

References

1. **Buslenko N. P.** *Modelirovanie slozhnykh sistem* (Modeling complex systems), Moscow, Nauka, 1978, 400 p. (in Russian).
2. **Sharakhshane A. S., Zheleznov I. G., Ivnickij V. A.** *Slozhnye sistemy* (Complex systems), Moscow, Vysshaya shkola, 1977, 246 p. (in Russian).
3. **Sharakhshane A. S., Haleckij A. K., Morozov I. A.** *Ocenka harakteristik slozhnykh avtomatizirovannykh sistem* (Evaluation of the characteristics of complex automated systems), Moscow, Mashinostroenie, 1993, 272 p. (in Russian).
4. **Grigorenko V. M., Mel'nik D. I.** *Osnovnye problemy modelirovaniya sistem i sredstv vozdushno-kosmicheskoy oborony na osnove perspektivnykh informacionnykh tehnologij* (The main problems of modeling systems and means of aerospace defense based on advanced information technologies), *Voennaya mysl'*, 2015, no. 6, pp. 62–67 (in Russian).
5. **Barskij A. B., Zhelenkov B. V., Shamenkov N. A.** *Vysokonadyyzhnyye upravlyayushhie komp'yuternye seti s nejrosetevoj adaptivnoj marshrutizaciej na baze perspektivnykh modelej serii "El'brus"* (Highly reliable control computer networks with neural network adaptive routing based on promising models of the Elbrus series), *Informacionnye Tehnologii*, 2018, vol. 24, no. 5, pp. 291–299 (in Russian).
6. **Barskij A. B.** *Nejrosetevye metody optimizacii reshenij* (Neural network methods for optimizing solutions), SPb, IC "Intermediya", 2016, 312 p. (in Russian).

УДК 004.023, 004.8

DOI: 10.17587/it.25.116-127

О. С. Амосов, д-р техн. наук, проф., гл. науч. сотр., e-mail: osal18@yandex.ru,
С. Г. Амосова, канд. техн. наук, доц., ст. науч. сотр., e-mail: amosovasg@yandex.ru,
Институт проблем управления им. В. А. Трапезникова РАН,
Ю. С. Иванов, канд. техн. наук, доц., e-mail: ivanov_ys@icloud.com,
С. В. Жиганов, аспирант, e-mail: zhiganov@knastu.ru,
Комсомольский-на-Амуре государственный университет

Моделирование интеллектуальной системы контроля и управления доступом транспортных средств с использованием глубоких нейронных сетей¹

Разработана математическая модель интеллектуальной системы контроля и управления доступом транспортных средств на некоторую территорию. Предложено выполнение локализации транспортного средства с помощью глубокой нейронной сети YOLO, что позволяет дополнительно определить тип объекта доступа. Решение задачи локализации и распознавания номерного знака основано на композиции традиционных методов обработки изображений и двухпроходной классификации, выполняемой модифицированной архитектурой сверточной нейронной сети MobileNet. Экспериментально доказано, что применение разработанного подхода дает процент правильных распознаваний номерных знаков на видеопотоке не ниже 96 % в зависимости от внешних условий. Комплекс программ реализован на языке Python.

Ключевые слова: математическая модель, система контроля и управления доступом, транспортное средство, локализация, распознавание, сверточная нейронная сеть, MobileNet

Введение

Современная система контроля и управления доступом (СКУД) транспортных средств (ТС) на охраняемую территорию должна обеспечивать их идентификацию и ограничение доступа. Основные визуальные признаки, описывающие объект доступа (ОД) — автомобиль:

цвет и тип кузова; марка, модель; категория и номерной знак (НЗ).

В работе [1] в качестве идентификационных признаков ТС предлагается использовать информацию о цвете автомобиля, однако такой подход затруднен в вечернее и ночное время, когда камера наблюдения переходит в черно-белый режим.

Авторами [2] предлагается метод распознавания марки и модели в условиях ограниченного освещения в ночное время. Но такой метод не может быть применен в СКУД как основной классификатор.

¹ Работа выполнена при поддержке Минобрнауки России научного проекта — госзадания в рамках проектной части № 2.1898.2017/4.6 "Создание математического и алгоритмического обеспечения интеллектуальной информационно-телекоммуникационной системы безопасности вуза".

В последнее время хорошие результаты в оптическом распознавании образов достигаются с помощью использования искусственных нейронных сетей (НС) [3, 4]. В работе [5] приводится сравнение различных нейросетевых алгоритмов для задачи распознавания рукописных цифр.

Авторами статьи [6] предлагается использование каскадной архитектуры из сверточной НС и сети с долго- и кратковременной памятью (long short-term memory, LSTM). Основным преимуществом такого подхода является отсутствие этапа сегментации, но при этом возникают дополнительные требования к вычислительным ресурсам, в частности к объему памяти.

В работе [7] предлагалось использовать три глубоких нейросети (Deep Neural Network) для детекции НЗ, локализации символов и их распознавания, однако используемые архитектуры также не позволяли работать в режиме реального времени (РВ).

Основным недостатком предлагаемых ранее систем являются высокие требования к вычислительным ресурсам, что затрудняет использование их в РВ для отслеживания нештатных ситуаций. Из последних достижений в области глубокого обучения стоит отметить архитектуру НС MobileNet [8], которая обладает низкими требованиями к вычислительным ресурсам, однако для решения задачи распознавания НЗ ТС ранее не применялась. Поэтому наиболее перспективным представляется использование глубоких НС [8, 9] для построения систем распознавания НЗ. При этом существует необходимость разработки подхода, который позволит обрабатывать каждый кадр видеопотока в РВ в штатной и нештатной ситуациях на основе совмещения технологий глубокого обучения и традиционных методов обработки изображений. По этой причине в данной статье предлагается модель интеллектуальной СКУД ТС на основе нового алгоритма локализации и распознавания НЗ ТС путем двухпроходной классификации с использованием сверточной НС с модифицированной архитектурой, что позволило получить высокую точность распознавания без потери скорости в режиме РВ.

1. Постановка задачи

Необходимо разработать математическую модель интеллектуальной СКУД ТС и применить разработанную модель для формализо-

ванного описания возможных штатных и нештатных ситуаций. Для распознавания ситуаций применить методы компьютерного зрения на базе сверточных НС. Требуется разработать комплекс программ в среде Python и провести натурный эксперимент с кадрами, поступающими с камеры наблюдения (КН) системы безопасности предприятия.

Математическая формулировка задачи контроля и управления доступом. В работе [10] нами предлагается структурная схема интеллектуальной СКУД на основе кибернетического подхода для задачи доступа ТС на территорию организации. В этой схеме контрольно-пропускной пункт (КПП) состоит из устройства управления (УУ), управляемого преграждающего устройства (УПУ), камеры наблюдения и устройств внешней подсветки (УВП). Необходимо при появлении на КПП объекта доступа разрешить или запретить ему доступ на охраняемую территорию с учетом воздействия внешней среды (ВС) и записать событие в базу данных (БД). Тогда математическая формулировка задачи контроля и управления доступом выглядит так.

