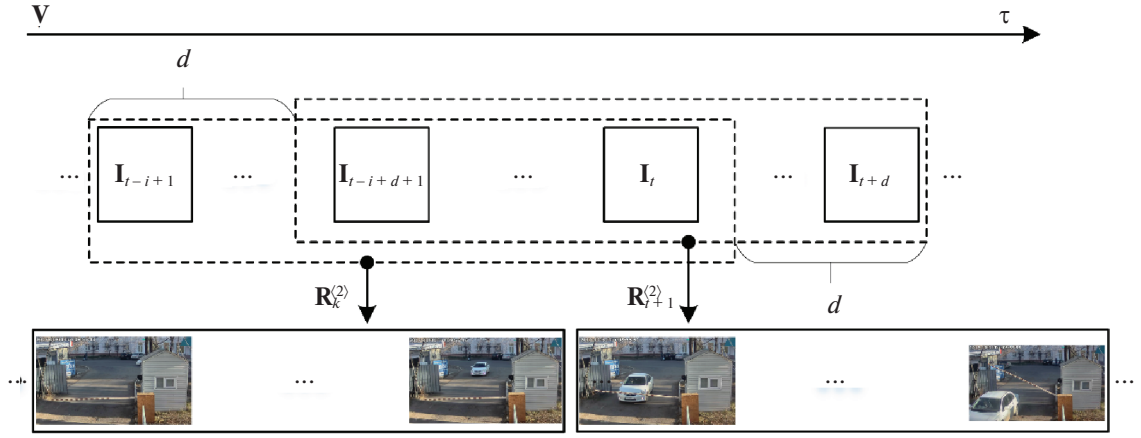

Рис. 2. Пример работы сканирующего окна по набору кадров

В качестве признакового пространства предлагается использовать признаки, полученные из предобученной сверточной НС Inception v3 [19]. Архитектура Inception v3 состоит из чередующихся слоев свертки, подвыборки и последнего полносвязного слоя.

Для получения карты признаков из НС Inception v3 необходимо выбрать уровень глубины, достаточный для выявления ключевых зависимостей. С учетом специфики задачи нам требуется самый глубокий уровень карты признаков, т.е. предлагается получать оценку каждого кадра из последнего слоя подвыборки с операцией GlobalAveragePooling. На данном слое оценка кадра будет представлять собой вектор-столбец размером 2048 × 1.

На вход НС последовательно подаются предобработанные матрицы-кадры из области $\mathbf{R}_k^{\langle 2*\rangle}$, начиная с $\mathbf{R}_{k_t}^{\langle 2*\rangle}$. Тогда оценка предбоработанного кадра с номером t записывается как его оценка $\tilde{\mathbf{x}}$. Признаковое описание видеофрагмента получается конкатенацией оценок всех кадров $\tilde{\mathbf{x}} = concat\ (\tilde{\mathbf{x}}_{t-i+1}, \tilde{\mathbf{x}}_{t-i+2}, \ldots, \tilde{\mathbf{x}}_t) = [\tilde{\mathbf{x}}_{:,t-i+1}, \tilde{\mathbf{x}}_{:,t-i+2}, \ldots, \tilde{\mathbf{x}}_{:,t}]$ и представляет собой матрицу размером 690×2048, каждый столбец которой является оценкой соответствующего кадра этого видеофрагмента.

7. Отнесение вектора признаков к одному из классов.

Для формирования множества классов **B** нами за основу был взят датасет UCF Crime [20], содержащий 1800 видеофрагментов штатных и нештатных ситуаций продолжительностью от 7 до 900 с.

Результаты, приведенные в оригинальной работе [20], на данном датасете UCF Crime показывают невысокую точность 28.4% с использованием временных сверточных НС (англ. temporal convolutional neural network, TCNN) и 23% с применением трехмерной сверточной сети (C3D со слоем 3d-max-pooling). Предложенные в [20] методы распознавания ситуаций неудовлетворительно работают с этим датасетом из-за высокой внутриклассовой вариации (изменчивости) и большой продолжительностью видеофрагментов.

Нами также были проведены дополнительные исследования, подтверждающие результаты авторов статьи [20], и выявлены недостатки существующего датасета, не позволяющие достигнуть более высокой точности:

– большая часть видеофрагментов содержит нештатную ситуацию длительностью намного меньше, чем длительность видеоролика;

– кроме того, что представленные видеофрагменты обладают внутриклассовой вариацией, некоторые ситуации маловероятны в системе безопасности предприятия.

Поэтому нами для обучающего множества был изменен датасет следующим образом:

– видеофрагменты были порезаны на части по 30 с окном сканирования с шагом в 10 с;

– каждый получившийся видеофрагмент был проверен и экспертом отнесен к одному из 5 классов множества **B**.

Таблица 2. Обучающая и тестирующая выборка

Номер класса	1	2	3	4	5
Количество обучающих роликов/тестирующих	66/24	46/16	43/16	41/14	230/77

В множестве **B** содержатся следующие ситуации: 1 – нападение/драка (assault), 2 – пожар/взрыв (fire/explosion), 3 – огнестрельное оружие (gun), 4 – дорожное происшествие (road accident), 5 – штатная ситуация (normal event). В табл. 2 приведено разбиение множества **B** по классам на обучающие и тестирующие примеры.

В качестве классификатора $\mathbf{f}_4$ нами разработана архитектура глубокой НС $\mathbf{f}_4^{event}$, построенная различными комбинациями слоев свертки и независимых рекуррентных слоев IndRNN [21] (рис. 3). Архитектура состоит из двух частей с независимыми входами.

Ветка архитектуры НС слева состоит из двух независимых рекуррентных слоев IndRNN с функцией активации σ^{ReLU}:

$$\mathbf{h}^l_{IndRNN_t} = \sigma^{ReLU}(\mathbf{W}\mathbf{h}^{l-1} + \mathbf{U} \circ \mathbf{h}^l_{IndRNN_{t-1}} + \mathbf{b}^l),$$

где $\mathbf{h}^l_{IndRNN}$ – выходной вектор; $\mathbf{W}$, $\mathbf{U}$ – матрицы весовых коэффициентов; $\mathbf{h}^{l-1}$ – входной вектор; $\circ$ – произведение Адамара; σ^{ReLU} – функция активации ReLU; $\mathbf{b}^l$ – вектор смещения, $\mathbf{h}^l_{IndRNN_{t-1}}$ – выходной вектор предыдущего шага.

