

15. Solov'ev V.V. Arkhitektury PLIS firmy Xilinx: CPLD i FPGA 7-y serii [Xilinx FPGA architectures: CPLD and FPGA 7th series], Moscow: Goryachaya liniya – Telekom, 2019, 392 p.
16. Avizienis A., Laprie J., Randell B., Landwehr C. Basic concepts and taxonomy of dependable and secure computing, *IEEE Transactions on Dependable and Secure Computing*, 2004, No. 1, pp. 11-33.
17. Mukha Ar A. Kolichestvennaya otsenka urovnya garantospobnosti komp'yuternykh sistem [Quantitative assessment of the level of guarantee capacity of computer systems], *MMS [MMS]*, 2019, No. 4. Available at: <https://cyberleninka.ru/article/n/kolichestvennaya-otsenka-urovnnya-garantospobnosti-kompyuternykh-sistem> (accessed 31 March 2025).
18. Veniaminov S.S. Kosmicheskii musor ugrozhaet planete [Space debris threatens the planet], *Vozdushno-kosmicheskaya sfera [Air and space sphere]*, 2016, No. 1 (86). Available at: <https://cyberleninka.ru/article/n/kosmicheskii-musor-ugrozhaet-planete> (accessed 24 July 2024).
19. Klyushnikov V.Yu. Sindrom Kesslera: budet li zakryta doroga v kosmos? [Kessler syndrome: will the road to space be closed?], *VKS [VKS]*, 2021, No. 4 (109). Available at: <https://cyberleninka.ru/article/n/sindrom-kesslera-budet-li-zakryta-doroga-v-kosmos> (accessed 25 October 2024).
20. Dzyuba A.P. Perspektivy razvitiya fazirovannykh antenykh reshetok [Prospects for the development of phased antenna arrays], *Vestnik DGTU. Tekhnicheskie nauki [Bulletin of DSTU. Technical sciences]*, 2013, No. 3. Available at: <https://cyberleninka.ru/article/n/perspektivy-razvitiya-fazirovannykh-antenykh-reshetok> (accessed 25 October 2024).

**Левин Илья Израилевич** – Южный федеральный университет; e-mail: levin@superevm.ru; г. Таганрог, Россия; тел.: +78634612111; кафедра интеллектуальных и многопроцессорных систем; зав. кафедрой; д.т.н.; профессор.

**Буряков Дмитрий Сергеевич** – Южный федеральный университет; e-mail: dburiakov@sfedu.ru; г. Таганрог, Россия; тел.: +79198955502; кафедра интеллектуальных и многопроцессорных систем; аспирант.

**Levin Ilya Izrailevich** – Southern Federal University; e-mail: levin@superevm.ru; Taganrog, Russia; phone: +78634612111; the Department of Intelligent and Multiprocessor Systems; head of department; dr. of eng. sc.; professor.

**Buryakov Dmitrii Sergeevich** – Southern Federal University; e-mail: dburiakov@sfedu.ru; Taganrog, Russia; phone: +79198955502; the Department of Intelligent and Multiprocessor Systems; postgraduate student.

УДК 004.932.75'1

DOI 10.18522/2311-3103-2025-3-134-144

**Д.А. Безуглов, М.С. Мищенко, С.Е. Мищенко**

## **АЛГОРИТМ ПОДГОТОВКИ ДАННЫХ ОБУЧЕНИЯ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РАСПОЗНАВАНИЯ БУКВ И СИМВОЛОВ**

*Точность распознавания текстовых изображений на практике остается ограниченной. Это связано с тем, что в алфавит символов могут входить строчные и прописные буквы со схожим начертанием, а также составные символы, образованные из нескольких более простых символов. Для решения этой проблемы систему распознавания символов дополняют системами семантического или структурного анализа, что существенно усложняет информационную систему для распознавания текста. В настоящее время для распознавания одиночных символов широко применяют сверточные нейронные сети, для обучения которых используют базу данных с изображениями распознаваемых символов. В работе предложен алгоритм, отличающийся тем, что в изображение одиночного символа для обучающей выборки включают фрагменты символов, которые могут быть расположены в строке в непосредственной близости от распознаваемого символа. Формирование изображений для обучающей выборки имитирует процесс сегментации символа по яркости, который обычно используют при выделении символа для дальнейшего распознавания. При этом оценивают размеры символа, дополняют его изображениями соседних символов, а затем оценивают размеры области, изображения, которое будет помещено в обучающую выборку. Полученное изображение масштабируют и обрезают таким образом, чтобы на вход нейронной сети поступали изображения заданного размера. В работе для распознавания алфавита символов, включающего прописные и строчные символы русского и английского алфавитов, цифры, символы*

и знаки препинания предложено использовать множество сверточных нейронных сетей, каждая из которых обучена распознавать один символ. Выбор символа осуществляется путем сравнения откликов всех нейронных сетей и выбора максимального отклика. Проведено сравнение предложенного алгоритма подготовки данных для обучения с известным алгоритмом, основанным на использовании изображений одиночных символов. Установлено, что предложенный алгоритм подготовки данных для обучения обеспечивает повышение точности распознавания алфавита из 138 символов более, чем в два раза.