Пусть имеются вектор состояний СКУД $\mathbf{x} = (\mathbf{x}_{\text{ОД}}^T, \mathbf{x}_{\text{КН}}^T, x_{\text{УПУ}}, x_{\text{УВП}}, \mathbf{x}_{\text{ВС}}^T)^T$, где $\mathbf{x}_{\text{ОД}}^T$ — вектор состояний объекта доступа; $\mathbf{x}_{\text{КН}}^T$ — вектор состояний камеры наблюдения; $x_{\text{УПУ}}$ — состояние управляемого преграждающего устройства; $x_{\text{УВП}}$ — состояние устройства внешней подсветки; $\mathbf{x}_{\text{ВС}}^T$ — вектор состояний внешней среды; и вектор выходных сигналов (измерений) для СКУД $\mathbf{y} = (\mathbf{y}_{\text{ОД}}^T, y_{\text{НВ}}, y_{\text{УПУ}}, y_{\text{УВП}}, \mathbf{y}_{\text{ВС}}^T)^T$, где $y_{\text{НВ}}$ — наличие кадра $\mathbf{I}^t = \mathbf{V}(t)$ с камеры наблюдения видеопотока \mathbf{V} , представленного многомерным массивом; $\mathbf{y}_{\text{ОД}}^T$ — сигнал с КН, причем $\mathbf{y}_{\text{ОД}}^T = \mathbf{I}^t$; $y_{\text{УПУ}}$ — сигнал с УПУ; $y_{\text{УВП}}$ — сигнал с УВП; $\mathbf{y}_{\text{ВС}}^T$ — вектор сигналов с датчиков, фиксирующих состояние ВС. Необходимо по результатам наблюдений \mathbf{y} состояний ОД \mathbf{x} выработать оценки состояний $\tilde{\mathbf{x}}$ и необходимые управляющие воздействия \mathbf{u} на объект управления, $\mathbf{u} = (u_{\text{УПУ}}, u_{\text{УВП}}, u_{\text{БД}})^T$, где $u_{\text{УПУ}}$ — сигнал на открытие или закрытие управляемого преграждающего устройства, $u_{\text{УВП}}$ — сигнал на устройство внешней подсветки, $u_{\text{БД}}$ — сигнал на запись события в базу данных. Каждый из элементов принадлежит соответствующему множеству: состояний $\mathbf{x} \in X$, их оценок $\tilde{\mathbf{x}} \in \tilde{X}$, видеопотоков $\mathbf{V} \in V$, выходных сигналов $\mathbf{y} \in Y$ и управлений $\mathbf{u} \in U$. T — множество моментов времени, $t \in T$ — произвольный момент времени. Функциональные зависимости на X , Y и U могут быть представлены в виде отображений

фильтров для извлечения признаков. Выделенные признаки записываются в состояние системы объекта доступа $\tilde{\mathbf{x}}_{\text{Од}} = (\tilde{x}_{\text{НОд}}, \tilde{x}_{\text{ТОд}}, \tilde{x}_{\text{НКП}}, \tilde{x}_{\text{СКП}}, \tilde{\mathbf{x}}_{\text{ЗКП}}^T)^T$, по которым и определяется штатная или нештатная ситуация в элементе сравнения, реализованном в виде нечеткой экспертной системы.

2. Решение задачи локализации и распознавания транспортного средства с использованием сверточных нейронных сетей

1. Выполняется локализация ТС на кадре видеопотока. Данную подзадачу предлагается решать глубокой НС YOLO (You Only Look Once) с предобученной архитектурой для детекции объектов [14]. Результатом работы является область интереса (Region of interest, ROI), содержащая изображение объекта.

2. Выполняется локализация НЗ в области интереса. Для решения данной задачи используется классический алгоритм Виолы — Джонса.

3. Выполняется сегментация и распознавание символов НЗ. Данная задача решается с использованием архитектуры глубокой НС MobileNet [8], которая была переобучена и модифицирована под решаемую задачу (рис. 2).

2.1. Локализация транспортного средства на кадре видеопотока

Пусть имеется кадр видеопотока \mathbf{I}^t размером $i \times j$, где t — номер текущего кадра. Областью интереса первого уровня $\mathbf{R}_o^{(1)}$ назовем прямоугольник, описанный вокруг полигона o -го объекта, найденного на исходном изображении \mathbf{I}^t . Тогда необходимо построить алгоритм, отображающий $\mathbf{f}^{yolo} : \mathbf{I}^t \rightarrow \mathbf{G}$, где \mathbf{G} — массив элементов, состоящий из o строк, причем каждая строка содержит x_o, y_o — координаты центра o -го объекта, w_o, h_o — его размеры и $\tilde{x}_{\text{ТОд}_o}$ — его тип, т. е. $\mathbf{G} = (x_o, y_o, w_o, h_o, \tilde{x}_{\text{ТОд}_o})$, где $o = \overline{1, n}$, n — число объектов в кадре видеопотока. Область интереса $\mathbf{R}_o^{(1)}$ является подматрицей матрицы

