

DOI: 10.18721/JCSTCS.11302

УДК 004.896

СИСТЕМА СИТУАЦИОННОГО УПРАВЛЕНИЯ И КОНТРОЛЯ ПЛОХО ФОРМАЛИЗУЕМЫХ СЦЕНАРИЕВ ДИНАМИЧЕСКИХ СЦЕН

А.В. Федоров, В.П. Шкодырев, Н.Д. Барсуков

Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Российская Федерация

Рассмотрена система прогнозирования аварийных и предаварийных ситуаций при вождении транспортного средства с использованием модели IDAS (интеллектуальная система помощи водителю). Система IDAS основана на комбинации длинной и кратковременной памяти (через рекуррентную нейронную сеть) и быстро реагирующей системы на базе контроллера нечеткой логики. Оценена эффективность комбинации нейро-нечеткого контроллера и рекуррентной нейронной сети. Разработаны алгоритм прогнозирования аварийных ситуаций и программное обеспечение для тестирования модели на видеозаписях с видеорегистраторов. Созданная модель позволяет определить аварийную ситуацию не менее чем за секунду до ее начала. Эффективность представленного подхода основана на данных с камер видеорегистраторов. В качестве тестовых данных использовано 1730 видеозаписей лаборатории VSLab с видеорегистраторов в Тайване. Создан мобильный прототип системы прогноза аварийных ситуаций на базе мини-компьютера Raspberry Pi 3.

Ключевые слова: машинное зрение, искусственный интеллект, глубокое обучение, нейронные сети, кластеризация, рекуррентные нейронные сети.

Ссылка при цитировании: Федоров А.В., Шкодырев В.П., Барсуков Н.Д. Система ситуационного управления и контроля плохо формализуемых сценариев динамических сцен // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. 2018. Т. 11. № 3. С. 20–28. DOI: 10.18721/JCSTCS.11302

SYSTEM FOR SITUATION MANAGEMENT AND CONTROL OF BAD FORMALIZED SCENARIOS OF DYNAMIC SCENES

A.V. Fedorov, V.P. Shkodyrev, N.D. Barsukov

Peter the Great St. Petersburg Polytechnic University,
St. Petersburg, Russian Federation

The article considers a system for forecasting emergency situations when driving a vehicle using the IDAS (intelligent driver assistance system) model. The IDAS system is based on a combination of long and short-term memory (via recurrent neural network) and a fast-reacting system based on a fuzzy logic controller. We have evaluated the efficiency of combining a neuro-fuzzy controller and a recurrent neural network. We have developed an algorithm for forecasting emergencies, as well as software for testing the model on video recordings from DVRs. The effectiveness of the proposed approach has been proved on data from dash-cameras. The final time before the accident is near one second before any type of road accident. We have created a mobile prototype for the system for forecasting emergency situations based on the Raspberry Pi 3 mini-computer.

Keywords: machine vision, artificial intelligence, deep learning, neural networks, clusterization, recourse neural networks.

Citation: Fedorov A.V., Shkodyrev V.P., Barsukov N.D. System for situation management and control of bad formalized scenarios of dynamic scenes. St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunications and Control Systems, 2018, Vol. 11, No. 3, Pp. 20–28. DOI: 10.18721/JCSTCS.11302

Введение

Данное исследование сосредоточено на создании системы предсказания аварийных ситуаций при управлении транспортным средством; основное назначение разработки – вернуть транспортное средство в исходное, нормальное состояние, при возникновении штатных и нештатных ситуаций, повседневно возникающих на дороге при вождении транспортных средств. Главный критерий управления такими системами – снижение неопределенностей в системе управления. Несмотря на большой скачок в развитии современных технологий и алгоритмов искусственного интеллекта, даже автономные транспортные средства Google'sself-drivingcar и Tesla все еще попадают в аварии, где в большинстве случаев виноваты другие участники дорожного движения [1]. Кроме того, по статистике только в США каждый день погибает порядка 90 человек в результате ДТП [2]. Для того чтобы избежать этих несчастных случаев предлагается способ для определения на основе данных с автомобильных видеорегистраторов предаварийных ситуаций и предоставления водителю предупреждающего сигнала за несколько секунд до возникновения аварии.

Цель исследования – разработка и изучение системы контроля дорожной ситуации на основе гибридной модели нейронной сети с применением алгоритма глубокого обучения для распознавания и прогнозирования плохо структурированных потоковых событий в условиях неопределенности при вождении транспортного средства.