*Алгоритм; алфавит; символ; распознавание; сверточная нейронная сеть; обучающая выборка.*

**D.A. Bezuglov, M.S. Mishchenko, S.E. Mishchenko**

#### **ALGORITHM FOR TRAINING DATA PREPARATION OF CONVOLUTIONAL NEURAL NETWORKS FOR LETTER AND CHARACTER RECOGNITION**

*The accuracy of text image recognition remains limited in practice. This is due to the fact that the alphabet of symbols can include lowercase and uppercase letters with a similar font, as well as composite characters formed from several simpler characters. To solve this problem, the character recognition system is supplemented with semantic or structural analysis systems, which significantly complicates the information system for text recognition. Currently, convolutional neural networks are widely used for recognizing single characters, for which a database with images of recognized characters is used for training. The paper proposes an algorithm characterized in that the image of a single character for a training sample includes fragments of characters that can be located in a line in close proximity to the recognized character. This allows you to expand the set of images for training and additionally include information in the image about the placement of the symbol in the string, its relative size and whether this symbol is composite. The formation of images for the training sample simulates the process of segmentation of a symbol by brightness, which is usually used when selecting a symbol for further recognition. At the same time, the size of the symbol is estimated, it is supplemented with images of neighboring symbols, and then the size of the area, the image that will be placed in the training sample, is estimated. The resulting image is scaled and cropped in such a way that images of a given size are received at the input of the neural network. In the work, to recognize the alphabet of symbols, including uppercase and lowercase characters of the Russian and English alphabets, numbers, symbols and punctuation marks, it is proposed to use a variety of convolutional neural networks, each of which is trained to recognize one character. The symbol is selected by comparing the responses of all neural networks and selecting the maximum response. The proposed algorithm for training data preparation is compared with a well-known algorithm based on the use of images of single characters. It is established that the proposed algorithm for preparing data for training provides an increase in the accuracy of recognizing the alphabet of 138 characters by more than two times.*

*Algorithm; alphabet; symbol; recognition; convolutional neural network; training sample.*

**Введение.** Распознавание символов на изображениях представляет собой одну из первых задач теории распознавания образов, которая начала интенсивно развиваться со второй половины 50-х годов [1] и была связана с созданием машинного зрения и читающих автоматов. В 1957г. Ф. Розенблатом был предложен подход к распознаванию образов на основе перцептрона [1], что послужило отправной точкой для создания современных сверточных нейронных сетей (СНС), способных распознавать символы и тексты [2–5]. В настоящее время системы распознавания текстовых изображений применяют при автоматическом переводе, для распознавания автомобильных номеров, регистрации железнодорожных составов [6–9]. Эффективность распознавания текстов значительно зависит от многих факторов. К их числу могут быть отнесены искажения при проецировании трехмерных объектов с текстовой информацией на плоскость, шумы, турбулентность атмосферы, разрешение текстового документа при сканировании. В результате в теории и практике распознавания текстов сложилось противоречие. С одной стороны, обучить СНС распознавать одиночные символы не сложно. При этом точность распознавания может достигать 99%. Однако, с другой стороны, при извлечении символов из текста приходится строить сложные классификаторы, которые анализируют различные дополнительные признаки, учитывающие контекстное содержание документа. Это

значительно усложняет систему распознавания текстов, а точность распознавания может снижаться приблизительно до 80% [8]. В информационных системах распознавания текстов с высокой стоимостью ошибки (в государственных или экономических структурах) системы распознавания текстов требуют участия оператора для контроля за корректностью работы искусственной нейронной сети (ИНС).

В известной литературе, посвященной глубокому обучению классическим примером является структура СНС, обеспечивающая распознавание рукописных изображений цифр [2, 10]. Цифр всего 10, они имеют одинаковую высоту кегля. Это позволяет использовать достаточно простой классификатор, состоящий из двух чередующихся сверточных слоев и слоев агрегирования, а также полносвязной нейронной сети с 10-ю выходами. На вход классификатора поступают изображения размером 32x32 или 28x28 пикселей. Примеры программной реализации подобных классификаторов известны и ссылаются на уже готовые базы для обучения [10].

Недостатком такого классификатора является то, что при увеличении алфавита символов для распознавания сложность СНС должна возрастать. При расширении алфавита символов будет увеличиваться число выходов СНС. Следовательно, при этом потребуются переобучение всей СНС.

В связи с этим в работе [8] было предложено строить классификатор из множества нейронных сетей. Каждая нейронная сеть имеет единственный выход и обучена распознавать только один символ. После того, как символу ставятся в соответствие выходные сигналы всех нейронных сетей принимается решение о том, изображение какого символа поступало на вход.

В этом случае для решения задачи распознавания одного символа достаточно применить простейшую полносвязную сеть, состоящую из трех слоев (входного, скрытого и выходного). Число нейронов входного слоя соответствует длине входного вектора. В скрытом слое было предложено сократить число нейронов до 70% по сравнению со входным слоем. Выходной слой содержит один нейрон. Для упрощения нейронной сети вместо сверточных слоев было предложено использовать аппарат вейвлет-преобразований и метод главных компонент. В то же время очевидно, что использование данного математического аппарата было продиктовано необходимостью подавления импульсных шумов на изображении.

Авторы [11] указывали, что распознавание символов при помощи нейронных сетей сталкивается с трудностями при распознавании схожих по начертанию (строчных и прописных). В связи с этим при принятии решения учитывались два максимальных отклика множества нейронных сетей. Однако в работе [11] отсутствует подробное описание этой части работы алгоритма.