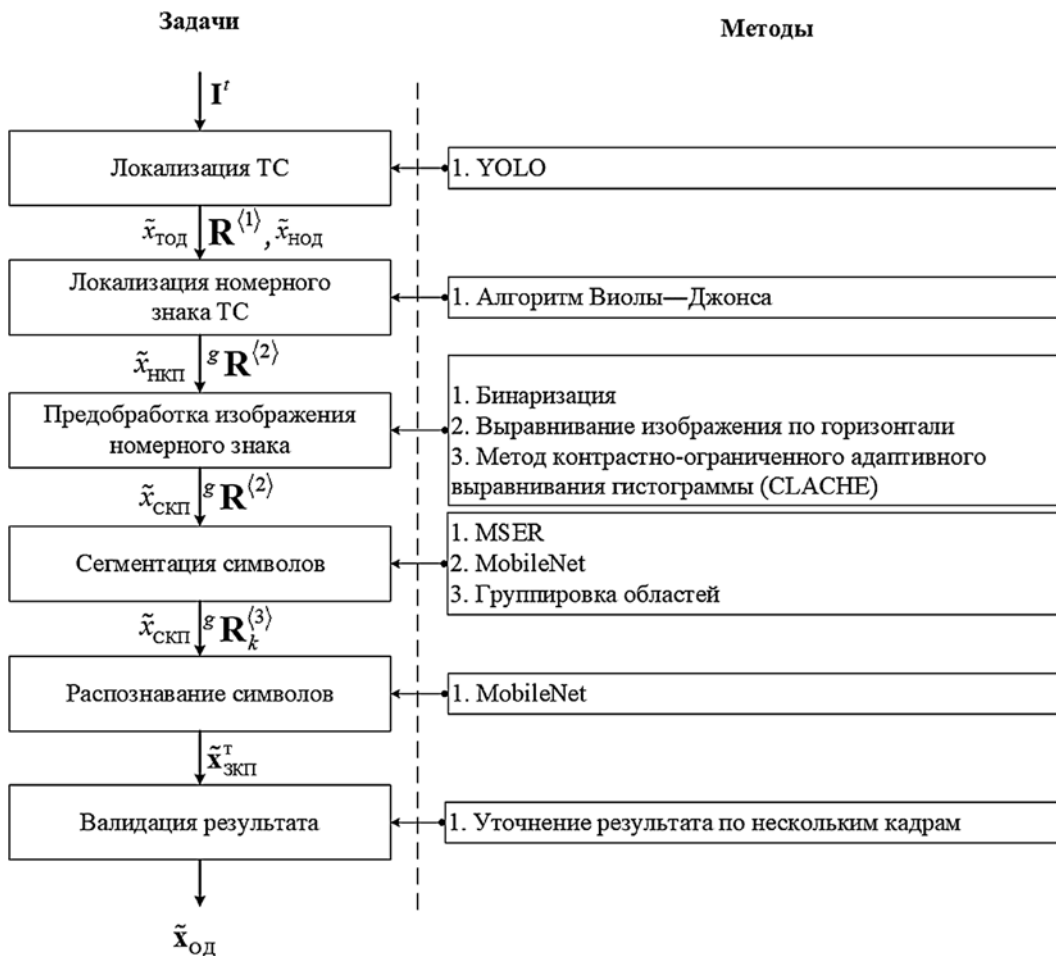


Рис. 2. Алгоритм локализации и распознавания номерного знака

I^t , вырезанной по координатам $(x_o - h_o, y_o - w_o)$, $(x_o + h_o, y_o + w_o)$, причем:

$$\tilde{x}_{\text{НОД}_o} = \begin{cases} 1, & \text{если } \mathbf{R}_o^{(1)} \neq \emptyset, \\ 0 & \text{иначе.} \end{cases}$$

В качестве f^{yolo} алгоритма локализации ТС предлагается использовать предобученную глубокую НС YOLO [14], способную определять местоположение и различать между собой 20 типов объектов на изображении. Архитектура сети основана на модели GoogLeNet [15] и состоит из 24 чередующихся слоев свертки (convolution), подвыборки (pooling, subsampling) и двух последних полносвязных (fully connected, dense) слоев. Сверточная НС представляет собой последовательный набор элементарных матричных операций, применяемых к многомерным массивам (тензорам).

На вход НС YOLO подается матрица изображения I^t размером 448×448 пикселей, которая разбивается на $q \times q$ одинаковых подматриц. Каждая подматрица вырезается из изображения I^t по следующим координатам: $((i/q)m - i/q, (j/q)n - j/q)$, $((i/q)m, (j/q)n)$, где $m, n = 1, q$, а i, j — размеры изображения. Выполняется поиск объектов сканирующим окном с изменяющимся шагом и размерами. Если координаты центра найденного объекта попадают в подматрицу $I_{((i/q)m - i/q, (j/q)n - j/q), ((i/q)m, (j/q)n)}^t$, то она "отвечает" за обнаружение этого объекта. Выходом сети является многомерный массив \mathbf{Q} , размером $m \times n$. В каждой ячейке (m, n) массива \mathbf{Q} содержатся два вектора $\mathbf{b}_1, \mathbf{b}_2$ размером 1×5 и вектор \mathbf{c} . Каждый из векторов $\mathbf{b}_1, \mathbf{b}_2$ содержит предполагаемые x, y — координаты центра объекта относительно подматрицы, ширину w и высоту h объекта относительно изображения I^t , а также степень доверия $P = P(o) \cdot IoU_{pred}^{truth}$, где IoU (Intersection over Union) — операция отношения площади пересечения прямоугольников к сумме их площадей [16], $P(o)$ — вероятность появления o -го объекта. Вектор \mathbf{c} содержит вероятности отнесения объекта к каждому из 20 классов, т. е. $\mathbf{c}_o = P(\tilde{x}_{\text{ТОД}_o} | o)$ для \mathbf{b}_1 и \mathbf{b}_2 , где $i = \overline{1, 20}$. В результате мы получаем массив \mathbf{G} , содержащий детектированные объекты, их координаты и метки классов. Если $\tilde{x}_{\text{ТОД}_o} = \{1, 2, 3\}$, т. е. детектирован "автомобиль", "грузовик" или "автобус", то формируется область интереса $\mathbf{R}_o^{(1)}$, в противном случае $\tilde{x}_{\text{НКП}_o} = 0$, и проезд запрещен $\mathbf{u} = (0, u_{\text{БД}})$.

2.2. Локализация номерного знака в области интереса

Пусть имеется область интереса $\mathbf{R}^{(1)}$, содержащая изображение ТС. Необходимо получить массив $\mathbf{L} = \{\mathbf{R}^{(2)}, x, y, w, h\}$, содержащий изображение НЗ $\mathbf{R}^{(2)}$, а также его координаты x, y относительно исходного изображения I^t и размеры $w \times h$. Задача локализации НЗ ТС сводится к нахождению следующего отображения $f^{haar}: \mathbf{R}^{(1)} \rightarrow \mathbf{L}$, причем

$$\tilde{x}_{\text{НКП}} = \begin{cases} 1, & \text{если } \mathbf{R}^{(2)} \neq \emptyset, \\ 0 & \text{иначе.} \end{cases}$$

В качестве алгоритма локализации используется классический алгоритм Виолы — Джонса [17] с каскадным классификатором "haarcascade_russian_plate_number.xml", представленным в библиотеке OpenCV [18].

Результатом работы алгоритма является матрица ${}^g\mathbf{R}^{(2)}$, содержащая полутоновое изображение НЗ, где каждый пиксель представлен значением яркости от 0 до 255. После выделения необходимо выполнить предобработку, которая проводится в несколько этапов.

1. ${}^g\mathbf{R}^{(2)}$ переводится в черно-белое представление с заданным порогом th :

$${}^{bw}\mathbf{R}_{i,j}^{(2)} = \begin{cases} 1 & \text{при } \mathbf{R}_{i,j}^{(2)} \geq th, \\ 0 & \text{при } \mathbf{R}_{i,j}^{(2)} < th. \end{cases}$$

2. Выполняются последовательные повороты изображения ${}^{bw}\mathbf{R}^{(2)}$ на произвольный угол $\theta = -10, \dots, 10^\circ$ с шагом $0,1^\circ$ путем умножения матрицы поворота на вектор-столбец, с координатами точки, т. е.

$${}^{r\theta}\mathbf{R}_{i^*,j^*}^{(2)} = {}^{bw}\mathbf{R}_{i,j}^{(2)},$$

$$i^* = w/2 + (i - w/2) \cos \theta - (j - h/2) \sin \theta.$$

3. Для каждого изображения ${}^{r\theta}\mathbf{R}_{i^*,j^*}^{(2)}$ необходимо вычислить нижнюю границу номерного знака и найти угол θ^* , при котором она будет минимальной:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \rho \left({}^{r\theta}\mathbf{R}_{i^*,j^*}^{(2)} \right) = \begin{cases} i^*, & \text{если } m_{i^*} = m_{i^*-1}, \\ h & \text{иначе,} \end{cases}$$

где $m_{i^*} = \sum_{j^*=0}^w {}^{r\theta}\mathbf{R}_{i^*,j^*}^{(2)}$ — сумма элементов i^* -й строки.