Нейросетевая адаптивная система прогнозирования нештатных ситуаций

Для построения алгоритма поиска нештатных ситуаций взято исследование [3]. Входными данными для алгоритма являются: описание текущей ситуации

$T = \{t_1, t_2, \dots, t_n\}$, включающее n значений параметров, характеризующих проблему абонента; BC – непустое множество HC ; $W = \{w_1, w_2, \dots, w_n\}$ – веса или коэффициенты важности параметров; M – количество рассматриваемых HC из БЗНС; K – пороговое значение степени сходства. Выходные данные: множество HC SC , которые имеют степень сходства больше (или равную) порогового значения K .

Суть поиска схожей нештатной ситуации заключается в определении дистанции параметров текущей ситуации с параметрами ситуаций, заложенных в базу знаний, умноженных на вес, определяющий значимость данного признака:

$$D(T, C_j) = \sqrt{\sum_{i=1}^n (w_i(x_i^{C_j} - x_i^T))^2}. \quad (1)$$

После определения схожести параметров определяется максимально приближенная ситуация. Если при этом степень схожести будет ниже порогового значения, то данная ситуация будет считаться новой:

$$S(C_j, T) = (1 - D_{C_j T} / D_{MAX}) * 100 \%. \quad (2)$$

Если текущая ситуация не подходит ни под одно описание HC , то текущая ситуация передается на вход нейро-нечеткого модуля (механизм работы алгоритма приведен на рис. 1). Нейро-нечеткий модуль реализован на основе нейросети NNFLC (Neurons Network Fuzzy Logic Controller) или НК (нейро-нечеткий контроллер).

NNFLC – нечеткий контроллер на основе HC . Структурно NNFLC – это многослойная сеть прямого распространения сигнала, реализующая принципы нейросетевой базы знаний, причем различные слои выполняют разные функции. Система NNFLC позволяет представить нечеткие продукционные системы классификации типа Max-Min или типа Takagi-Sugeno в форме ИНС. При проектировании возник-

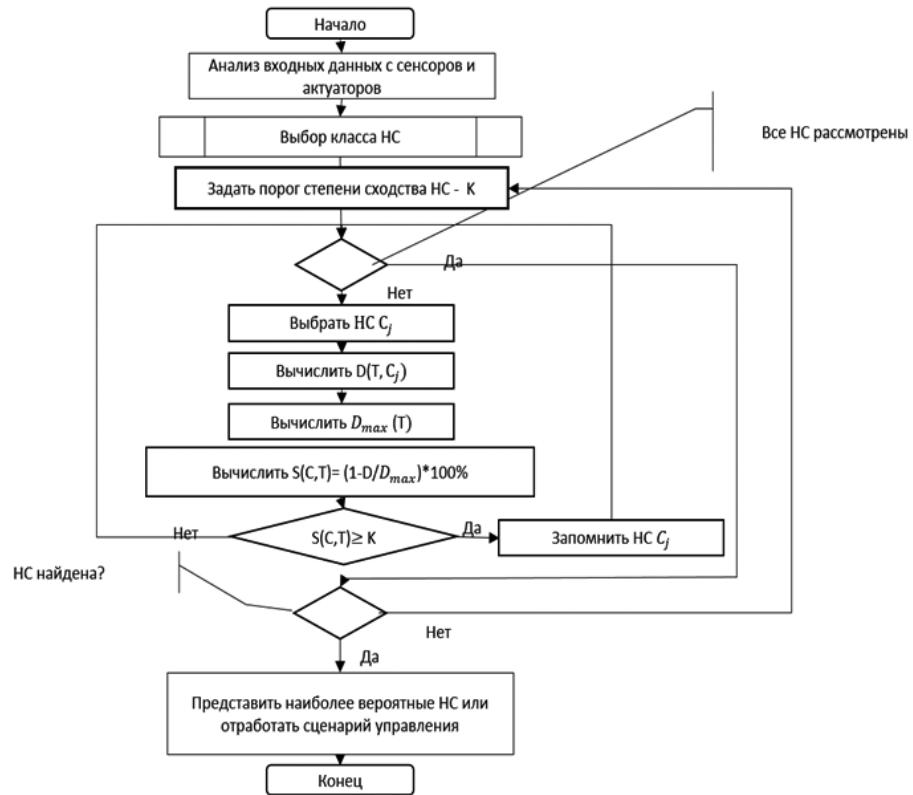


Рис. 1. Алгоритм поиска нештатной ситуации

Fig. 1. The alarms search algorithm

кает четырехслойная нечеткая FF-сеть, в которой каждый слой образуется нейронами определенного типа. Веса связей либо постоянны, либо варьируемы в процессе обучения. Путем использования различных алгоритмов обучения могут варьироваться как свободные параметры, так и структура нейросети, причем в реальном времени.