Правая часть представлена слоем подвыборки с операцией GlobalMaxPooling и полносвязным слоем функций активации σ^{ReLU}. Выходы обеих ветвей архитектуры НС объединяются слоем конкатенации и двумя последовательно идущими полносвязными слоями с функциями активации σ^{ReLU} и $\sigma^{Softmax}$ соответственно. Результатом классификатора $\mathbf{f}_4^{event}$ является получение вектора вероятностей $\mathbf{p}_{\tilde{\mathbf{x}}}$ размером 5×1.

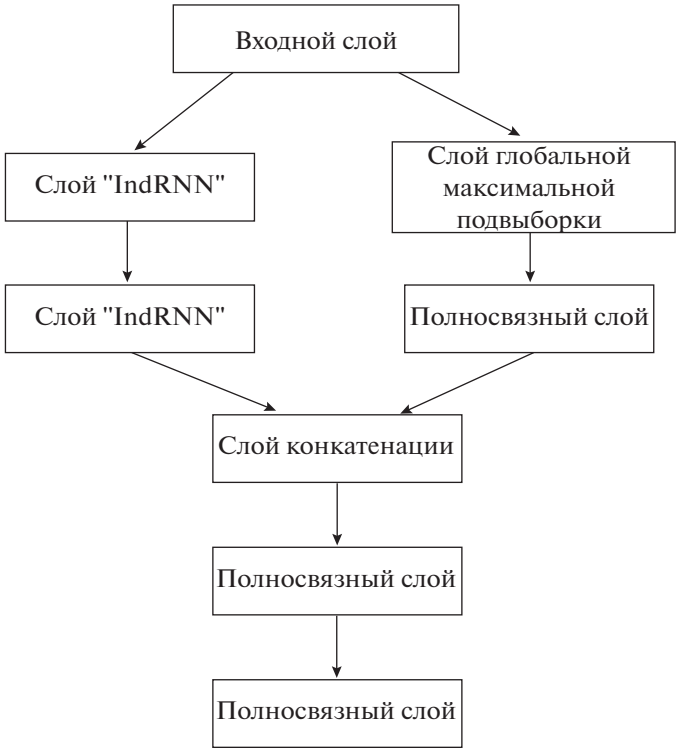

Рис. 3. Архитектура предлагаемой глубокой НС для классификации видеофрагментов

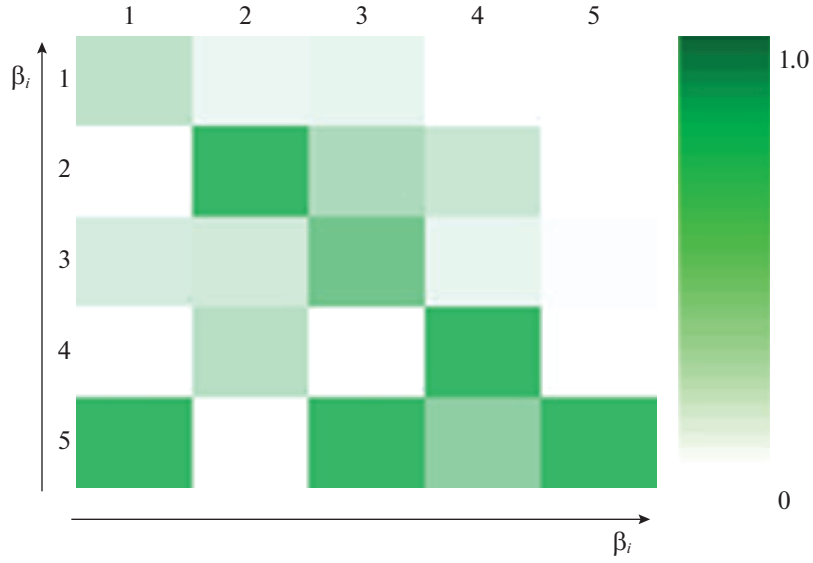

Рис. 4. Результаты тестирования $\mathbf{f}^{event}$

На рис. 4 приведена матрица ошибок для тестирующей выборки. Метрики Precision и Accuracy алгоритма составили 55 и 65.5%. Учитывая, что критерий классификации при распознавании ситуаций $\max \mathbf{p}_{\tilde{\mathbf{x}}} \geq 0.7$, ставится задача усилить оценку классификатора.

Для подкрепления полученной оценки классификации $\mathbf{p}_{\tilde{\mathbf{x}}}$ предлагается извлекать ключевые слова из автоматической аннотации кадра, что позволит исключить ошибки классификации и распознавать нештатные ситуации, не учтенные ранее в системе, тем самым усиливая или ослабляя данную оценку.

Пусть имеется $\tilde{\mathbf{x}}$ оценка конечного видеофрагмента, заданная признаковым описанием. Произвольный столбец матрицы $\tilde{\mathbf{x}}$ представляет собой признаковое описание соответствующего кадра $\mathbf{I}_t$. Алгоритмом автоматического аннотирования кадра будет называться отображение $\mathbf{f}^{agr}: \tilde{\mathbf{x}}_{:,t} \rightarrow \mathbf{S}$, сопоставляющее вектор признаков кадра видеопотока вектору его текстового описания $\mathbf{S} = (v_1,\ldots,v_i,\ldots,v_n)$, где n – количество слов в предложении, а v_i – слово с присвоенным уникальным индексом из словаря $\mathbf{W}$, полученного из часто используемых слов при обучении НС.