4. Изображение ${}^g\mathbf{R}^{(2)}$ поворачивается на угол θ^* и обрезаются все строки и столбцы, для которых выполняются условия $\sum_{j=0}^w {}^g\mathbf{R}_{i,j}^{(2)} = 0$ и $\sum_{i=0}^h {}^g\mathbf{R}_{i,j}^{(2)} = 0$.

5. К изображению ${}^g\mathbf{R}^{(2)}$ применяется метод контрастно-ограниченного адаптивного выравнивания гистограммы (Contrast limited adaptive histogram equalization, CLACHE) [19], с помощью которого анализируются и выравниваются гистограммы распределения значений интенсивностей пикселей локальных областей изображения:

$${}^g\mathbf{R}_{i,j}^{(2)} = (\max({}^g\mathbf{R}^{(2)}) - \min({}^g\mathbf{R}^{(2)}))\delta + \min({}^g\mathbf{R}^{(2)}),$$

где δ — функция распределения пикселей. Функция δ может быть выражена как

$$\delta({}^g\mathbf{R}^{(2)}) = \sum_{i=0}^{255} \frac{n_i}{N},$$

где n_i — число пикселей области, значение яркости которых равно i , N — общее число пикселей области. Адаптивное выравнивание гистограммы, в отличие от глобального, анализирует небольшие участки изображения, что позволяет усилить локальный контраст, не делая предположений о распределении яркостей на всем изображении. Такой подход более приемлем при анализе кадра, полученного с уличных камер наблюдения.

2.3. Сегментация и распознавание символов номерного знака

Алгоритм сегментации и распознавания символов выполняется в несколько проходов. Пусть имеется изображение ${}^g\mathbf{R}^{(2)}$, содержащее НЗ ТС. Необходимо построить алгоритм для сегментации (разбиения) изображения ${}^g\mathbf{R}^{(2)}$ на p отдельных областей $\mathbf{R}_p^{(3)}$, каждая из которых содержит символ НЗ: $\mathbf{f}^{mser}: {}^g\mathbf{R}^{(2)} \rightarrow \mathbf{S}$, где $\mathbf{S} = (\mathbf{R}_1^{(3)}, \mathbf{R}_2^{(3)}, \dots, \mathbf{R}_k^{(3)})$ — массив, содержащий матрицы изображений символов НЗ, а $k = 1, 9$ — максимальное число символов в НЗ.

Для выделения символов НЗ применен алгоритм максимально стабильных экстремальных регионов (Maximally stable extremal regions, MSER) [20].

Создается массив $\mathbf{J} = ({}^0\mathbf{R}^{(2)}, \dots, {}^{255}\mathbf{R}^{(2)})$, состоящий из бинарных матриц, полученных из полутонового изображения

$${}^{bw}\mathbf{R}_{i,j}^{(2)} = \begin{cases} 1, & \text{если } {}^g\mathbf{R}_{i,j}^{(2)} = bw, \\ 0 & \text{иначе,} \end{cases}$$

где $bw = \overline{0, 255}$.

Белые области в бинарных матрицах называются областями экстремума [20]. Если область не изменялась на r матрицах из массива \mathbf{J} , то она называется максимально устойчивой областью экстремума.

Результатом работы алгоритма MSER является множество матриц-кандидатов в символы $\mathbf{R}_p^{(3)}$, где $p \geq k$, причем каждый символ может принадлежать нескольким областям (рис. 3, см. третью сторону обложки).

Необходимо отфильтровать полученные изображения на принадлежность к тексту или не к тексту, т. е. решить задачу классификации.

Пусть имеется изображение $\mathbf{R}_p^{(3)}$, содержащее предполагаемый символ номерного знака. $\mathbf{V}^{MNet} = \{\beta_1, \beta_2, \dots, \beta_9, \beta_{10}, \dots, \beta_{23}\}$ — конечное множество классов возможных символов, составляющих номерной знак, причем для нераспознанных символов (шумы, помехи) зарезервирован класс β_{13} . Тогда функцией классификации (классификатором) является сопоставление каждой $\mathbf{R}_p^{(3)}$ классу β_c , где $c = \overline{1, 23}$ т. е. $\mathbf{f}^{MNet}: \mathbf{R}_p^{(3)} \rightarrow \beta_c$.

Однако на данном этапе необходимо решить задачу бинарной классификации, для этого множество классов \mathbf{V}^{MNet} сводится к множеству \mathbf{V}^{MNet*} следующим образом:

$$\beta_c^* = \begin{cases} -1, & \text{если } \beta_c = 13, \\ +1 & \text{иначе,} \end{cases}$$

где β_c^* — элемент множества $\mathbf{V}^{MNet*} = \{-1, +1\}$.

В качестве классификатора \mathbf{f}^{MNet} мы предлагаем использовать модифицированную архитектуру глубокой искусственной НС MobileNet.

Среди существующих основных глубоких архитектур, таких как GoogleNet, VGG16 и Inception v3 [21], архитектура MobileNet показывает высокую точность классификации при меньшем числе слоев, что позволяет ее использовать при работе с потоковым видео [22].

Базовая архитектура MobileNet [8] состоит из 26 слоев и предназначена для отнесения изображений размером 224×224 пикселей к одному из 1000 классов ImageNet [23].

В базовой архитектуре два последних слоя — это слой подвыборки с функцией $s^{AvgPool}$ — операцией вычисления среднего значения по всем k каналам предыдущего слоя \mathbf{h}^{l-1} (Global Average Pooling); выходной слой свертки с ядром размером (1,1), функцией активации Softmax и выходным вектором \mathbf{h}'_{conv} размером 1×1000 .

Для решения задачи классификации символов номерного знака нами предлагается модифицировать базовую архитектуру следующим образом: 1) понизить размерность входного слоя до 50×50 для соответствия входной матрице изображения $\mathbf{R}_p^{(3)}$; 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 $2 \times 2 \times 1024$, а в качестве функции активации используется функция ReLU (Rectified linear unit), которая обнуляет отрицательные элементы входного вектора;
- для лучшего выделения признаков добавляется второй скрытый полносвязный слой размером $2 \times 2 \times 128$ нейронов. Экспериментально нами было установлено, что два скрытых полносвязных слоя показывают лучший результат в решении задачи по сравнению с одним скрытым полносвязным слоем;
- для приведения выходного массива $2 \times 2 \times 128$ к одномерному вектору 1×512 добавляется слой выравнивания (Flatten): $\mathbf{h}'_{flat} = \text{vec}(\mathbf{h}^{l-1})$, где vec — операция векторизации, т.е. линейного преобразования матрицы в вектор-столбец;
- для приведения к задаче отнесения к одному из β_c классов добавляется последний слой из 23 нейронов и функцией активации Softmax.