В работе [11] также указывалось, что в ряде случаев при распознавании текстов могут возникать сложности распознавания некоторых символов, которые сложно разделить, используя обычную сегментацию по яркости.

Наконец, требуется отметить известную проблему распознавания составных символов, распознавание которых потребует использования контекстных анализаторов [3, 12–15].

Цель работы состоит в повышении точности распознавания текстовых документов при помощи ИНС без дополнения системы распознавания текстов контекстными или иными дополнительными классификаторами.

**Алгоритм подготовки данных для распознавания символов и формирования обучающей выборки.** На наш взгляд, проблемы с распознаванием текстов в работах [8, 11] были связаны не с общим подходом к построению классификатора, а с подготовкой данных для обучения. В связи с этим для распознавания текста на изображении будем использовать множество идентичных СНС, структура которых показана на рис. 1.

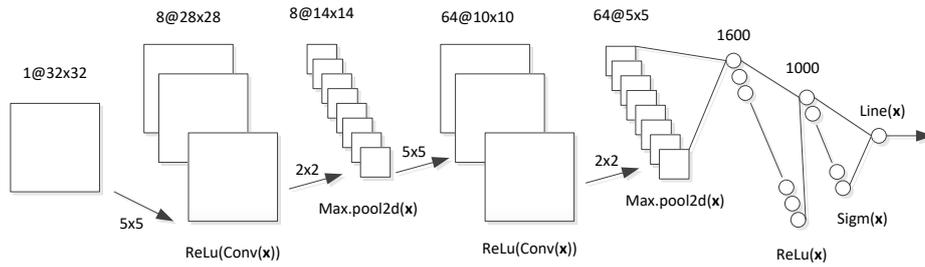


Рис. 1. Структура СНС для распознавания символа

Поступающее на вход классификатора изображение получает набор откликов всех нейронных сетей (см. рис. 1), из которых выбирают сеть с максимальным откликом. Поскольку выбранная СНС обучена распознавать единственный символ, то решение о выборе символа становится однозначным.

В соответствии с работой [8] для формирования обучающей выборки и проведения экспериментов с распознаванием использовались файлы шрифтов. Известно, что файлы шрифтов операционной системы содержат векторные изображения символов шрифта. При формировании растрового изображения, выводимого на экран, задают  $H_0$  размер (высоту) кегля символа, что позволяет требуемым образом масштабировать векторное изображение символа. Отметим, что в отличие от типографской печати ширина кегля символа может варьироваться. В результате при подготовке символа задавалось имя шрифта и высота кегля. Далее изображение масштабировалось до необходимого размера. При необходимости к изображению добавляются пустые строки и столбцы.

При работе с ИНС входное изображение символа помещают в центр квадратной области размером  $L \times L$  (как правило,  $L = 32$  или  $28$  [8]). При этом, если  $H_0 < L$ , изображение символа масштабируют с коэффициентом  $C$  так, что  $L \approx \lfloor H_0 \rfloor \leq L$  (здесь  $\lfloor H_0 \rfloor$  обозначает операцию округления).

Это позволяет формировать обучающую выборку из изображений символа одного и того же шрифта с разными значениями  $H_0$ . Расширения вариаций изображений символа в обучающей выборке также добиваются использованием нескольких шрифтов с различными стилями (могут использоваться шрифты с засечками и без, нормальные и жирные начертания символа). В работе [8] для формирования обучающей выборки использовались 10 размеров кегля от 12 до 36, 8 шрифтов (4 без засечек и 4 с засечками), а также гарнитуры с нормальным и жирным представлением символа. В результате каждому символу можно было поставить в соответствие 160 изображений. Обучающая выборка состоит из нескольких тысяч изображений, в которой число изображений распознаваемого символа приблизительно равно числу других изображений символов.

В качестве примера на рис. 2 приведены примеры изображений символа «А» с высотой кегля 24 и 12, извлекаемые из файла шрифта arial.ttf и преобразуемые до изображения заданного размера. Как видно из данного рисунка, изменение размеров кегля позволяет варьировать изображение символа, для создания обучающей выборки.

Недостаток такого подхода становится очевиден, если формировать изображения символов не из файла шрифта, а из изображения с текстом. В этом случае понятие высоты кегля не может быть использовано. После сегментации символа из текста он имеет высоту и ширину, которые обозначим  $h_0, w_0$ . На самом деле эти размеры могут отличаться от размеров символа и соответствовать его части. Дело в том, что при сегментации по яркости или при помощи других алгоритмов [16–18] вместо символа «Й» можно получить изображение «И», а вместо символа «Ы» – только «Ь» или «І». Большие сложности могут возникнуть при выделении небольших по высоте элементов, например «.».

В этом случае вероятна ошибка в выборе коэффициента масштабирования  $C$ . Эта ошибка, скорее всего, приведет к ошибке распознавания символа. Для разрешения данных проблем в систему распознавания приходится внедрять сложную многоуровневую логическую или контекстную обработку.

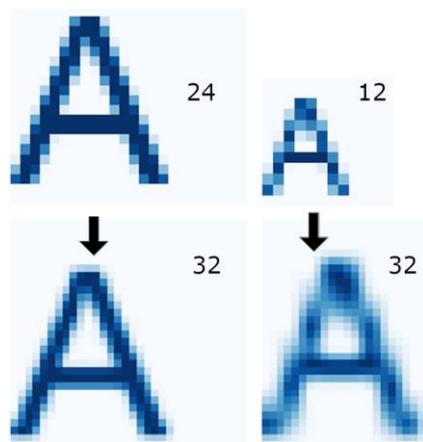


Рис. 2. Преобразование изображений различной высоты в стандартное изображение 32x32 пикселя

В работе предлагается при формировании входного изображения для распознавания и обучения СНС использовать следующий алгоритм.