В качестве $\mathbf{f}^{agr}$ нами предлагается архитектура НС, представленная на рис. 5. В отличие от используемых ранее архитектур, вместо слоя LSTM предлагается применение слоя IndRNN, что повысит скорость работы НС. Предложенная архитектура НС была обучена на ГПУ Nvidia GeForce 1080Ti и с помощью обучающего множества MS COCO [22], содержащего 414113 текстовых описаний для 82783 изображений. Общее время обучения составило 24 ч: 60 эпох, 24846780 итераций.

Предлагаемая архитектура состоит из двух частей с независимыми входами. Первая часть представлена полносвязным слоем функций активации σ^{linear}, обрабатывающим признаки кадра. Вторая часть принимает последовательность индексов слов и состоит из следующих слоев:

– слой встраивания (англ. embedding) выполняет приведение целочисленных значений присвоенных словам индексов к векторам фиксированного размера;

– слой IndRNN позволяет выявить закономерности в последовательности векторов фиксированного размера.

В итоге происходит объединение частей слоем конкатенации и двумя последовательно идущими полносвязными слоями с функциями активации $\sigma^{\mathrm{Re}\,LU}$ и $\sigma^{Soft\max}$ соответственно.

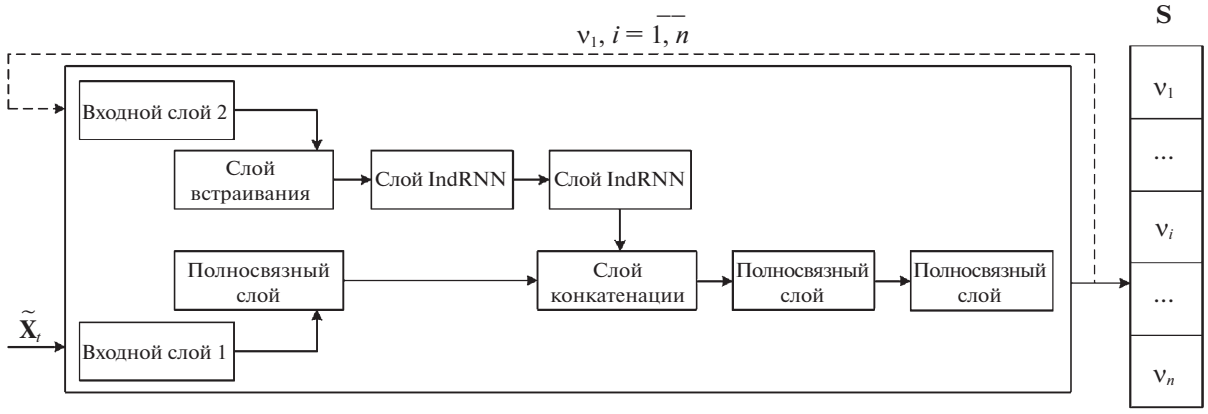

Рис. 5. Архитектура глубокой НС для аннотирования событий

Алгоритм работы $\mathbf{f}^{agr}$ следующий.

1. Вход $\tilde{\mathbf{x}}_{:,t}$ и $v_0 = 0$ – пустое начальное слово.
2. Выход v_1 – первое слово.
3. $\mathbf{S} \leftarrow v_1$.
4. Пока $v_i \neq$ "STOP":
   а) вход $\tilde{\mathbf{x}}_t$ и вектор $\mathbf{S}$;
   б) выход v_i – следующее слово;
   в) $\mathbf{S} \leftarrow concat(\mathbf{S}, v_i)$, где $concat$ – операция конкатенации.
5. Выход $\mathbf{S}$.

Для повышения точности аннотирования видеофрагмента предлагается извлекать три кадра $\tilde{\mathbf{x}}_{:,t}$, $\tilde{\mathbf{x}}_{:,(t+i)/2}$, $\tilde{\mathbf{x}}_{:,t+i}$ из $\tilde{\mathbf{x}}$, выполнять аннотирование каждого из них по алгоритму $\mathbf{f}^{agr}$ и формировать вектор $\mathbf{S}_k$ из слов, встретившихся более 1 раза.

Для подкрепления классификации нештатных ситуаций нами использовалась таблица ключевых слов, сгруппированных вручную по первым четырем классам событий из $\mathbf{B}$.

В качестве ключевых предлагаются следующие слова из аннотаций MS COCO, содержащих 17 000 слов. В табл. 3 приведен фрагмент списка используемых ключевых слов. Стоит обратить внимание, что одно и то же слово может учитываться при подкреплении нескольких нештатных ситуаций.

Пусть $\mathbf{S} = \{v_1, \ldots, v_d\}$ – конечное множество ключевых слов размером d, причем $\mathbf{S} \subseteq \mathbf{W}$. Множество $\mathbf{S}$ разбито на четыре части так, что $\mathbf{S}_m \subset \mathbf{S}$, $\mathbf{S}_m \neq \varnothing$ и, возможно, $\boldsymbol{S}_m \subset \boldsymbol{S}_l$, где $m, l = \overline{1,4}$.

Таблица 3. Ключевые слова

Номер класса	1 (драка)	2 (пожар)	3 (стрельба)	4 (дорожное происшествие)
Ключевые слова	Blood	Fire	Fire	Fire
	Group	Smoke	Smoke	Smoke
	Jumping	Blaze	Blood	Oil
	Hit	Flash	Blaze	Blood
	Pulling	Lights	Flash	Damage
	Running	Lit	Wii	Smashed
	...	...	...	...

Таблица 4. Результаты расчетов основных метрик для метода

M^{Pr}	M^{Rec}	M^{AC}
0.55	0.50	0.655

Таблица 5. Время обработки видеофрагмента из 690 кадров/с

Конфигурация ПК			
1	2	3	4
7.05	3.36	3.06	1.43

Результатом алгоритма аннотирования $\mathbf{f}^{agr}$ является вектор $\mathbf{S}_k = (v_1,...,v_j,...,v_r)$, содержащий r ключевых слов из трех кадров. Для расчета качества алгоритма аннотирования $\mathbf{f}^{agr}$ используется метрика BLEU (англ. bilingual evaluation understudy) [23]. Результаты расчетов точности алгоритма аннотирования составили 48%.