Модель NNFLC выглядит следующим образом.

Слой 1 (входные термы):

$$y_i^{(1)}(x) = \exp\left[-\frac{(x_i - c_i)^2}{2} * \sigma_i^2\right]. \quad (3)$$

Слой 1 представляет собой функции принадлежности, которые реализованы как радиальные базисные нейроны.

Слой 2 (правила):

$$y_i^{(2)} = \min[y_1^{(1)}, \dots, y_n^{(1)}]. \quad (4)$$

Слой 2 моделирует И-условия правил.

Слой 3 (выходные термы):

$$y_i^{(3)} = \max[y_1^{(1)}, \dots, y_n^{(1)}]. \quad (5)$$

Слой 3 представляет собой ИЛИ-комбинацию правил. В режиме обучения данный слой перенастраивает параметры функций принадлежности выходных переменных, а в рабочем режиме формирует значение выхода.

Слой 4 (выход).

В рабочем режиме нейроны выполняют дефаззификацию:

$$z_i^{(4)} = \sum_j w_{ji} * y_i^{(3)} * \frac{w_{ij}}{\sum_j w_{ji}}, \quad (6)$$

а в режиме обучения это дополнительный вход, позволяющий настроить функцию принадлежности выходной переменной.

Структура АС NNFLC инициализируется по принципу формирования полной матрицы правил:

$$T = \prod_{i=1}^n \tau(x_i). \quad (7)$$

В ходе исследования разработано

математико-программное обеспечение нейросетевой БЗ. Описана разработанная библиотека программ – правил принятия решений. Программа разработана в трех вариантах. Первый вариант работает с промышленными контроллерами (например, с химической станцией) и умеет определять нештатные ситуации в зависимости от данных с датчиков. Вторая версия программы разрабатывалась для определения нештатных ситуаций на дороге, разработка велась на языке Java. Программа умеет определять подключаемое видеоустройство и применять алгоритм обнаружения объектов и их классификации с последующим определением нештатной ситуации и выводом действия, которое нужно предпринять водителю. Третья версия программы включает в себя адаптацию под микрокомпьютер Raspberry Pi 3B, а стандартная модель Tensorflow была дополнена собственноручно обученной нейронной сетью на основе 1730 видеозаписей лаборатории VSLab с видеорегистраторов в Тайване. Пример работы программы приведен на рис. 2.

Сначала определяется наличие движущихся объектов и выделяется область вокруг них при помощи алгоритма SURF и DBSCAN, далее RNN определяет, к чему относится данный объект, и при помощи NNFLC определяется тип нештатной ситуации и вывод принимаемого решения.

Рекурсивная нейронная сеть была реализована при помощи LSTM-ячеек. Идея использовать LSTM-ячейки для прогнозирования аварий встречается в публикациях [7, 8]. В работе [7] исследована зависимость аварий от внешней среды, например от погодных условий, а в [8] предпринята попытка предугадать маневр водителя за несколько секунд до его выполнения. Цель нейронной сети состоит в том, чтобы как можно раньше предсказать будущую аварию с учетом наблюдений $(x_1, x_2, \dots, x_t) \mid t < y$ до того, как авария произойдет в кадре y . T – количество фреймов на видео, y_j – маркер аварии для v_j видео. Стандартная RNN – это особый тип сети, принимающий последовательность наблюдений (x_1, x_2, \dots, x_t) в качестве входных данных и