Далее необходимо переобучить полученную модифицированную архитектуру без использования технологии "переноса знаний" [24], так как новый набор данных существенно отличается от набора данных ImageNet. В качестве метода оптимизации обучения предлагается использовать метод среднеквадратического распространения (Root Mean Square Propagation, RMSProp) [25]. Метод заключается в том, чтобы разделить скорость обучения для нейронов по среднему значению последних градиентов для этого нейрона, что позволяет адаптировать скорость обучения по каждому из параметров:

$$r_\tau = (1 - \gamma)f'(\vartheta_\tau)^2 + \gamma r_{\tau-1};$$

$$\vartheta_{\tau+1} = \vartheta_\tau - (a/\sqrt{r_\tau})f'(\vartheta_\tau),$$

где $f'(\vartheta_\tau)$ — производная от потерь по отношению к параметрам времени на эпохе τ , a — скорость обучения, γ — параметр забывания (forget).

Предложенная модифицированная архитектура НС \mathbf{f}^{MNet} была обучена на графическом процессоре Nvidia GeForce 1080Ti. Для обучения использовалось обучающее множество, содержащее 14 936 размеченных изображений символов номерных знаков. Общее время обучения составило 4 ч, а точность на тестирующем множестве — 98 %, что свидетельствует о возможности применения ее на реальных данных.

После классификации символов НЗ все области $\mathbf{R}_p^{(3)}$, отнесенные к β_c^* , удаляются. Для оставшихся $\mathbf{R}_p^{(3)}$ выполняется алгоритм группировки областей (Group Rectangles) [18].

Необходимо определить, находятся ли две области в одном кластере:

$$\delta = 0,5\varepsilon \min \left(w^{\mathbf{R}_p^{(3)}}, w^{\mathbf{R}_{p+1}^{(3)}} \right) + \min \left(h^{\mathbf{R}_p^{(3)}}, h^{\mathbf{R}_{p+1}^{(3)}} \right),$$

где $w^{\mathbf{R}_p^{(3)}}, w^{\mathbf{R}_{p+1}^{(3)}}, h^{\mathbf{R}_p^{(3)}}, h^{\mathbf{R}_{p+1}^{(3)}}$ — размеры прямоугольников, $\varepsilon = 0,2$ — коэффициент перекрытия.

Прямоугольники объединяются, если смещение меньше или равно δ :

$$\begin{aligned} x^{\mathbf{R}_p^{(3)}} - x^{\mathbf{R}_{p+1}^{(3)}} &\leq \delta; \quad y^{\mathbf{R}_p^{(3)}} - y^{\mathbf{R}_{p+1}^{(3)}} &\leq \delta; \\ \left(x^{\mathbf{R}_p^{(3)}} + w^{\mathbf{R}_p^{(3)}} \right) - \left(x^{\mathbf{R}_{p+1}^{(3)}} + w^{\mathbf{R}_{p+1}^{(3)}} \right) &\leq \delta; \\ \left(y^{\mathbf{R}_p^{(3)}} + h^{\mathbf{R}_p^{(3)}} \right) - \left(y^{\mathbf{R}_{p+1}^{(3)}} + h^{\mathbf{R}_{p+1}^{(3)}} \right) &\leq \delta. \end{aligned}$$

Удаляются все прямоугольники $\mathbf{R}_p^{(3)}$, вписанные в прямоугольник $\mathbf{R}_{p+1}^{(3)}$, если выполняются следующие условия:

$$\begin{aligned} x^{\mathbf{R}_p^{(3)}} &\geq \left(x^{\mathbf{R}_{p+1}^{(3)}} - \Delta x \right); \quad y^{\mathbf{R}_p^{(3)}} &\geq \left(y^{\mathbf{R}_{p+1}^{(3)}} - \Delta y \right); \\ \left(x^{\mathbf{R}_p^{(3)}} + w^{\mathbf{R}_p^{(3)}} \right) &\leq \left(x^{\mathbf{R}_{p+1}^{(3)}} + w^{\mathbf{R}_{p+1}^{(3)}} + \Delta x \right); \\ \left(y^{\mathbf{R}_p^{(3)}} + h^{\mathbf{R}_p^{(3)}} \right) &\leq \left(y^{\mathbf{R}_{p+1}^{(3)}} + h^{\mathbf{R}_{p+1}^{(3)}} + \Delta y \right), \end{aligned}$$

где $\Delta x = \varepsilon w^{\mathbf{R}_{p+1}^{(3)}}; \Delta y = \varepsilon h^{\mathbf{R}_{p+1}^{(3)}}$.

Результатом группировки областей являются непересекающиеся области $\mathbf{R}_k^{(3)}$, где k — номер области, причем число областей должно быть не меньше 8 и не больше 9. В противном случае $\tilde{x}_{\text{СКП}} = 0$.

Для всех ОД, которые могут быть описаны вектором $\tilde{\mathbf{x}}_{\text{ОД}} = (1, \{1, 2, 3\}, 1, 1, \tilde{x}_{\text{СКП}})^T$, необходимо вычислить значение $\tilde{\mathbf{x}}_{\text{СКП}}$, т. е. отнести каждую из областей $\mathbf{R}_k^{(3)}$ к одному из возможных классов \mathbf{V}^{MNet} . Для решения данной задачи используется второй проход через модифицированную НС MobileNet, однако результат работы не приводится к задаче бинарной классификации.

Для повышения точности распознавания каждая оценка классификатора умножается на коэффициент доверия. Расчет коэффициента выполняется S_2 -образной функцией принадлежности $\mu(y^{\mathbf{R}^{(2)}})$ от координат центра НЗ:

$$\mu(y^{\mathbf{R}^{(2)}}) = \begin{cases} 0, & \text{если } y^{\mathbf{R}^{(2)}}/h^{\mathbf{I}^t} \leq a; \\ 2((y^{\mathbf{R}^{(2)}}/h^{\mathbf{I}^t} - a)/(b - a))^2, & \\ \text{если } a \leq y^{\mathbf{R}^{(2)}}/h^{\mathbf{I}^t} \leq (a + b)/(y^t/h^{\mathbf{I}^t}); \\ 1 - 2((b - y^{\mathbf{R}^{(2)}}/h^{\mathbf{I}^t})/(b - a))^2, & \\ \text{если } (a + b)/(y^{\mathbf{R}^{(2)}}/h^{\mathbf{I}^t}) < y^{\mathbf{R}^{(2)}}/h^{\mathbf{I}^t} < b; \\ 1, & \text{если } b < y^{\mathbf{R}^{(2)}}/h^{\mathbf{I}^t}, \end{cases}$$

где $y^{\mathbf{R}^{(2)}}$ — координаты центра номерного знака, $h^{\mathbf{I}^t}$ — высота изображения \mathbf{I}^t , $a = 0,4$, $b = 1$ — вещественные коэффициенты. Тогда значение каждого символа номерного знака может быть рассчитано следующим образом:

$$\tilde{\mathbf{x}}_{\text{СКП}_k} = \arg \max_n \mathbf{d}_n^{\mathbf{R}_k^{(3)}} = \left(\sum_{t=4}^{60} \mu(y^{\mathbf{R}^{(2)t}}) \mathbf{d}_n^{\mathbf{R}_k^{(3)t}} \right),$$

где t — номер кадра, причем учитывается не менее четырех и не более 60 кадров, $\mathbf{d}_n^{\mathbf{R}_k^{(3)t}}$ — вектор 1×23 , содержащий вероятность отнесения области $\mathbf{R}_k^{(3)}$ к одному из возможных классов на кадре \mathbf{I}^t , $\mathbf{d}_n^{\mathbf{R}_k^{(3)}}$ — суммарный результат по позиции k .