Для того, чтобы соотнести вектор $\mathbf{S}_k$ к одному из классов, необходимо построить вектор подкрепления $^*\mathbf{p}$ размером 5×1, состоящий из единиц. Тогда для каждого слова v_j выполняется правило “Если слово входит в один или несколько классов, то необходимо прибавить 0.1 к элементам вектора $^*\mathbf{p}$, индекс которых соответствует классам ключевого слова”:

$$\text{Если} \quad v_j \in \mathbb{S}_m \quad \text{то} \quad {}^*\mathbf{p}_m = {}^*\mathbf{p}_m + 0.1,$$

где ${}^*\mathbf{p}_m$ – элемент вектора $^*\mathbf{p}$ с индексом $m = \overline{1,4}$.

После выполнения правила для каждого слова верно произведение Адамара ($\circ$) над вектором подкрепления и вектором классификации $\mathbf{p}_{\tilde{\mathbf{x}}} = \mathbf{p}_{\tilde{\mathbf{x}}} \circ {}^*\mathbf{p}$ с последующим нормированием элементов итогового вектора. Нормирование выполняется путем деления каждого элемента вектора $\mathbf{p}_{\tilde{\mathbf{x}}}$ на сумму всех его элементов, что позволит привести результат к вероятностному виду.

8. Критерий классификации определяется как $J(\mathbf{f}_4) = \max \mathbf{p}_{\tilde{\mathbf{x}}}$. Если $J(\mathbf{f}_4) \geq \varepsilon$, где $\varepsilon = 0.7$, то $\beta_i = \arg\max_{i \in 1..c}(\mathbf{p}_{\tilde{\mathbf{x}}})$, в противном случае классификация считается ошибочной.

Задача алгоритма $\mathbf{f}_4^{event}$ в действующей системе безопасности – оповещать и акцентировать внимание на нужной сцене многомониторного пульта для принятия решения.

Так как результатом работы алгоритма классификации является вектор $\mathbf{p}_{\tilde{\mathbf{x}}}$, содержащий вероятности классификации для каждого видеофрагмента, то для работы в реальной системе нам необходимо выставить порог доверия, исключающий ложные срабатывания. Если $\max \mathbf{p}_{\tilde{\mathbf{x}}} \geq 0.7$, то необходимо найти индекс максимального элемента $i^* = \arg\max_{i \in 1..5}(\mathbf{p}_{\tilde{\mathbf{x}}_i})$, в противном случае классификация считается ошибочной.

Индекс i^* определяет номер класса ситуации из множества $\mathbf{B}$. С учетом порога доверия алгоритм $\mathbf{f}_4^{event}$ становится “жестче” к ложным срабатываниям, однако понизит точность алгоритма.

Эксперимент 1. Классификация ситуаций видеопотока возможна как в стационарной системе видеонаблюдения, так и с помощью БПЛА.

В эксперименте использовались видеофрагменты, полученные с камер действующей системы безопасности и видеоролики из сети Интернет. В качестве алгоритма $\mathbf{f}_4^{event}$ применялись архитектуры НС на базе слоя IndRNN. Видеофрагменты содержали различные виды ситуаций длительностью от 7 до 120 с.

В табл. 4 и 5 приведены метрики метода при решении задачи распознавания штатных и нештатных ситуаций.

Соотношение штатных ситуаций к нештатным в обучающей выборке составляет примерно 1:1, что позволяет использовать предлагаемые архитектуры как бинарные классификаторы. Точность бинарной классификации составила 80%.

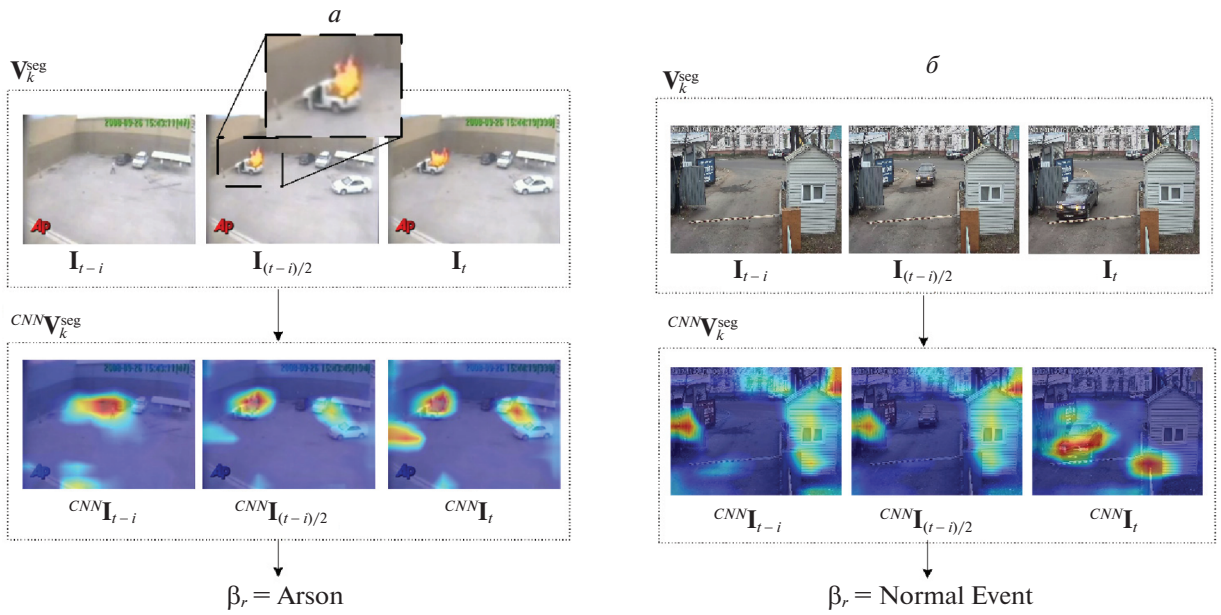

Рис. 6. Результаты классификации алгоритма $\mathbf{f}^{event}$: *а* – пример ситуации с пожаром, *б* – пример штатной ситуации с камеры наблюдения университета