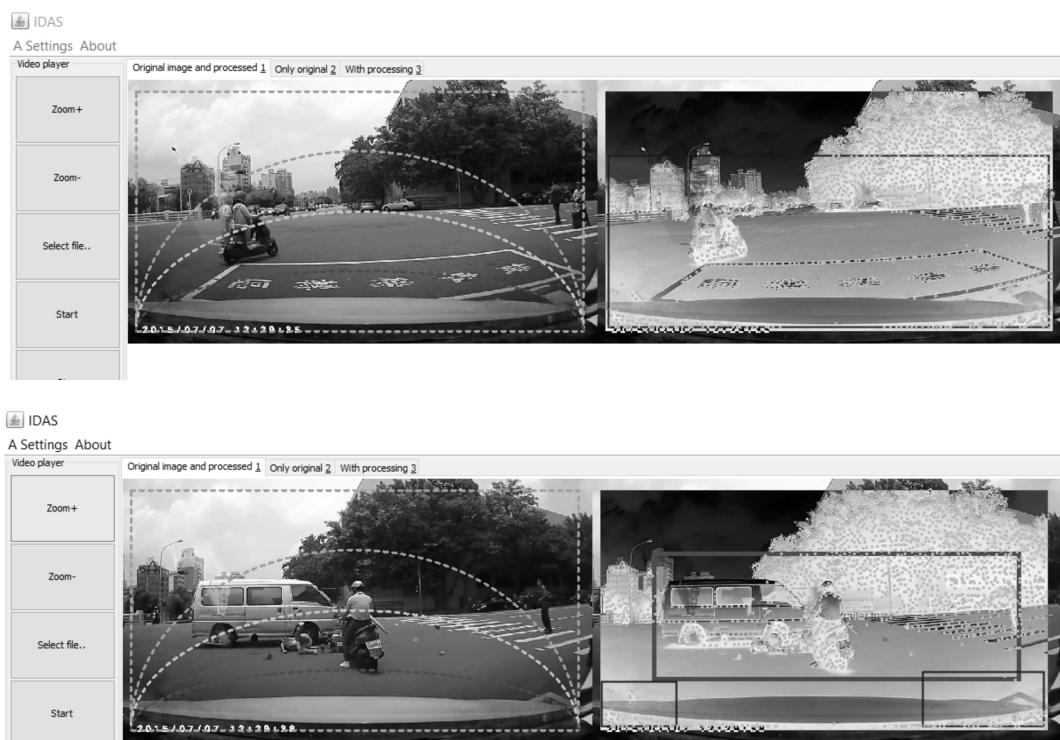


Рис. 2. Принцип работы алгоритма в течение времени

Fig. 2. Algorithm's principle of operation over time

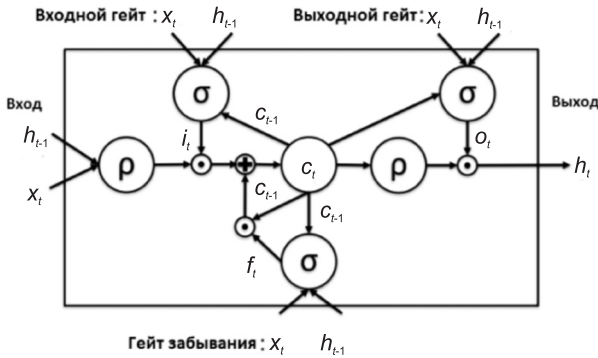


Рис. 3. Модель LSTM:
входной гейт i , гейт забывания f и выходной гейт o ,
 h – скрытое представление, c – ячейка памяти

Fig. 3. LSTM model:
the input gate i , the forgetting gate f and output gate o ,
 h – are a hidden representation, c – is a memory cell

выводящий последовательность изученных скрытых представлений (h_1, h_2, \dots, h_t) , где h_t кодирует наблюдения последовательности (x_1, x_2, \dots, x_t) . Определение опасного объекта на изображении ведется при помощи алгоритма SURF и метода кластеризации DBSCAN. Основным недостатком данного подхода – большая сложность в определении границ объекта. Возможным решением этой проблемы является применение Non-Maximum Suppression (NMS) Object Detection, как описано в статье [9].

Скрытое представление с LSTM генерируется рекурсивным уравнением:

$$(h_t, c_t) = \text{LSTM}(x_t, h_{t-1}, c_t). \quad (8)$$

Принцип работы LSTM-ячейки показан на рис. 3. На рисунке обозначено входной вектор x_t во время t , h_t – вектор скрытого состояния во время t , W_t – матрица весов, применяющаяся к входу, W_h – матрица весов в рекуррентных соединениях, b – вектор свободных членов.

На каждом входе x_t , имея скрытое представление h_{t-1} и состояние ячейки c_{t-1} , последовательно вычисляются:

$$\begin{aligned} c'_t &= \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c), \\ i_t &= y(W_{xi}x_t + W_{hi}h_{t-1} + b_i), \\ f_t &= y(W_{xf}x_t + W_{hf}h_{t-1} + b_f), \\ o_t &= y(W_{xo}x_t + W_{ho}h_{t-1} + b_o), \end{aligned} \quad (9)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot c'_t, \quad (9)$$

$$h_t = o_t \odot \tanh(c_t),$$

где c'_t и c_t – ячейки памяти, i_t – входной гейт, f_t – забывающий гейт, o_t – выходной гейт. Принцип построения и объединения LSTM-ячеек в нейронной сети аналогичен описанной модели в [10]. Таким образом, оптимальное количество ячеек было выбрано равным 512. Подробно механика работы LSTM изложена авторами в [11].