3. Эксперимент для оценки качества алгоритма локализации и распознавания номерного знака

Предложенный подход был реализован на языке Python с использованием библиотеки Tensorflow и Keras. С камеры наблюдения, установленной на КПП "ФГБОУ ВО КНАГУ", нами

была собрана и промаркирована тестирующая выборка, состоящая из 2453 видеофрагментов, на которых содержатся или отсутствуют ТС [26].

Значения метрик рассчитываются на основании таблицы контингентности (табл. 1), где TP — число истинно-положительных решений, TN — число истинно-отрицательных решений, FP — число ложноположительных решений, FN — число ложноотрицательных решений [13].

Таблица 1

Таблица контингентности для \mathbf{F}_3 с использованием \mathbf{f}^{MNet}

Алгоритм \mathbf{F}_3		Экспертная оценка	
		$x_{\text{НКП}} = 1$	$x_{\text{НКП}} = 0$
Оценка алгоритма	$\tilde{x}_{\text{НКП}} = 1$	$TP = 885$	$FP = 0$
	$\tilde{x}_{\text{НКП}} = 0$	$FN = 78$	$TN = 1490$

Следует отметить, что рассматривается результат работы алгоритма относительно всего НЗ, т. е. как не нахождение существующего НЗ на изображении ТС, так и ошибка алгоритма только в одном символе записывается в FN . Характеристика точности вычисляется с использованием следующих традиционных метрик: точность M^{Pr} , полнота M^{Rec} , общая точность M^{AC} , F -мера M^F , доля истинно-положительных срабатываний M^{TPR} , доля ложноположительных срабатываний M^{FPR} и площадь под кривой ошибок M^{AUC} .

Результаты расчетов представлены в табл. 2.

Таблица 2

Расчеты основных метрик для \mathbf{F}_3 с использованием \mathbf{f}^{MNet}

M^{Pr}	M^{Rec}	M^{AC}	M^F	M^{TPR}	M^{FPR}	M^{AUC}
1	0,919	0,9682	0,95779	0,95779	0	0,9595

Также был оценен алгоритм распознавания символов НЗ \mathbf{f}^{MNet} в сравнении с классическим алгоритмом Tesseract [27] $\mathbf{f}^{tesseract}$. Для этого рассматривалась точность классификации каждого символа НЗ с использованием матрицы неточностей (Confusion matrix) [13] и рассчитывалась как арифметическое среднее его точности по всем классам. Результаты представлены в табл. 3.

Таблица 3

Результирующая точность и полнота классификаторов

\mathbf{f}^{MNet}		$\mathbf{f}^{tesseract}$	
M_c^{Pr}	M_c^{Rec}	M_c^{Pr}	M_c^{Rec}
0,998	0,998	0,867	0,8665

Время обработки 1 кадра, с

Этап		Конфигурация ПК			
		Intel Core i3-7100	Intel Core i5-7400	Intel Core i5-4690	Intel Core i7-5820K
		GeForce 1030	GeForce 1050	GeForce 1050 Ti	GeForce 1080 Ti
Локализация ТС		0,277	0,094	0,089	0,03125
Локализация номерного знака		0,055	0,013	0,015	0,01562
Предобработка		0,063	0,021	0,023	0,01562
Сегментация		0,051	0,041	0,031	0,03124
Распознавание символов	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

Точность M^{Pr} для предложенного алгоритма равна 1 и характеризует долю объектов, действительно принадлежащих к данному классу относительно всех объектов, которые алгоритм отнес к данному классу. Учитывается только ошибка первого рода (FP), которая оказалась равной 0. Данный показатель достигнут "жесткими" ограничениями алгоритма при локализации НЗ. Полнота M^{Rec} для алгоритма составила 0,919 и характеризует долю найденных классификатором объектов, принадлежащих классу, относительно всех объектов этого класса. Зависит от ошибки второго рода (FN), которой соответствует 78 пропущенных объектов.

В целях сравнения скорости тестирование выполнялось на различных конфигурациях оборудования. В табл. 4 приведены результаты оценки скорости алгоритмов.

Время обработки одного кадра, содержащего номерной знак ТС, — от 0,03 до 0,09375 с, что позволяет применять предложенный алгоритм в СКУД в режиме РВ.

Были проанализированы ситуации, приводящие к ошибкам. Типичными нештатными ситуациями являются: некорректная работа подсветки, приводящая к "засвету"; нечитаемый номерной знак; отсутствующий номерной знак; автомобиль не выполнил остановку перед проездом и т. д. На рис. 4 (см. третью сторону обложки) приведены некоторые из указанных ситуаций.

С учетом предлагаемой математической модели СКУД любая нештатная ситуация может быть описана соответствующим вектором состояний. Например: 1) объект обнаружен, но

его тип не определен (рис. 4, а); 2) определен тип объекта, локализован номер, но часть символов не сегментирована (рис. 4, б); 3) при распознавании символов НЗ возникли ошибки, вследствие чего не было выполнено корректное сопоставление с БД (рис. 4, в).

При возникновении нештатной ситуации система должна уведомить оператора. Следует отметить, что ложноотрицательное срабатывание F_3 (см. табл. 1) не является критичной ошибкой и не приведет к последствиям, так как наличие обратных связей СКУД (см. рис. 1) позволяет ее компенсировать и повысить точность работы алгоритма распознавания.

Заключение

Предлагается математическая модель интеллектуальной системы контроля и управления доступом транспортных средств, учитывающая описание штатных и нештатных ситуаций. Для локализации транспортных средств и распознавания их номерных знаков предлагается композиция традиционных методов обработки изображений и двухпроходной классификации, выполняемой модифицированной архитектурой сверточной нейронной сети MobileNet. Натурный эксперимент показал, что комплексирование современных аппаратно-программных средств, алгоритмов обнаружения и распознавания объектов, даже в сложных условиях уличного видеонаблюдения, обеспечивает точность не менее 96 %. Обработка кадра, содержащего номерной знак, занимает

не менее 0,03 с и не более 0,094 с на основе графического процессора Nvidia GeForce 1080Ti, что позволяет применять предложенный алгоритм в реальном времени в системе комплексной безопасности. Разработан комплекс программ на языке Python для осуществления контрольно-пропускных функций. Высокая точность распознавания без потери скорости в режиме реального времени достигнута за счет комплексирования современных аппаратно-программных средств и алгоритмов обнаружения и распознавания объектов. Применение глубоких нейросетей совместно с современными графическими ускорителями позволяет достигнуть хороших результатов при решении задач распознавания и классификации изображений.