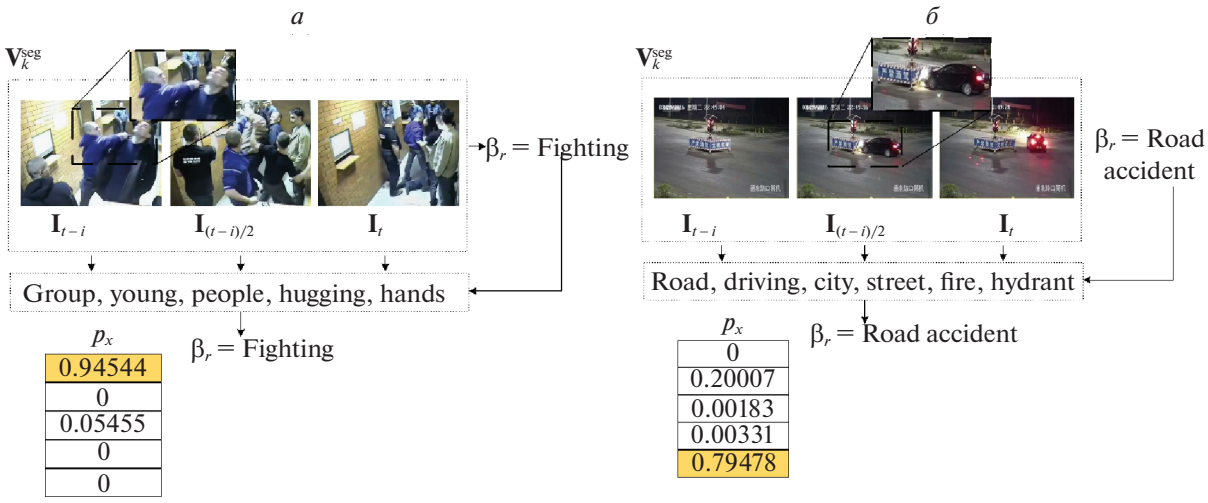

Рис. 7. Ситуации, требующие дополнительного анализа: *а* – драка, *б* – дорожное происшествие

При замерах производительности на конфигурации 4 (табл. 1) были получены следующие показатели:

– время классификации видеофрагмента в 30 с (690 кадров) алгоритмом $\mathbf{f}^{event}$ составляет 1.43 с;

– время аннотации одного кадра алгоритмом $\mathbf{f}^{agr}$ составляет 1.39 с.

На рис. 6 приведен пример пожара (рис. 6, *а*) и штатной ситуации (рис. 6, *б*). Показано признаковое описание кадров видеофрагмента и результат классификации.

На рис. 7 рассмотрены примеры ситуаций, требующие дополнительного анализа: пример драки (рис. 7, *а*) и пример дорожного происшествия (рис. 7, *б*).

При аннотировании каждой из ситуаций были выделены ключевые слова. Если слово попадает в таблицу ключевых слов (табл. 3), то значение соответствующего элемента $\mathbf{p}_{\tilde{\mathbf{x}}}$ увеличивается. Для ситуации рис. 7, *а* ключевые слова: hands, shaking, для ситуации рис. 7, *б* ключевые слова: fire, car.

Результатом работы блока подкрепления является увеличение вероятностной оценки классификации или вывод об ошибке работы алгоритма $\mathbf{f}^{event}$, если ни один из элементов вектора $\mathbf{p}_x$ так и не превысил порога 0.7.

При выполнении алгоритма подкрепления итоговая точность алгоритма обнаружения и распознавания штатных и нештатных ситуаций повысилась до 80%.

Стоит отметить, что при использовании БПЛА обработку необходимо выполнять на ГПУ Jetson TX2, что по производительности примерно соответствует конфигурации 1 (табл. 5).

2.2. Распознавание НЗ ТС. С помощью предложенного выше вычислительного метода можно осуществлять распознавание объектов по отдельным кадрам. Например, кратко рассмотрим распознавание НЗ ТС, уделяя внимание только важным особенностям применения вычислительного метода.

В качестве алгоритма поиска объекта (этап 2) предлагается использовать предобученную глубокую НС YOLO [24].

Архитектура сети YOLO основана на модели GoogLeNet [25] и состоит из 24 чередующихся слоев свертки (англ. convolution), подвыборки (англ. pooling, subsampling) и 2 последних полносвязных (англ. fully connected, dense) слоев.

Как алгоритм локализации НЗ (этап 4) применяется классический алгоритм Виолы–Джонса [26] с каскадным классификатором “haarcascade_russian_plate_number.xml”, представленным в библиотеке OpenCV [27].

В качестве предобработки области интереса (этап 5) после поворотов изображения для выравнивания символа к изображению ${}^{g}\mathbf{R}^{\langle 2\rangle}$ применяется метод контрастно-ограниченного адаптивного выравнивания гистограммы (англ. contrast limited adaptive histogram equalization, CLAHE) [28], который анализирует и выравнивает гистограммы локальных областей изображения.

Необходимо также выполнить повтор этапов вычислительных методов 4 и 5 для выделения символов НЗ алгоритмом поиска максимально устойчивых областей экстремума (англ. maximally stable extremal regions, MSER) [29] и с последующей бинаризацией области интереса.

В качестве признакового пространства (этап 6) предлагается использовать признаки, полученные модифицированной архитектурой глубокой НС MobileNet [30]. Необходимо переобучить НС с помощью собственного промаркированного датасета **D**.

Для распознавания символов НЗ предлагается модифицировать базовую архитектуру MobileNet следующим образом.

1. Понизить размерность входного слоя до 50 × 50.

2. Удалить два последних слоя.

3. Добавить четыре новых слоя:

– для снижения признакового пространства добавляется скрытый полносвязный слой размером 128 нейронов, в котором $\mathbf{h}^{l-1}$ имеет размер 2 × 2 × 1024, а в качестве функции активации используется $\sigma^{\mathrm{Re}\,LU}$;

– для лучшего выделения признаков добавляется второй скрытый полносвязный слой размером 2 × 2 × 128 нейронов;

– для приведения выходного массива 2 × 2 × 128 к одномерному вектору 1 × 512 добавляется слой выравнивания (англ. flatten).

