

НЕЙРОКОМПЬЮТИНГ В ТЕХНОЛОГИЧЕСКИХ ЦИКЛАХ БИБЛИОТЕК

УДК 004.8.032.26:02 + 004.93

<https://doi.org/10.33186/1027-3689-2026-4-159-176>

Распознавание печатных и рукописных текстов на основе нейронной сети

А. В. Фролов¹, Е. А. Верещагина²

¹*Морской государственный университет им. адмирала Г. И. Невельского,
Владивосток, Российская Федерация*

²*Дальневосточный Федеральный университет,
Владивосток, Российская Федерация*

¹alekandr.911@mail.ru, <https://orcid.org/0000-0002-7223-3642>

²vereschagina.ea@dfvu.ru

Аннотация. Рассматривается проблема машиночитаемости текста в век цифровых технологий. Оптическое распознавание символов (OCR) печатного и рукописного текста является одним из вариантов решения актуальной проблемы. Основные задачи данного метода – анализ существующих систем и алгоритмов распознавания, разработка собственного кода и его тестирование на различных шрифтах и почерках. Проблема OCR особенно актуальна в контексте распознавания рукописного текста, что подчеркивает необходимость создания эффективных решений с использованием современных технологий. В статье исследуется реализация одной из функций компьютерного зрения по обработке печатного и рукописного текста. Кроме того, рассматривается возможность улучшить распознавание за счет деления изображений на черно-белые и выделения дополнительных частей символов. Существует ряд систем, распознающих печатный текст с низкой вероятностью ошибки, но проблема распознавания рукописного текста на разных языках мира будет всегда актуальна. Не все существующие системы распознавания рукописного текста могут быть применены для печатного, особенно с использованием нейронных сетей. Такие системы чаще всего работают на основе признаковых или шаблонных методов.

Ключевые слова: нейронная сеть, текст, данные, будущее, технология, символы, OCR, код

Для цитирования: Фролов А. В., Верещагина Е. А. Распознавание печатных и рукописных текстов на основе нейронной сети // Научные и технические библиотеки. 2026. № 4. С. 159–176. <https://doi.org/10.33186/1027-3689-2026-4-159-176>

NEUROCOMPUTING IN LIBRARY TECHNOLOGICAL CYCLES

UDC 004.8.032.26:02 + 004.93

<https://doi.org/10.33186/1027-3689-2026-4-159-176>

Recognition of printed and handwritten texts based on the neural networks

Alexander V. Frolov¹ and Elena A. Vereschagina²

¹*Admiral Nevelskoy Maritime State University, Vladivostok, Russian Federation*

²*Far Eastern Federal University, Vladivostok, Russian Federation*

¹alekandr.911@mail.ru, <https://orcid.org/0000-0002-7223-3642>

²vereschagina.ea@dfvu.ru

Abstract. The authors examine the issues of text machine readability in the age of digital technologies. The optical character recognition (OCR) of printed texts and manuscripts offer solutions. The main tasks of the proposed method is to analyze existing recognition systems and algorithms, own code design and testing for various fonts The authors examine the computer vision function for handwritten and printed text processing. Besides, recognition can be improved by dividing images into the black and white, and highlighting symbol parts. Many systems recognize printed text at low error rate, however recognition of handwriting is a challenge for many global languages. Not every handwritten text recognition system can be applied to printed texts, especially with use of neural networks. Most often, such systems operate with feature- or template-driven methods.

Keywords: neural network, text, data, the future, technology, symbol, OCR, code

Cite: Frolov A. V., Vereshchagina E. A. Recognition of printed and handwritten texts based on the neural networks // Scientific and technical libraries. 2026. No. 4, pp. 159–176. <https://doi.org/10.33186/1027-3689-2026-4-159-176>

Введение

Цифровые технологии произвели революцию не только в способах упаковки, обработки, хранения и распространения информации, но и в способах поиска и доступа пользователей к ней. Оцифровка делает материалы доступными в электронном виде и улучшает доступ к ним и совместную работу. Цифровые проекты позволяют пользователям эффективно искать нужные материалы для обучения, преподавания и исследований из любой точки мира в любое время.

Оцифровка идет очень активно. Например, в 2025 г. в Президентской библиотеке основные источники архивных материалов предоставили Российский государственный исторический архив в Санкт-Петербурге и Государственный архив Российской Федерации в Москве. Такие архивы содержат большое количество рукописных материалов, зачастую на листах за долгое время хранения накопилось много дефектов, затрудняющих распознавание текстов обычными программами распознавания.

Оцифровкой материалов занимаются и различные коммерческие организации. Например, во Владивостоке уже несколько лет оцифровываются архивы Дальневосточного морского пароходства, FESCO (ПАО «ДВМП»). Как показывает практика, такие архивы обычно просто сканируются или фотографируются (качество бумажных документов, которым от ста до трехсот лет, не всегда позволяет загрузить их в сканер) и далее сохраняются в виде графических файлов, например, в формате JPEG. Такой подход затрудняет перевод документов в базу данных с последующим качественным поиском в ней.