Список литературы

1. **Aarathi K. S., Abraham A.** Vehicle color recognition using deep learning for hazy images // ICICCT. 2017. P. 335—339.
2. **Boonsim N., Prakoonwit S.** Car make and model recognition under limited lighting conditions at night // Pattern Analysis and Applications. 2016. P. 1—13.
3. **Ciresan D., Meier U., Schmidhuber J.** Multi-column deep neural networks for image // Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2012. P. 3642—3649.
4. **Амосов О. С., Иванов Ю. С., Жиганов С. В.** Локализация человека в кадре видеопотока с использованием алгоритма на основе растущего нейронного газа и нечеткого вывода // Компьютерная оптика. 2017. Т. 41, № 1. С. 46—58.
5. **Солдатова О. П., Гаршин А. А.** Применение сверточной нейронной сети для распознавания рукописных цифр // Компьютерная оптика. 2010. Т. 34, № 2. С. 252—259.
6. **Hui Li, Peng W., Mingyu Y., Chunhua S.** Reading car license plates using deep neural networks // Image and Vision Computing. 2018. Vol. 72. P. 14—23.
7. **Masood S. Z., Guang S., Dehghan A., Ortiz E. G.** License Plate Detection and Recognition Using Deeply Learned Convolutional Neural Networks // arXiv preprint. 2017. URL: <https://arxiv.org/abs/1703.07330>.
8. **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>.
9. **Verma A., Hebbalaguppe R., Vig L., Kumar S., Hassan E.** Pedestrian detection via mixture of CNN experts and thresholded aggregated channel features // Proc. of the 2015 IEEE International Conference on Computer Vision Workshop. 2015. P. 555—563.
10. **Amosov O. S., Baena S. G., Ivanov Y. S., Soe H.** Roadway Gate Automatic Control System with the Use of Fuzzy Inference and Computer Vision Technologies // The 12th IEEE Conference on Industrial Electronics and Applications. 2017. P. 706—711.
11. **Амосов О. С.** Нейросетевые и нечеткие методы оценивания стохастических систем. Диссертация ... доктора технических наук: 05.13.18. Комсомольск-на-Амуре, 2004. 352 с.
12. **Амосов О. С.** Системы нечеткой логики для фильтрации марковских последовательностей // Информационные технологии. 2004. № 11. С. 16—22.
13. **Binary classification performances measure cheat sheet.** URL: <https://kaggle2.blob.core.windows.net/forum-message-attachments/144412/5346/modelperfcheatsheet.pdf> (дата обращения: 20.12.2017).
14. **Redmon J., Divvala S. K., Girshick R. B., Farhadi A.** You Only Look Once: Unified, Real-Time Object Detection // arXiv preprint. 2015. URL: <https://arxiv.org/pdf/1506.02640v4.pdf>.
15. **Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A.** Going deeper with convolutions // 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015. P. 1—9.
16. **Intersection over Union (IoU) for object detection.** URL: <https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/> (дата обращения: 20.10.2017).
17. **Viola P., Jones M., Show D.** Detecting Pedestrians Using Patterns of Motion and Appearance // Int. J. Comput. Vision. 2005. Vol. 63, N. 2. P. 153—161.
18. **Библиотека компьютерного зрения OpenCV.** URL: <http://docs.opencv.org/> (дата обращения: 20.09.2017).
19. **Yadav G., Maheshwari S., Agarwal A.** Contrast limited adaptive histogram equalization based enhancement for real time video system // ICACCI. 2014. P. 2392—2397.
20. **Matas J., Chum O., Urban M., Pajdla T.** Robust wide baseline stereo from maximally stable extremal regions // Proc. of British Machine Vision Conference, 2002. P. 384—396.
21. **Keras is a high-level neural networks API.** URL: <https://keras.io/> (дата обращения: 02.09.2017).
22. **Siam M., Gamal M., Abdel-Razek M., Yogamani S., Jagersand M.** RTSeg: Real-time Semantic Segmentation Comparative Study // arXiv preprint. 2018. URL: <https://arxiv.org/abs/1803.02758>.
23. **ImageNet.** URL: <http://www.image-net.org/> (дата обращения: 02.10.2017).
24. **Yosinski J., Clune J., Bengio Y., Lipson H.** How transferable are features in deep neural networks? // NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems. 2014. Vol. 2. P. 3320—3328.
25. **Hinton G., Srivastava N., Swersky K.** Lecture 6a Overview of mini-batch gradient descent // Computer Science University of Toronto. 2012. URL: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6b.pdf (дата обращения: 30.09.2017).
26. **Видео с камеры видеонаблюдения СКУД КНАГУ.** URL: <http://scm.tframe.ru/public/dataset.tar.gz> (дата обращения: 30.01.2018).
27. **Tesseract OCR.** URL: <https://github.com/tesseract-ocr/tesseract> (дата обращения: 10.01.2018).

O. S. Amosov, D. Sc., Prof., Principal Researcher, e-mail: osa18@yandex.ru,
S. G. Amosova, Ph. D., Assistant Professor, Senior Researcher, e-mail: amosovsg@yandex.ru,
V. A. Trapeznikov Institute of Control Sciences of RAS,
Y. S. Ivanov, Ph. D., Assistant Professor, e-mail: ivanov_ys@icloud.com,
S. V. Zhiganov, Postgraduate Student, e-mail: zhiganov@knastu.ru,
Komsomolsk-na-Amure State University

Modelling of Intelligent Access Monitoring and Control System for Vehicles with Using the Deep Neural Networks

Mathematical model of intelligent access monitoring and control system for vehicle is developed. It differs from the existing ones as it allows generating the control actions during the handling of normal and abnormal situations for a significant reduction of recognition errors. The realization of vehicle localization using the YOLO deep neural network, allowing additional determining the type of the access object, is proposed. The solution of the license plate localization and recognition problem is based on the composition of the traditional image processing methods and the two-pass classification carried out by the modified architecture of the MobileNet convolutional network. It has been experimentally proved that the application of the developed approach gives a percentage of correct license plates recognition in the video stream of not less than 96 %, depending on external conditions. The programs complex by using Python is realized.

Keywords: mathematical model, monitoring and control system, vehicle, localization, recognition, convolutional neural network, MobileNet

DOI: 10.17587/it.25.116-127

Acknowledgement: The work was financially supported by the Russian Ministry of Education and Science — the project № 2.1898.2017/4.6 "Designing the Mathematical and Algorithmic Ware of Intelligent Information and Telecommunication System for Higher Educational Institution Security".