Для решения задачи классификации (этап 7) к модифицированной НС добавляется последний полносвязный слой из 23 нейронов и функцией активации Softmax.

Эксперимент 2. С камеры наблюдения, установленной на въезде в ФГБОУ ВО “КнАГУ”, нами была собрана и промаркирована тестирующая выборка, состоящая из 2453 видеофрагментов, на которых содержатся или отсутствуют ТС [31].

Следует отметить, что рассматривается результат работы вычислительного метода относительно всего НЗ, т.е. пропуск существующего НЗ на изображении ТС или ошибка метода только в одном символе НЗ приводит к записи результата в *FN*. Результаты расчетов работы метода представлены в табл. 6 и 7.

На собранной тестирующей выборке также была проведена оценка использования классического алгоритма TesseractOCR [32] в качестве шагов 6 и 7. В табл. 7 приведены результирующая

Таблица 6. Результаты расчетов основных метрик

M^{Pr}	M^{Rec}	M^{AC}	M^{F}	M^{TPR}	M^{FPR}	M^{AUC}	$M_c^{\overline{Pr}}$	$M_c^{\overline{Rec}}$
1	0.919	0.9682	0.95779	0.95779	0	0.9595	0.998	0.998

Таблица 7. Результаты расчетов основных метрик классификатора на базе Tesseract

$M_c^{\overline{Pr}}$	$M_c^{\overline{Rec}}$
0.867	0.8665

точность и полнота классификатора на базе TesseractOCR для пользователей. На рис. 8 продемонстрирована работа метода в СКУД университета.

В результате точность M^{Pr} для предложенного метода равна 1. Данный показатель достигнут “жесткими” ограничениями метода, посредством увеличения порога ε при локализации НЗ. Время обработки одного кадра (табл. 8), содержащего НЗ ТС – от 0.03 до 0.09375 с, что позволяет применять предложенный метод в СКУД в режиме РВ.

2.3. Распознавание лиц. Также кратко рассмотрим распознавание людей по лицу, уделяя внимание только важным особенностям вычислительного метода.

В качестве алгоритма поиска объекта (этап 2) предлагается использовать предобученную глубокую НС YOLO.

Как алгоритм локализации лица (этап 4) может быть применен алгоритм HOG [33] или SSD MultiBox [34].

Для нормализации и предобработки (этап 5) выполняется поиск “точек ориентира” лица при помощи алгоритма FaceLandmark [35].

Далее, после нахождения центров обоих глаз выполняется поворот изображения вокруг средней точки между ними (x^{Ceye}, y^{Ceye}) и масштабирование изображения по матрице:

$$\mathbf{M} = \begin{bmatrix} k\cos(\theta) & k\sin(\theta) & (1-k\cos(\theta))x^{Ceye} - k\sin(\theta)y^{Ceye} \\ -k\sin(\theta) & k\cos(\theta) & k\sin(\theta)x^{Ceye} + (1-k\cos(\theta))y^{Ceye} \end{bmatrix}.$$

Результатом аффинных преобразований является матрица $\mathbf{R}^{\langle 2* \rangle}$.

В качестве признакового пространства (этап 6) предлагается использовать признаки, полученные из модифицированной архитектуры глубокой НС MobileNet v2 [36] без двух последних слоев.

НС была переобучена на выборке CASIA-WebFace [37], в которой содержится 452960 лиц, разделенных на 10575 классов. Время переобучения НС составило 5 дней.

Рис. 8. Пример работы метода в СКУД ТС

Таблица 8. Время обработки 1 кадра/с

Этапы метода		Конфигурация ПК			
		1	2	3	4
1, 2. Захват кадра и локализация ТС		0.277	0.094	0.089	0.03125
3, 4. Локализация номерного знака		0.055	0.013	0.015	0.01562
5. Предобработка		0.063	0.021	0.023	0.01562
Повтор 4, 5. Сегментация символов НЗ		0.051	0.041	0.031	0.03124
6, 7. Классификация	Оригинальный алгоритм	0.022	0.008	0.007	0.0001
	TesseractOCR	1.705	0.746	0.644	0.500
Общее время обработки кадра	Оригинальный алгоритм	0.472	0.1779	0.16	0.09375
	TesseractOCR	1.781	0.868	0.7063	0.555

Таблица 9. Результаты расчетов основных метрик

M^{Pr}	M^{Rec}	M^{AC}	M^{F}	M^{TPR}	M^{FPR}	M^{AUC}
0.982	0.879	0.931	0.927	0.879	0.9839	0.936

Как алгоритм сравнения (этап 7) предлагается использовать дуальную архитектуру глубокой НС, в которой предусмотрено два входа: на первый вход поступает эталон в виде вектора признаков, а на второй вход – вектор признаков объекта $\tilde{\mathbf{x}}$. В качестве выхода НС применяется полносвязный слой с одним нейроном и функцией активации $\sigma^{Sigmoid}$.

Эксперимент 3. Для тестирования предложенной архитектуры глубокой НС использовались 874 (позитивных 439/негативных 435) примера из общедоступного набора данных для тестирования LFW [38], в которых была правильно выполнена локализация лица. Результаты оценки вычислительного метода представлены в табл. 9.

Точность предлагаемого нами метода (0.931) на общедоступном датасете LFW выше, чем точность, приведенная на странице LFW, полученная классическим методом [4] (0.929).

Для тестирования предложенного метода в холле университета была установлена камера наблюдения, а также использовался набор снятых видеороликов (рис. 9).

В качестве базы эталонов применялись 566 фотографий лиц различных сотрудников ФГБОУ ВО “КнАГУ”. Оценивание быстродействия вычислительного метода определялось по времени обработки одного кадра из видеопотока при различных конфигурациях персонального компьютера (ПК) (табл. 10).