1. Выполним сегментацию изображения и выделим фрагмент символа с размерами  $h_0, w_0$ . Массив соответствующего изображения обозначим  $\mathbf{b}_0$ . Изображение  $\mathbf{b}_0$  содержит яркие точки, пустые строки и столбцы по краям изображения отсутствуют. Если символ составной, то выделенный фрагмент должен занимать наибольшую площадь.

В качестве примера на рис. 3 показан процесс выделения фрагмента составного символа «Й» с исходным размером 24. Параметры фрагмента символа  $h_0 = 17, w_0 = 15$ . Отметим, что высота исходного изображения всего символа больше на 4 пикселя.

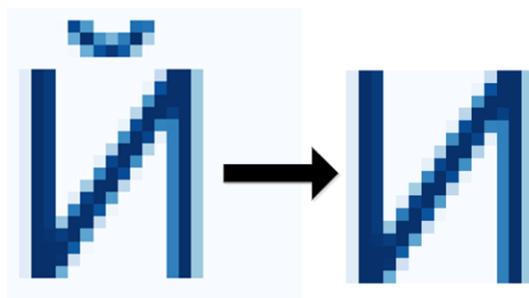


Рис. 3. Сегментация фрагмента символа, занимающего максимальную площадь

2. Дополним изображение  $\mathbf{b}_0$  с размерами  $h_0, w_0$  так, чтобы получить новое изображение  $\mathbf{b}_1$  с размерами  $h_1, w_1$ . Изображение  $\mathbf{b}_1$  должно быть размещено в центральной области изображения  $\mathbf{b}_0$ , высота  $h_1$  должна быть выбрана таким образом, чтобы в новое изображение не попали целые изображения символов из текстовых строк над и под символом, но были включены все элементы символа, которые могли быть утеряны при сегментации по яркости. По всей видимости достаточно задать  $h_1 = 1.5h_0 \dots 2h_0$ . Несколь-

ко сложнее с выбором  $w_1$ . Минимальная ширина символа может занимать несколько пикселей, а максимальная – совпадать с его высотой. В связи с этим данный параметр предлагается выбирать из соотношения  $w_1 = 9w_0$ .

На рис. 4 представлено изображение символа «Й», размещенного между двумя другими символами, выбранными случайным образом.

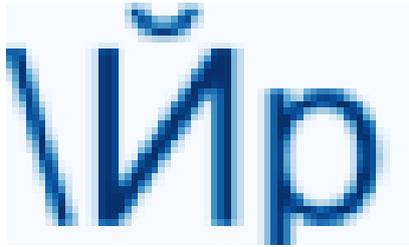


Рис. 4. Изображение символа «Й» в окружении случайно выбранных символов

3. Далее определим высоту прямоугольной области  $h_2$ , которая содержит все яркие точки изображения  $\mathbf{b}_1$ .

4. Выделим центральную квадратную область изображения, включающую  $\mathbf{b}_0$ , но имеющую высоту и ширину, равные  $h_2$ . В результате того, что расширенное изображение в тексте сможет захватить фрагменты изображений соседних символов получим более корректную дополнительную информацию о фактической высоте символа и его положении на кегле.

5. Масштабируем изображение так, чтобы  $\lfloor Ch_2 \rfloor = L$ . В результате будет получено изображение для распознавания, показанное на рис. 5.

Особенность изображения состоит в том, что по краям изображение «зашумляется» фрагментами соседних символов, но содержит все необходимые фрагменты, образующие символ. Соседние символы несут информацию о высоте буквы, что позволяет корректно идентифицировать ее положение в строке.

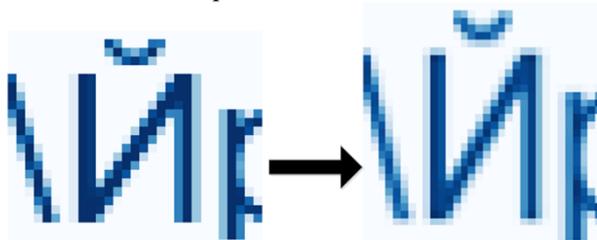


Рис. 5. Вариация изображения символа «Й», помещаемая в обучающую выборку

Предлагаемый алгоритм используется только для распознавания при извлечении символа из изображения. Для формирования обучающей выборки вместо расширения изображения символа и перехода от изображения  $\mathbf{b}_0$  к  $\mathbf{b}_1$  предлагается случайным образом генерировать изображения соседних символов текста, которые попадают в изображение для обучения. Вариации изображений символа будут включать в себя выбранные шрифты, гарнитуры, размеры символов и варианты фрагментов соседних символов. Естественно, что число вариантов изображений одного и того же символа может значи-

тельно увеличиться. Это приведет к необходимости увеличения обучающей выборки. При этом фрагменты соседних символов, попадающих в обучающую выборку должны в результате обучения восприниматься не только как шум, но и как дополнительная информация о положении символа в строке и, соответственно, на кегле.