Оценка эффективности ННК

Оценка эффективности ННК (нейро-нечеткого контроллера) проводилась по таким критериям, как точность, скорость обучения, количество обрабатываемых правил, скорость принятия решений. Тестирование системы производилось на ПК с процессором Core i7 и 16 Гб оперативной памяти. Тестирование проводилось в программной среде Matlab, также для сравнения были реализованы на языке Java нейросети NNFLC и ANFIS.

Для тестирования была создана модель химической станции, в которой одновременно обрабатывались данные с 45 сенсоров и актуаторов. Моделирование НС ANFIS также проводилось в Toolbox Matlab. Система базы правил первоначально была разработана на основе модифицированного ANFIS-метода, реализованного в базовом пакете программ Matlab. ANFIS показал свою несостоятельность при масштабировании модели, т. к. при количестве параметров больше 70 расчеты было невозможно проводить из-за закрытия программы Matlab. Основные отличия состоят в том, что в Matlab можно задать ограниченное число правил, при превышении числа правил система не справляется с нагрузкой. Проблема ANFIS toolbox состоит в том, что для семи термов и всех сочетаний правил система выполняет переобучение за длительный период времени. Например, для 6804 правил время обучения модели занимает 6 мин на современной машине ЭВМ, что неприемлемо для подобного рода проблем с таким небольшим набором данных. Для более быстрого обучения использовались нейро-нечеткие системы на языке

программирования Java и Python, которые можно легко конфигурировать и обучать [5, 6].

Оценка эффективности гибридной НС на основе ННК и РНС

При обучении модели РНС (рекуррентной нейронной сети) были применены и выведены эмпирическим путем следующие параметры.

1. Параметры алгоритма обучения:

- коэффициент скорости обучения (learningrate) – 0,0001;
- количество «эпох» (циклов) – 30;
- размер батч-пакета (–size) для батч-нормализации – 10.

2. Параметры сети:

- входное количество параметров – 4096;
- количество объектов для каждого изображения (включая функции изображения) – 20;

- количество скрытых слоев LSTM (длительная краткосрочная память) – 512;
- встроенные особенности изображения – 256;
- встроенные особенности объектов – 256;
- количество классов (есть аварийная ситуация или нет) – 2;
- количество кадров в каждом видео – 100.

На основе данных параметров, а также 1730 видеозаписей были проведены обучение и тестирование нейронной сети. Процесс обучения на ПК с Corei7 2,8 ГГц и 30 Гб оперативной памяти занял 8 ч.

Подготовив модель, необходимо адекватно оценить ее качество. Для этого введены следующие понятия:

- TP (True Positive) – истинно-положительный. Классификатор решил, что авария произойдет, и она произошла.
- FP (False Positive) – ложноположи-