Точность M^{Pr} метода составила 0.982. Результат данного показателя достигнут за счет того, что при обучении НС в обучающей выборке в 2 раза преобладали негативные образы при сравнении. Время обработки одного кадра, содержащего лицо человека – от 0.03 до 0.05 с, т.е. применение вычислительного метода для задачи распознавания лица человека возможно в режиме РВ.

Все вышеперечисленные примеры соответствуют этапам предложенного вычислительного метода.

Реализация рассмотренного метода с помощью глубоких НС позволяет достичь необходимой точности и быстродействия, достаточного для работы в режиме РВ.

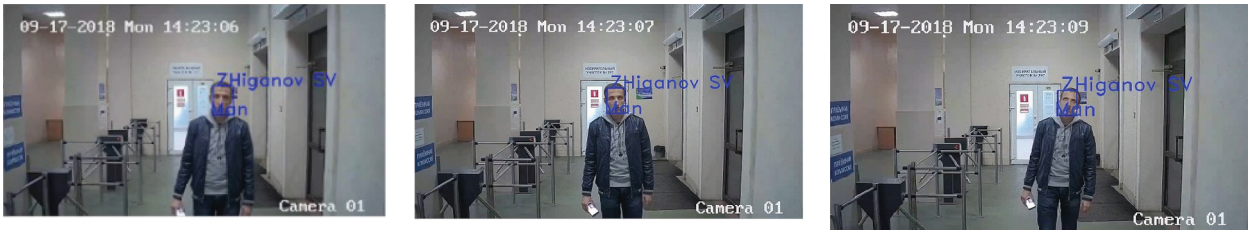

Рис. 9. Пример работы метода в РВ по видеофрагментам с камеры наблюдения, установленной в холле университета

Таблица 10. Время обработки 1 кадра/с

Конфигурация ПК			
1	2	3	4
0.14	0.09	0.08	0.05

Заключение. Приведена математическая формулировка задачи обнаружения и классификации объектов и ситуаций. Для ее решения предложен и разработан вычислительный метод распознавания образов в непрерывном видеопотоке с использованием глубоких НС. Эффективность метода с точки зрения точности и быстродействия достигается классификацией с последующим подкреплением на основе нескольких кадров видеопотока и алгоритмом автоматического аннотирования изображений.

Предложена структурная схема интеллектуальной СКУД, задачами которой являются: распознавание символов номерного знака, лиц людей, нештатных ситуаций. Отличительная особенность предлагаемой СКУД – использование мультироторного БПЛА с вычислительным модулем.

Для решения задач интеллектуальной СКУД разработаны и протестированы адаптированные архитектуры НС с независимыми рекуррентными слоями для классификации нештатных ситуаций по видеофрагментам, дуальная (сиамская) сеть для распознавания лиц, глубокая НС для распознавания символов НЗ ТС. Для обучения и тестирования НС были созданы оригинальные базы данных с использованием общедоступных наборов и промаркированных видеофрагментов, полученных с действующей СКУД. Все нейросетевые алгоритмы были реализованы в виде программных модулей на языке Python.

Применение предлагаемого вычислительного метода к задаче распознавания НЗ в сложных условиях уличного видеонаблюдения обеспечивает точность не менее 96%, а время обработки одного кадра – не более 0.094 с на базе графического процессора Nvidia GeForce 1080Ti. Точность вычислительного метода при решении задачи распознавания лиц составила 98.2% на общедоступной базе, время обработки одного кадра, полученного с видеокамеры, – не более 0.05 с. При решении задачи распознавания нештатных ситуаций с помощью вычислительного метода была достигнута точность до 80% и скорость 1.43 с для видеофрагмента длиной 30 с. В результате натурных экспериментов доказана эффективность и возможность применения предлагаемого метода для распознавания образов в различных предметных областях в режиме РВ.

СПИСОК ЛИТЕРАТУРЫ

1. *Alimi A.M., Pal U., Halima M.B., Selmi Z.* DELP-DAR System for License Plate Detection and Recognition // Pattern Recognition Letters. 2020. № 129. P. 213–223.
2. *Silva S.M., Jung C.R.* License Plate Detection and Recognition in Unconstrained Scenarios // European Conf. on Computer Vision (ECCV). Munich, Germany, 2018. P. 580–596.
3. *Aarathi K.S., Abraham A.* Vehicle Color Recognition Using Deep Learning for Hazy Images // International Conf. on Inventive Communication and Computational Technologies (ICICCT). Coimbatore, India, 2017. P. 335–339.
4. *Amos B., Ludwiczuk B., Satyanarayanan M.* OpenFace: A General-purpose Face Recognition Library with Mobile Applications // CMU-CS-16-118, CMU School of Computer Science, Tech. Rep. 2016. URL: https://www.cs.cmu.edu/~satya/docdir/CMU-CS-16-118.pdf (дата обращения: 20.08.2019).
5. *Chen S., Liu Y., Gao X., Han Z.* MobileFaceNets: Efficient CNNs for Accurate Real-Time Face Verification on Mobile Devices // Chinese Conf. on Biometric Recognition (CCBR). Urumchi, China, 2018. P. 428–438.
6. Results page [электронный ресурс] // Labeled Faces in the Wild: [сайт]. URL: http://vis-www.cs.umass.edu/lfw/results.html (дата обращения: 10.01.2020).
7. *Schroff F., Kalenichenko D., Philbin J.* FaceNet: A Unified Embedding for Face Recognition and Clustering // IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Boston, MA, USA, 2015. P. 815–823.
8. *Organisciak D., Riachy C., Aslam N., Shum H.P.H.* Triplet Loss with Channel Attention for Person Re-identification // J. of WSCG. 2019. V. 27. № 2. P. 161–169.
9. *Hinami R., Mei T., Satoh S.* Joint Detection and Recounting of Abnormal Events by Learning Deep Generic Knowledge // IEEE Intern. Conf. on Computer Vision. Venice, Italy, 2017. P. 3619–3627.
10. *Anala M.R., Makker M., Ashok A.* Anomaly Detection in Surveillance Videos // 26th Intern. Conf. on High Performance Computing. Data and Analytics Workshop (HiPCW). Hyderabad, India, 2019. P. 93–98.
11. *Амосов О.С.* Фильтрация марковских последовательностей на основе байесовского, нейросетевого подходов и систем нечеткой логики при обработке навигационной информации // Изв. РАН. ТиСУ. 2004. Т. 43. № 4. С. 61–69.
12. ImageNet [электронный ресурс] URL: http://www.image-net.org/ (дата обращения: 15.12.2019).