**Численные эксперименты.** Предложенный алгоритм обучения был использован для подготовки данных для обучения СНС, представленной на рис. 1. При обучении СНС с данными, подготовленными по известной процедуре каждый символ варьировался 64 изображениями. Общее число символов включало в свой состав 138 символов (основные символы – 29, строчные и прописные символы русского алфавита – по 33 символа), 15 прописных и 18 строчных символов английского алфавита, отличающихся по начертанию от русских букв, а также 10 цифр. Перечень символов приведен в таблице 1.

Для обучения использовались 8 шрифтов: arial.ttf, arialbd.ttf, calibri.ttf, calibril.ttf, cour.ttf, courbd.ttf, times.ttf, timesbd.ttf с размерами, 12, 16, 18, 20, 22, 24, 26, 32. При формировании обучающей базы изображений по предлагаемому алгоритму число сгенерированных изображений возросло в 10 раз.

Таблица 1

Перечень символов для распознавания

Группа символов	Символы
Прописные русские буквы	'А','Б','В','Г','Д','Е','Ё','Ж','З','И','Й','К','Л','М', 'Н','О','П','Р','С','Т','У','Ф','Х','Ц','Ч','Ш','Щ','Ъ', 'Ы','Ь','Э','Ю','Я'
Строчные русские буквы	'а','б','в','г','д','е','ё','ж','з','и','й','к','л','м','н','о','п','р', 'с','т','у','ф','х','ц','ч','ш','щ','ъ','ь','ы','э','ю','я'
Символы и знаки препинания	'~','!','@','#','\$','%','^','&',*','(',')','+', '-', '=', '[', ']', '{', '}', '\\', ' ', '<', '>', '/', '?', ':', ';', ',', '.', '№', '%'
Цифры	'1','2','3','4','5','6','7','8','9','0'
Прописные английские буквы	'Q','W','R','Y','U','T','J','S','D','F','G','L','Z','V','N'
Строчные английские буквы	'q','w','r','t','i','j','s','d','f','g','h','k','l','z','v','b','n','m'

Модель СНС была реализована на языке Python с использованием фреймворка PyTorch [19, 20]. При обучении использовался оптимизатор Adam с параметром 0.001. Время обучения всех нейронных сетей для первого и второго набора данных отличалось приблизительно в 10 раз и для подготовленных данных по предлагаемому алгоритму составило около 12 часов. При обучении использовалось до 200 эпох обучения для каждого символа. При обучении использовались обучающая и тестовые выборки, разделенные по объему в соотношении 70% и 30%. Тестовая выборка использовалась для контроля переобучения. При этом в качестве критерия остановки процесса обучения использовалось монотонное удаление качества сети по тестовой выборке в течение 8-ми эпох обучения. При достижении предельного числа эпох или обнаружения переобучения сети использовались коэффициенты СНС, при которых качество сети на тестовых данных было наилучшим.

После обучения был проведен эксперимент по распознаванию символов с использованием двух вариантов обучения при распознавании символов, генерируемых с использованием шрифта calibril.ttf, cambriab.ttf. Перечень ошибок распознавания приведен в табл. 2.

Таблица 2

**Перечень ошибок при распознавании**

Данные для обучения	Шрифт	Число ошибок (процент ошибок)	Не распознанные символы
Одиночные символы	cambriab.ttf	53 (38%)	'(', '+', '=', ']', ' ', '<', '>', '/', ';', ':', '№', '%', 'В', 'Г', 'Д', 'Е', 'Й', 'О', 'П', 'У', 'Х', 'Ц', 'Ш', 'Ы', 'Э', 'Б', 'Е', 'Ё', 'Ж', 'И', 'О', 'Ч', 'Ш', 'Э', 'Q', 'Y', 'U', 'T', 'J', 'F', 'G', 'q', 'r', 'i', 's', 't', 'v', 'b', 'z', '5', '7', '8', '0'
	calibril.ttf	20 (14%)	'=', ']', ' ', ';', ':', '№', '%', 'Е', 'З', 'Й', 'Щ', 'Б', 'Ы', 'Г', 'Ё', 'Й', 'Р', 'Ы', 'Q', 'T'
Предлагаемый алгоритм	cambriab.ttf	21 (15%)	'<', '+', ';', ':', '№', 'Д', 'Л', 'О', 'Ц', 'Ш', 'Ы', 'Ю', 'Б', 'Ш', 'W', 'J', 'Z', '3', '5', '8', '9'
	calibril.ttf	9 (5%)	'№', 'З', 'Ы', 'К', 'О', 'Ы', 'Т', 'Т', 'З'

Анализ результатов в табл. 2 показывает, что в некоторых случаях использование СНС, обученных для распознавания одиночных символов, позволяет правильно распознать составные символы. Такой результат может иметь место в том случае, если при одном и том же входном сигнале отклик СНС близок. Однако в этом случае результат распознавания можно считать неустойчивым. Это подтверждают зависимости, приведенные на рис. 6 и 7. На рис. 6 приведены отклики СНС при распознавании символов, в которых допущена ошибка. На рис. 7, напротив, обе сети распознали символ. Кривая 1 на рис. 6 и 7 соответствует обучению СНС на одиночных символах, а кривая 2 при подготовке данных для обучения предлагаемым алгоритмом. Данные штриховой кривой показывают, что буква «Ы» (номер 57) перепутана с символами «о», «Б», «б» – номера 77, 56, 125 в порядке убывания по амплитуде. Сплошная кривая предлагает два наиболее вероятных кандидата для распознавания и перепутала «Б» и букву «Ы». Возможно, эту неоднозначность можно устранить, если обучение производить на большем числе вариаций изображений символов. Как видно, несмотря на ошибку при распознавании составного символа «Ы», он является вторым кандидатом на распознавание. В то же время обучение по одиночным символам даже не предлагает такой вариант для распознавания. Для рассмотренных примеров число ошибок благодаря использованию предложенного алгоритма подготовки символов для распознавания сокращено более чем в 2 раза.