References

1. Aarathi K. S., Abraham A. Vehicle color recognition using deep learning for hazy images, *ICICCT*, 2017, pp. 335–339.
2. Boonsim N., Prakoonwit S. Car make and model recognition under limited lighting conditions at night, *Pattern Analysis and Applications*, 2016, pp. 1–13.
3. Ciresan D., Meier U., Schmidhuber J. Multi-column deep neural networks for image, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3642–3649.
4. Amosov O. S., Ivanov Ju. S., Zhiganov S. V. Lokalizacija cheloveka v kadre videopotoka s ispol'zovaniem algoritma na osnove rastushhego nejronnogo gaza i nechetkogo vyvoda (Localization of the human in the frame of the video stream by using an algorithm based on neural growing gas and fuzzy inference), *Komp'yuternaja Optika*, 2017, vol. 41, no. 1, pp. 46–58 (in Russian).
5. Soldatova O. P., Garshin A. A. Primenenie svertochnoj nejronnoj seti dlja raspoznavanija rukopisnyh cifr (Application of convolutional neural network for handwriting recognition), *Komp'yuternaja Optika*, 2010, vol. 34, no. 2, pp. 252–259 (in Russian).
6. Hui Li, Peng W., Mingyu Y., Chunhua S. Reading car license plates using deep neural networks, *Image and Vision Computing*, 2018, vol. 72, pp. 14–23.
7. Masood S. Z., Guang S., Dehghan A., Ortiz E. G. License Plate Detection and Recognition Using Deeply Learned Convolutional Neural Networks, arXiv preprint, 2016, available at: <https://arxiv.org/abs/1703.07330>.
8. 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, available at: <https://arxiv.org/pdf/1704.04861.pdf>.
9. Verma A., Hebbalaguppe R., Vig L., Kumar S., Hassan E. Pedestrian detection via mixture of CNN experts and thresholded aggregated channel features, *Proceedings of the 2015 IEEE International Conference on Computer Vision Workshop*, 2015, pp. 555–563.
10. Amosov O. S., Baena S. G., Ivanov Y. S., Soe H. Roadway Gate Automatic Control System with the Use of Fuzzy Inference and Computer Vision Technologies, *The 12th IEEE Conference on Industrial Electronics and Applications*, 2017, pp. 706–711.
11. Amosov O. S. *Nejrosetevye i nechetkie metody ocenivaniya stohasticheskikh sistem. Dissertacija... doktora tehniceskikh nauk: 05.13.18* (Neural network and fuzzy methods for estimation stochastic systems), Komsomol'sk-na-Amure, 2004, 352 p. (in Russian).
12. Amosov O. S. *Sistemy nechetkoj logiki dlja fil'tracii markovskih posledovatel'nostej* (Fuzzy logic systems for filtering Markov sequences), *Informacionnye Tehnologii*, vol. 11, 2004, pp. 16–22 (in Russian).
13. Binary classification performances measure cheat sheet, available at: https://kaggle2.blob.core.windows.net/forum-message-attachments/144412/5346/modelperf_cheatsheet.pdf (accessed: 20.12.2017).
14. Redmon J., Divvala, S. K., Girshick R. B., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection, arXiv preprint. 2015, available at: <https://arxiv.org/pdf/1506.02640v4.pdf>.

15. Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A. Going deeper with convolutions, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.
16. **Intersection over Union (IoU)** for object detection, available at: <https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/> (accessed: 20.10.2017).
17. Viola P., Jones M., Snow D. Detecting Pedestrians Using Patterns of Motion and Appearance, *Int. J. Comput. Vision.*, 2005, vol. 63, no. 2, pp. 153–161.
18. *Библиотека комп'ютрного зренія OpenCV* (Computer vision library OpenCV), available at: <http://docs.opencv.org/> (accessed: 20.09.2017) (in Russian).
19. Yadav G., Maheshwari S., Agarwal A. Contrast limited adaptive histogram equalization based enhancement for real time video system, *ICACCI*, 2014, pp. 2392–2397.
20. Matas J., Chum O., Urban M., Pajdla T. Robust wide baseline stereo from maximally stable extremal regions, *Proc. of British Machine Vision Conference*, 2002, pp. 384–396.
21. **Keras** is a high-level neural networks API, available at: <https://keras.io/> (accessed: 02.09.2017).
22. Siam M., Gamal M., Abdel-Razek M., Yogamani S., Jagersand M. RTSeg: Real-time Semantic Segmentation Comparative Study, *arXiv preprint*, 2018, available at: <https://arxiv.org/abs/1803.02758>.
23. **Image Net**, available at: <http://www.image-net.org/> (accessed: 02.10.2017).
24. Yosinski J., Clune J., Bengio Y., Lipson H. How transferable are features in deep neural networks?, *NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2014, vol. 2, pp. 3320–3328.
25. Hinton G., Srivastava N., Swersky K. Lecture 6a Overview of mini-batch gradient descent, *Computer Science University of Toronto*, 2012, available at: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf (accessed: 30.09.2017).
26. *Video s kamery videonabljudenija SKUD KnAGU* (Video from access control system security camera of KnAGU), available at: <http://scm.tframe.ru/public/dataset.tar.gz> (accessed: 30.01.2018) (in Russian).
27. **Tesseract OCR**, available at: <https://github.com/tesseract-ocr/tesseract> (accessed: 10.01.2018).

XX ВСЕРОССИЙСКАЯ КОНФЕРЕНЦИЯ МОЛОДЫХ УЧЕНЫХ ПО МАТЕМАТИЧЕСКОМУ МОДЕЛИРОВАНИЮ И ИНФОРМАЦИОННЫМ ТЕХНОЛОГИЯМ

28 октября — 1 ноября 2019 г.

Конференция организуется в целях обсуждения актуальных результатов исследований молодых научных сотрудников, аспирантов и студентов старших курсов в области вычислительной и прикладной математики и информатики. Участие в конференции позволит молодым ученым получить представление о результатах исследований в области современного математического моделирования, вычислительных и информационных технологий, установить научные контакты, познакомиться с широким кругом задач, представленных в докладах участников. В рамках конференции планируется работа двух основных секций: вычислительные технологии и информационные технологии.

Организаторы конференции

- Институт вычислительных технологий СО РАН
- Институт динамики систем и теории управления СО РАН
- Институт вычислительного моделирования СО РАН
- Институт вычислительной математики и математической геофизики СО РАН
- Новосибирский национальный исследовательский государственный университет
- Новосибирский государственный технический университет
- Сибирский государственный университет телекоммуникаций и информатики

Научные направления

- Математическое моделирование
- Численные методы и методы оптимизации
- Высокопроизводительные и распределенные вычисления
- Информационные и геоинформационные системы
- Управление, обработка, защита и хранение информации
- Автоматизация и теория управления

Тезисы принятых докладов публикуются в печатном виде до начала конференции.