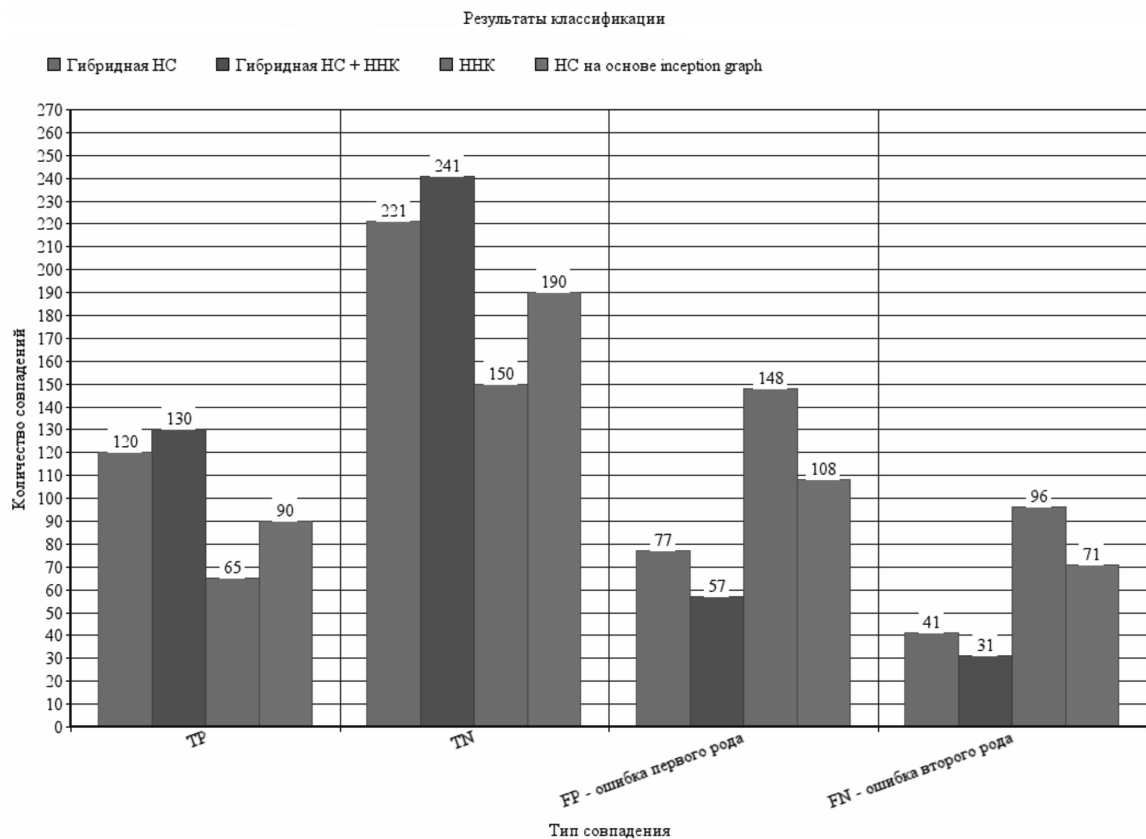


Рис. 4. Сравнение результатов классификации разных моделей

Fig. 4. Comparison of classification results for different models

тельный. Классификатор решил, что авария произойдет, но она не произошла. Это т. н. ошибка первого рода. Она не так страшна, как ошибка второго рода.

$$\alpha = \mathbb{P}\{T \in \Omega_\alpha | H_0\}. \quad (10)$$

• TN (True Negative) – истинно-отрицательный. Классификатор решил, что

Таким образом, получились следующие результаты:

	TP	TN	FP(α)	FN(β)
Гибридная НС	120	221	77	41
Гибридная НС + ННК	130	241	57	31
ННК	65	150	148	96
НС + inception graph	90	190	108	71

Как мы видим, гибридная НС в связке с NNFLC имеет наилучшие результаты: ошибка второго рода в три раза меньше, чем в случае обычной модели на основе NNFLC, и более чем в два раза лучше, чем нейронная сеть на основе стандартной модели для определения автомобилей.

Основные научные и практические результаты

Обучение и тестирование модели гибридной сети проводилось на основе записей с видеорегистраторов от компании VSLab в Тайване, всего было предоставлено 1730 записей по сто кадров в каждом с разрешением 720 p [4]. Данный набор видеозаписей и послужил данными для апробации модели.

На рис. 5 приведены графики, полученные в результате тренировки модели и те-

авария не произойдет, и она произошла.

$$\beta(H_1) = \mathbb{P}\{T \notin \Omega_\alpha | H_1\}, \quad (11)$$

где Ω_α – критическое множество наименее вероятных значений T ; H_0 – нулевая гипотеза; H_1 – альтернативная гипотеза (не H_0).

Сравнение результатов классификации разных моделей показано на рис. 4.

стового прогона на основе тестовой выборки. Первый график показывает зависимость Precision v.s. Recall (PR). Второй график показывает среднее время до момента аварии Time-to-Accident для различной полноты Recall (ToAR). Как видно из графика, метод способен в среднем предсказывать аварию за 0,38 с до того момента, когда она произойдет, с вероятностью равной 0,62 и 80 % полнотой (отношение верно угаданных объектов класса ко всем представителям этого класса).

В ходе работы над проектом разработан прототип мобильной системы на основе обученной модели гибридной сети вместе с ННК (рис. 6). В роли вычислительной машины выступает мини-компьютер Raspberry Pi 3 с 1 Гб оперативной памяти и ARM-процессором 1,5 ГГц.