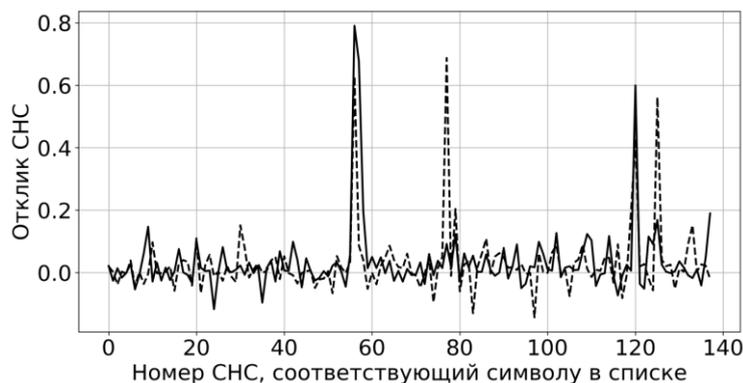


Рис. 6. Отклики СНС при неправильном распознавании символа «Ы» – номер 57: штриховая кривая – СНС обучена на одиночных символах, сплошная – предлагаемый алгоритм

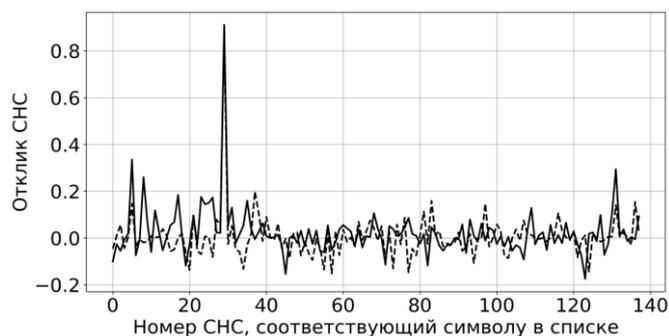


Рис. 7 Отклики СНС при правильном распознавании символа «А» – номер 29: штриховая кривая – СНС обучена на одиночных символах, сплошная – СНС обучена после преобразования данных

**Выводы.** Таким образом, предложенный алгоритм подготовки символов для распознавания отличается от известных включением в формируемое изображение фрагментов других символов, соответствующих соседним символам в строке и фрагментов, расположенных от связанного множества ярких точек символа, образующих основной по площади неразрывный фрагмент символа. Для рассмотренных примеров предлагаемый алгоритм обеспечивает сокращение числа ошибок более чем в 2 раза по сравнению с СНС, обученной для распознавания одиночных символов. Процент ошибок предлагаемого алгоритма находится в диапазоне от 15% до 5%, без использования предложенного алгоритма ошибки изменяются в диапазоне от 14% до 38%.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Горелик А.Л., Скрипкин В.А. Методы распознавания. – М.: Высшая школа, 1984. – 208 с.
2. Goodfellow I., Bengio Y., Courville A. Deep Learning. – MIT Press, 2016. – <http://www.deeplearningbook.org>.
3. Патент РФ 2661750: МПК G06K 9/20. Распознавание символов с использованием искусственного интеллекта / Чупинин Ю.Г.; заявл. 30.05.2017; опубл. 19.07.2018; Бюл. № 20.
4. Николенко С., Кадурын А., Архангельская Е. Глубокое обучение. Погружение в мир нейронных сетей. – СПб.: Питер, 2021. – 476 с.
5. Forsyth D.A., Ponce J. Computer Vision: A Modern Approach. – 2nd ed. – New Jersey: Prentice Hall, 2011. – 792 p.
6. Болотова Ю.А., Спицын В.Г., Рудомёткина М.Н. Распознавание автомобильных номеров на основе метода связанных компонент и иерархической временной сети // Компьютерная оптика. – 2015. – Т. 39, № 2. – С. 275-280.
7. Казанский Н.Л., Попов С.Б. Распределённая система технического зрения регистрации железнодорожных составов // Компьютерная оптика. – 2012. – Т. 36, № 3. – С. 419-428.
8. Изотов П.Ю., Суханов С.В., Головашкин Д.Л. Технология реализации нейросетевого алгоритма в среде CUDA на примере распознавания рукописных цифр // Компьютерная оптика. – 2010. – Т. 34, № 2. – С. 243-251.
9. Спицын В.Г., Болотова Ю.А., Фан Н.Х., Буй Т.Т.Ч. Применение вейвлет-преобразования Хаара, метода главных компонент и нейронных сетей для оптического распознавания символов на изображениях в присутствии импульсного шума // Компьютерная оптика. – 2016. – Т. 40, № 2. – С. 249-257. – DOI: 10.18287/2412-6179-2016-40-2-249-257.
10. Загинайло М.В., Фатхи В.А. Распознавание символов с помощью аппарата искусственных нейронных сетей // Инновации и инвестиции. – 2005. – № 5. – С. 145-147.
11. Рашид Т. Создаем нейронную сеть. – СПб.: ООО «Альфа-книга», 2017. – 272 с.
12. Фан Н.Х., Буй Т.Т.Ч., Спицын В.Г. Распознавание печатных текстов на основе применения вейвлет-преобразования и метода главных компонент // Известия Томского политехнического университета. – 2012. – Т. 36, № 5. – С. 154-157.
13. Miller E.G., Viola P.A. Ambiguity and constraint in mathematical expression recognition // in AAAI-98/IAAI-98 Proceedings, July 26-30, 1998, Madison, Wisconsin: AAAI, 1998. – P. 784-791.