13. Machine Learning Tips and Tricks Cheatsheet [электронный ресурс] URL: https://stanford.edu/~shervine/teaching/cs-229/cheatsheet-machine-learning-tips-and-tricks (дата обращения: 20.12.2019).
14. *Amosov O.S., Baena S.G., Ivanov Y.S., Htike S.* Roadway Gate Automatic Control System with the Use of Fuzzy Inference and Computer Vision Technologies // 12th IEEE Conf. on Industrial Electronics and Applications (ICIEA). Siem Reap, Cambodia, 2017. P. 707–712.
15. *Амосов О.С., Амосова С.Г., Иванов Ю.С.* Интеллектуальная система контроля и управления доступом физических лиц // Междунар. конф. по мягким вычислениям и измерениям. Санкт-Петербург, Россия, 2018. Т. 1. С. 352–355.
16. *Amosov O.S., Ivanov Y.S., Zhiganov S.V.* Semantic Video Segmentation with Using Ensemble of Particular Classifiers and a Deep Neural Network for Systems of Detecting Abnormal Situations // IT in Industry. 2018. V. 6. P. 14–19.
17. *Kendall A., Badrinarayanan V., Cipolla R.* Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding // Proc. British Machine Vision Conf. (BMVC). London, UK, 2017. V. 57. P. 57.1–57.12.
18. *Амосов О.С., Иванов Ю.С., Жиганов С.В.* Локализация человека в кадре видеопотока с использованием алгоритма на основе растущего нейронного газа и нечёткого вывода // Компьютерная оптика. 2017. Т. 41. № 1. С. 46–58.
19. *Szegedy C., Vanhoucke V., Ioffe S., Shlens J., Wojna Z.* Rethinking the Inception Architecture for Computer Vision // IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA, 2016. P. 2818–2826.
20. *Sultani W., Chen C., Shah M.* Real-world Anomaly Detection in Surveillance Videos // IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Salt Lake City, Utah, USA, 2018. P. 6479–6488.
21. *Li S., Li W., Cook C., Zhu C., Gao Y.* Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper RNN // IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Salt Lake City, Utah, USA, 2018. P. 5457–5466.
22. COCO dataset [электронный ресурс] URL: http://mscoco.org/ (дата обращения: 10.01.2020).
23. *Papineni K., Roukos S., Ward T., Zhu W.J.* BLEU: a Method for Automatic Evaluation of Machine Translation // ACL-2002: 40th Annual meeting of the Association for Computational Linguistics. Philadelphia, Pennsylvania, USA, 2002. P. 311–318.
24. *Redmon J., Divvala S.K., Girshick R.B., Farhadi A.* You Only Look Once: Unified, Real-Time Object Detection // IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA, 2016. P. 779–788.
25. *Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A.* Going deeper with convolutions. Technical Report // IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Boston, MA, USA, 2015. P. 1–9.
26. *Viola P.J., Snow D.* Detecting Pedestrians Using Patterns of Motion and Appearance // Int. J. Comput. Vision. 2005. V. 63. № 2. P. 153–161.
27. Библиотека компьютерного зрения OpenCV [электронный ресурс] URL: https://github.com/opencv/opencv (дата обращения: 20.02.2020).
28. *Yadav G., Maheshwari S., Agarwal A.* Contrast Limited Adaptive Histogram Equalization Based Enhancement for Real Time Video System // Intern. Conf. on Advances in Computing, Communications and Informatics (ICACCI). New Delhi, India, 2014. P. 2392–2397.
29. *Matas J., Chum O., Urban M., Pajdla T.* Robust Wide Baseline Stereo from Maximally Stable Extremal Regions // Image and Vision Computing. 2004. V. 22. P. 761–767.
30. *Howard A.G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H.* MobileNets: Efficient Convolutional Neural Networks for Mobile Vision // arXiv preprint. 2017. URL: https://arxiv.org/pdf/1704.04861.pdf.
31. Видео с камеры видеонаблюдения СКУД ФГБОУ ВО "КнАГУ" [электронный ресурс] URL: http://evernow.ru/acs.zip (дата обращения: 30.01.2020).
32. Tesseract OCR [электронный ресурс] URL: https://github.com/tesseract-ocr/tesseract (дата обращения: 20.01.2020).
33. *Amosov O.S., Ivanov Y.S., Zhiganov S.V.* Human Localization in the Video Stream Using the Algorithm Based on Growing Neural Gas and Fuzzy Inference // XII Intelligent Systems Sympos. (INTELS'16), Procedia Computer Science. V. 103. Moscow, Russia, 2017. P. 403–490.
34. *Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C., Berg A.C.* SSD: Single Shot MultiBox Detector // Proc. of the European Conference on Computer Vision (ECCV). Amsterdam, Netherlands, 2016. Part 1. V. 9905. P. 21–37.
35. *Shaoqing R., Xudong C., Yichen W., Jian S.* Face Alignment at 3000 FPS via Regressing Local Binary Features // The IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Washington, USA, 2014. P. 1685–1692.
36. *Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L.* MobileNetV2: Inverted Residuals and Linear Bottlenecks // IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Salt Lake City, Utah, USA, 2018. P. 4510–4520.
37. *Dong Y., Zhen L., Shengcai L., Stan Z.L.* Learning Face Representation from Scratch // arXiv preprint. 2014. URL: https://arxiv.org/pdf/1411.7923.pdf (дата обращения: 08.04.2020).
38. Labeled Faces in the Wild [электронный ресурс] [2007]. URL: http://vis-www.cs.umass.edu/lfw/ (дата обращения: 10.02.2020).