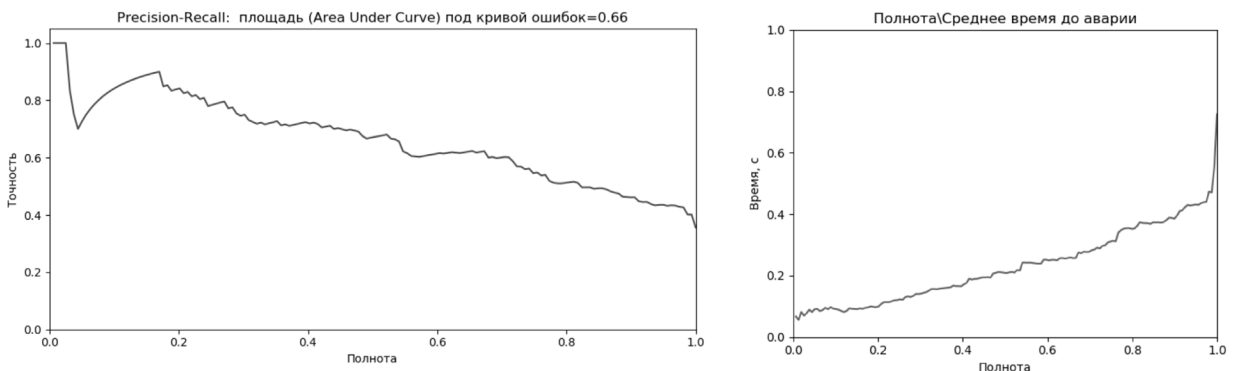


Рис. 5. График зависимости Precision v.s. Recall (PR)

Fig. 5. Dependency graph Precision v.s. Recall (PR)

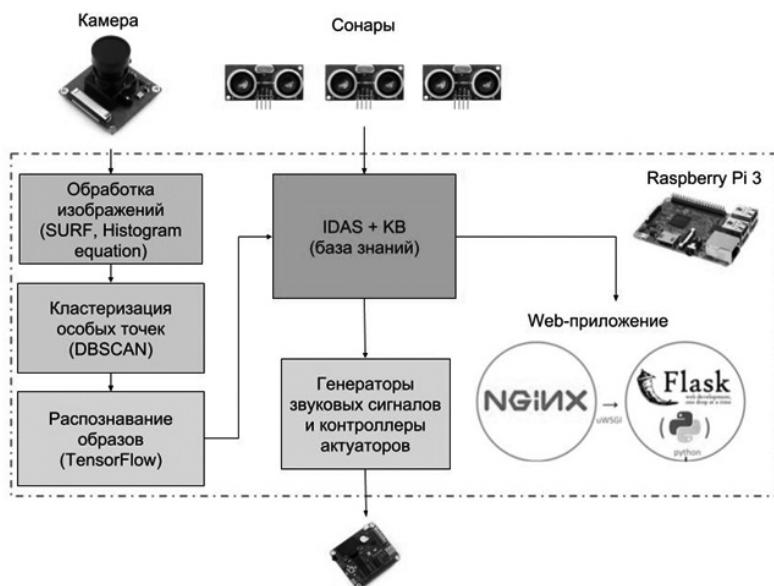


Рис. 6. Прототип мобильной системы прогноза нештатных ситуаций на основе мини-компьютера Raspberry Pi 3

Fig. 6. The prototype of a mobile system for predicting emergency situations based on the Raspberry Pi 3 mini-computer

Заключение

Разработаны программные модули обучения, тестирования и применения иерархической рекуррентной нейронной сети для предсказания аварий при помощи алгоритмов и моделей глубокого обучения. Проведено комплексное имитационное моделирование различных аварийных ситуаций на тестовых данных – видеозаписей с видеорегистраторов. Разработан прототип

для считывания видеопотока и обработки на мобильном компьютере Raspberry Pi 3. Для данного исследования был подготовлен набор данных для обучения нейронной сети, состоящий из 1730 видеозаписей, и разработана система контроля дорожной ситуации на основе гибридной модели нейронной сети. Разработанная модель показала приемлемые результаты на тестовых данных, но требует большего количества данных и дальнейшего улучшения.

СПИСОК ЛИТЕРАТУРЫ

1. Google Inc.: Google self-driving car project monthly report. 2015 // URL: <https://static.googleusercontent.com/media/www.google.com/ru//selfdrivingcar/files/reports/report-0515.pdf>
2. N. Highway Traffic Safety Administration: 2012 motor vehicle crashes: overview // URL: <https://www.cdc.gov/vitalsigns/pdf/2016-07-vitalsigns.pdf>
3. **Малыхина М.П., Бегман Ю.В.** Оценка эффективности гибридации интеллектуальных методов на примере нейросетевой экспертной системы на основе прецедентов // Научный журнал КубГАУ. Краснодар: КубГАУ, 2013. № 86 (02). С. 253–262.
4. **Fu-Hsiang Chan, Yu-Ting Chen, Yu Xiang, Min Sun.** Anticipating Accidents in Dashcam Videos. ACCV, 2016.
5. **Fedorov A., Shkodyrev V., Zobnin S.** Knowledge based planning framework for intelligent distributed manufacturing systems // Lecture Notes in Computer Science. 2015. Vol. 9141. Pp. 300–307.
6. **Fedorov A., Zobnin S., Shkodyrev V., Goloschchapov E., Potekhin V.** Fuzzy decision support in distributed v2v systems // Source: Annals of DAAAM & Proceedings. 2015. Vol. 26. Iss. 1. Pp. 0097–0103.
7. **Honglei Ren, You Song, Jingwen Wang, Yucheng Hu, Jinzhi Lei.** A deep learning approach to the citywide traffic. Accident risk prediction // arXiv: 1710.09543v2 [cs.CY]. 15 Apr. 2018.
8. **Ashesh Jain, Avi Singh, Hema S Koppula, Shane Soh, Ashutosh Saxena.** Recurrent neural networks for driver activity anticipation via sensory-