14. Ong Kai Bin, Yew Kwang Hooi, Said Jadid Abdul Kadir, Haruhiro Fujita and Luqman Hakim Rosli. Enhanced Symbol Recognition based on Advanced Data Augmentation for Engineering Diagrams // International Journal of Advanced Computer Science and Applications (IJACSA). – 2022. – 13 (5). – <http://dx.doi.org/10.14569/IJACSA.2022.0130563>.
15. Bhanbhro H., Yew K.H., Kusakunniran W., Amur Z. A Symbol Recognition System for Single-Line Diagrams Developed Using a Deep-Learning Approach // Applied Sciences. – 2023. – 13. – P. 8816. – <https://doi.org/10.3390/app13158816>.
16. Moreno-García, C.F.; Elyan, E.; Jayne, C. Heuristics-Based Detection to Improve Text/Graphics Segmentation in Complex Engineering Drawings // In Proceedings of the Engineering Applications of Neural Networks: 18th International Conference (EANN 2017), Athens, Greece, 25–27 August 2017. – P. 87-98.
17. Pratt W.K. Digital image processing. – New York: Wiley, 1991. – 698 p.
18. Muthukrishnan R, Radha M. Contour selection algorithms for image segmentation // International Journal of Computer Science & Information Technology (IJCSIT). – 2014. –Vol. 3, No. 6. – P. 259-267.
19. Пойнтер Я. Программируем с PyTorch: Создание приложений глубокого обучения. – СПб.: Питер, 2020. – 256 с.
20. Liu Yuxi (Hayden). PyTorch 1.x Reinforcement Learning Cookbook. Over 60 recipes to design, develop, and deploy self-learning AI models using Python. – Birmingham–Mumbai: Packt, 2019. – 527 p.

## REFERENCES

1. Gorelik A.L., Skripkin V.A. Metody raspoznavaniya [Recognition methods]. Moscow: Vysshaya shkola, 1984, 208 p.
2. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016. Available at: <http://www.deeplearningbook.org>.
3. Chupinin Yu.G. Patent RF 2661750: MPK G06K 9/20. Raspoznavanie simvolov s ispol'zovaniem iskusstvennogo intellekta [Patent Ru No. 2661750, G06K 9/20. Character recognition using artificial intelligence]; Prior. 30.05.2017, Publ. 07/19/2018, Bul. No. 20.
4. Nikolenko S., Kadurin A., Arkhangel'skaya E. Glubokoe obuchenie. Pogruzenie v mir neyronnykh setey [Deep learning. Dive into the world of neural networks]. Saint Petersburg: Piter, 2021, 476 p.
5. Forsyth D.A., Ponce J. Computer Vision: A Modern Approach. 2nd ed. New Jersey: Prentice Hall, 2011, 792 p.
6. Bolotova Yu.A., Spitsyn V.G., Rudometkina M.N. Raspoznavanie avtomobil'nykh numerov na osnove metoda svyaznykh komponent i ierarkhicheskoy vremennoy seti [Recognition of license plates based on the method of connected components and a hierarchical time network], *Komp'yuternaya optika* [Computer Optics], 2015, Vol. 39, No. 2, pp. 275-280.
7. Kazanskiy N.L., Popov S.B. Raspredeleonnaya sistema tekhnicheskogo zreniya registratsii zheleznodorozhnykh sostavov [Distributed vision system for registration of railway trains], *Komp'yuternaya optika* [Computer Optics], 2012, Vol. 36, No. 3, pp. 419-428.
8. Izotov P.Yu., Sukhanov S.V., Golovashkin D.L. Tekhnologiya realizatsii neyrosetevogo algoritma v srede CUDA na primere raspoznavaniya rukopisnykh tsifr [The technology of implementing a neural network algorithm in the cuda environment using the example of handwritten digit recognition], *Komp'yuternaya optika* [Computer Optics], 2010, Vol. 34, No. 2, pp. 243-251.
9. Spitsyn V.G., Bolotova Yu.A., Fan N.Kh., Buy T.T.Ch. Primenenie veyvlet-preobrazovaniya Khaara, metoda glavnykh komponent i neyronnykh setey dlya opticheskogo raspoznavaniya simvolov na izobrazheniyakh v prisutstvii impul'snogo shuma [Application of the Haar wavelet transform, the principal component method and neural networks for optical character recognition in images in the presence of pulsed noise], *Komp'yuternaya optika* [Computer Optics], 2016, Vol. 40, No. 2, pp. 249-257. DOI: 10.18287/2412-6179-2016-40-2-249-257.
10. Zaginaylo M.V., Fatkhi V.A. Raspoznavanie simvolov s pomoshch'yu apparata iskusstvennykh neyronnykh setey [Character recognition using artificial neural networks], *Innovatsii i investitsii* [Innovations and Investments], 2005, No. 5, pp. 145-147.
11. Rashid T. Sozdaem neyronnyuyu set' [Make your own neural network]. Saint Petersburg: OOO «Al'fa-kniga», 2017, 272 p.
12. Fan N.Kh., Buy T.T.Ch., Spitsyn V.G. Raspoznavanie pechatnykh tekstov na osnove primeneniya veyvlet-preobrazovaniya i metoda glavnykh komponent [Recognition of printed texts based on the application of the wavelet transform and the principal component method], *Izvestiya Tomskogo politekhnicheskogo universiteta* [Proceedings of Tomsk Polytechnic University], 2012, Vol. 36, No. 5, pp. 154-157.