fusion architecture // arXiv: 1509.05016 [cs.CV]. 16 Sep. 2015.

9. **Navaneeth Bodla, Bharat Singh, Rama Chellappa, Larry S. Davis.** Improving object detection with one line of code // arXiv: 1704.04503v2 [cs.CV]. 8 Aug. 2017.

10. **Joe Yue-Hei Ng, Matthew Hausknecht,**

Статья поступила в редакцию 23.06.2018.

Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, George Toderici. Beyond short snippets: deep networks for video classification // arXiv: 1503.08909v2 [cs.CV]. 13 Apr. 2015.

11. **Николенко С., Кадурин А., Архангельская Е.** Глубокое обучение. СПб.: ИД «Питер», 2017.

REFERENCES

1. Google Inc.: Google self-driving car project monthly report. 2015. Available: <https://static.googleusercontent.com/media/www.google.com/ru//selfdrivingcar/files/reports/report-0515.pdf>

2. N. Highway Traffic Safety Administration: 2012 motor vehicle crashes: overview. Available: <https://www.cdc.gov/vitalsigns/pdf/2016-07-vitalsigns.pdf>

3. **Malykhina M.P., Begman Yu.V.** Evaluation of efficiency of hybridization of the intelligent methods on the example of the neural network expert system based on precedents. *Scientific Journal of KubSAU*. Krasnodar: KubGAU Publ., 2013, No. 86 (02), Pp. 253–262. (rus)

4. **Fu-Hsiang Chan, Yu-Ting Chen, Yu Xiang, Min Sun.** *Anticipating Accidents in Dashcam Videos*. ACCV, 2016.

5. **Fedorov A., Shkodyrev V., Zobnin S.** Knowledge based planning framework for intelligent distributed manufacturing systems. *Lecture Notes in Computer Science*, 2015, Vol. 9141, Pp. 300–307.

6. **Fedorov A., Zobnin S., Shkodyrev V., Goloschchapov E., Potekhin V.** Fuzzy decision

support in distributed v2v systems. *Source: Annals of DAAAM & Proceedings*, 2015, Vol. 26, Iss. 1, Pp. 0097–0103.

7. **Honglei Ren, You Song, Jingwen Wang, Yucheng Hu, Jinzhi Lei.** A deep learning approach to the citywide traffic. Accident risk prediction, *arXiv:1710.09543v2 [cs.CV]*. 15 Apr. 2018.

8. **Ashesh Jain, Avi Singh, Hema S Koppula, Shane Soh, Ashutosh Saxena.** Recurrent neural networks for driver activity anticipation via sensory-fusion architecture, *arXiv: 1509.05016 [cs.CV]*, 16 Sep. 2015.

9. **Navaneeth Bodla, Bharat Singh, Rama Chellappa, Larry S. Davis.** Improving object detection with one line of code, *arXiv:1704.04503v2 [cs.CV]*, 8 Aug. 2017.

10. **Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, George Toderici.** Beyond short snippets: deep networks for video classification, *arXiv:1503.08909v2 [cs.CV]*, 13 Apr. 2015.

11. **Nikolenko S., Kadurin A., Arkhangelskaya E.** *Deep Learning*. St. Petersburg: Piter Publ., 2017. (rus)

Received 23.06.2018.

СВЕДЕНИЯ ОБ АВТОРАХ / THE AUTHORS

ФЕДОРОВ Александр Викторович
FEDOROV Aleksandr V.
E-mail: alexff91@gmail.com

ШКОДЫРЕВ Вячеслав Петрович
SHKODYREV Viacheslav P.
E-mail: shkodyrev@mail.ru

БАРСУКОВ Никита Дмитриевич
BARSUKOV Nikita D.
E-mail: nikbars1997@gmail.com