13. Miller E.G., Viola P.A. Ambiguity and constraint in mathematical expression recognition, in *AAAI-98/AAAI-98 Proceedings, July 26-30, 1998, Madison, Wisconsin: AAAI, 1998*, pp. 784-791.
14. Ong Kai Bin, Yew Kwang Hooi, Said Jadid Abdul Kadir, Haruhiro Fujita and Luqman Hakim Rosli. Enhanced Symbol Recognition based on Advanced Data Augmentation for Engineering Diagrams, *International Journal of Advanced Computer Science and Applications (IJACSA)*, 2022, 13 (5). Available at: <http://dx.doi.org/10.14569/IJACSA.2022.0130563>.
15. Bhanbhro H., Yew K.H., Kusakunniran W., Amur Z. A Symbol Recognition System for Single-Line Diagrams Developed Using a Deep-Learning Approach, *Applied Sciences*, 2023, 13, pp. 8816. Available at: <https://doi.org/10.3390/app13158816>.
16. Moreno-García, C.F.; Elyan, E.; Jayne, C. Heuristics-Based Detection to Improve Text/Graphics Segmentation in Complex Engineering Drawings, In *Proceedings of the Engineering Applications of Neural Networks: 18th International Conference (EANN 2017), Athens, Greece, 25–27 August 2017*, pp. 87-98.
17. Pratt W.K. Digital image processing. New York: Wiley, 1991, 698 p.
18. Muthukrishnan R, Radha M. Contour selection algorithms for image segmentation, *International Journal of Computer Science & Information Technology (IJCSIT)*, 2014, Vol. 3, No. 6, pp. 259-267.
19. Poynter Ya. Программирование с PyTorch: Создание приложений глубокого обучения [Programming PyTorch for Deep Learning]. Saint Petersburg: Piter, 2020, 256 p.
20. Liu Yuxi (Hayden). PyTorch 1.x Reinforcement Learning Cookbook. Over 60 recipes to design, develop, and deploy self-learning AI models using Python. Birmingham–Mumbai: Packt, 2019, 527 p.

**Безуглов Дмитрий Анатольевич** – Ростовский филиал Российской таможенной академии; e-mail: bezuglovda@mail.ru; г. Ростов-на-Дону, Россия; д.т.н.; профессор.

**Мищенко Марина Сергеевна** – Южный федеральный университет; e-mail: yourhuckleberrybtw@mail.ru; г. Ростов-на-Дону, Россия; студент.

**Мищенко Сергей Евгеньевич** – ФГУП «Ростовский научно-исследовательский институт радиосвязи»; e-mail: mihome@yandex.ru; г. Ростов-на-Дону, Россия; д.т.н.; профессор.

**Bezuglov Dmitry Anatolyevich** – Rostov branch of the Russian Customs Academy; e-mail: bezuglovda@mail.ru; Rostov-on-Don, Russia; dr. of eng. sc.; professor.

**Mishchenko Marina Sergeevna** – Southern Federal University; e-mail: yourhuckleberrybtw@mail.ru; Rostov-on-Don, Russia; student.

**Mishchenko Sergey Evgenievich** – FSUE Rostov Scientific Research Institute of Radio Communications; e-mail: mihome@yandex.ru; Rostov-on-Don, Russia; dr. of eng. sc.; professor.

УДК 004.67

DOI 10.18522/2311-3103-2025-3-144-159

**А.Г. Бондаренко, А.Г. Кравец**

## **ИДЕНТИФИКАЦИЯ КЛЮЧЕВЫХ ТЕХНОЛОГИЙ НА ОСНОВЕ СБОРА И АНАЛИЗА ДАННЫХ ИЗ ОТКРЫТЫХ РУССКОЯЗЫЧНЫХ ИСТОЧНИКОВ**

Данная статья посвящена разработке и апробации нового подхода к сбору, обработке и анализу открытых данных на русском языке для идентификации ключевых технологических направлений. Для решения задачи формирования и последующего анализа структурированных датасетов разработаны и программно реализованы методы веб-скрейпинга, обработки естественного языка и анализа временных рядов. Описанный в статье подход впервые применен для извлечения и структурирования информации из научных статей, новостных ресурсов и патентной документации на русском языке. В результате анализа полученного датасета научных публикаций выделены 30 наиболее часто упоминаемых биграмм и столько же триграмм технологических терминов. На основе анализа частотности биграмм и триграмм выделены ключевые технологические термины, которые затем использованы для комплексной фильтрации по ключевым технологиям. Комплексная фильтрация позволила осуществить поиск русскоязычных патентов и их сбор для дальнейшего анализа. В результате предварительной обработки полученной патентной информации сформированы временные ряды патентной активности. Программная система идентификации ключевых технологий реализована на JavaScript и Python с использованием библиотек Selenium