Системы распознавания изображений включают различные технологии. ABBYY FineReader – одна из популярных программ, обеспечивающая точность распознавания (до 98%) и поддержку различных форма-

тов с сохранением структуры документа. SimpleOCR – бесплатное ПО, работает с английским и французским языками, но не поддерживает казахский и русский. FreeOCR (SODA PDF OCR) также бесплатно, работает с изображениями и PDF-файлами, поддерживает русский язык с 2010 г. OCRFeeder автоматизирует распознавание текста, анализируя макет документа и поддерживая различные форматы экспорта.

Общий алгоритм OCR включает захват кадра, предварительную обработку, локализацию и распознавание объекта. Важной частью алгоритма является сохранение изображения в виде матрицы пикселей. Основные методы OCR включают шаблонные, признаковые, структурные (топологические) методы и методы с использованием нейронных сетей.

Шаблонные методы распознают символы путем сравнения с ранее загруженными шаблонами. Признаковые основываются на сопоставлении изображению N-мерного вектора признаков. Структурные методы учитывают топологию символа, что позволяет игнорировать размер и шрифт текста, но затрудняет распознавание дефектных символов.

Методы с использованием нейронных сетей стали основой современных систем OCR. Нейронная сеть представляет собой математическую модель, имитирующую биологические нейронные сети. Для корректного функционирования сеть должна быть обучена на большом количестве изображений. Это обеспечивает высокую точность распознавания.

Алгоритм распознавания текста. Нейронные сети

Реализуемая программа для распознавания текста должна считывать получаемое изображение, детектировать символы и размещать их по секторам, а после производить распознавание и выдавать результат (рис. 1).

В обработку изображения входят все манипуляции, которые происходят до момента распознавания текста: захват изображения, предварительная обработка, сегментация, проверка, секторизация. При получении изображения в электронном формате для дальнейшего его анализа, редактирования и достижения желаемых результатов используется сканер или камера смартфона (рис. 2).



Рис. 1. Общий алгоритм с нейронной сетью

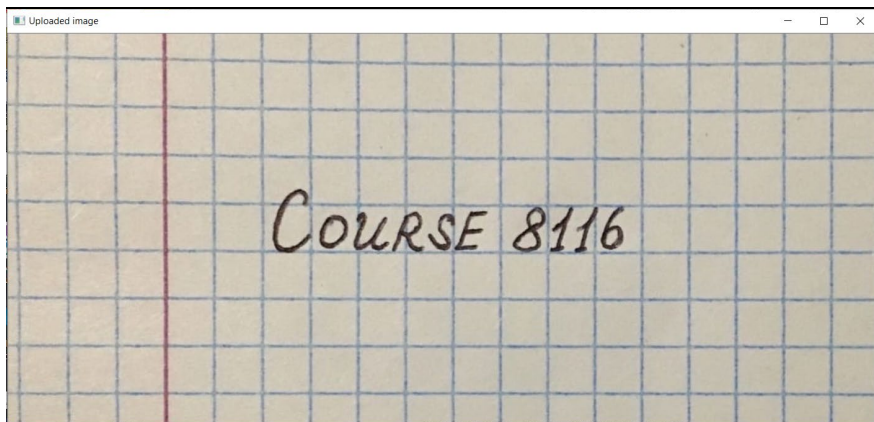


Рис. 2. Загруженное изображение

Предварительная обработка придает изображению серые тона, чтобы легче было производить бинаризацию, четко определяющую оттенки на черное и белое, или на 0 и 1 (рис. 3).

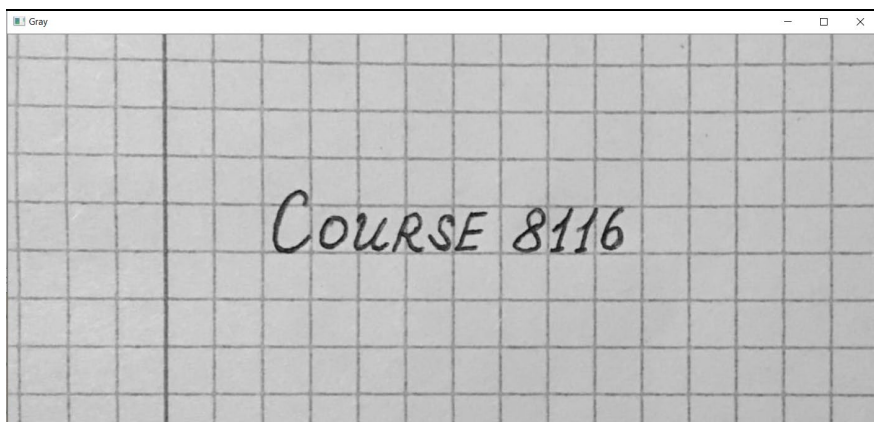


Рис. 3. Обработка в оттенках серого

Далее идет выделение трех видов контура: контура символа, внутреннего и внешнего. Контур символа проходит «поверх» самого символа, делая его зрительно чуть шире. Внешние представляют собой тер-

риторию, или сектор, в котором находится буква или цифра, ничто другое не заходит за эту границу (рис. 4, 5).



Рис. 4. Бинаризация

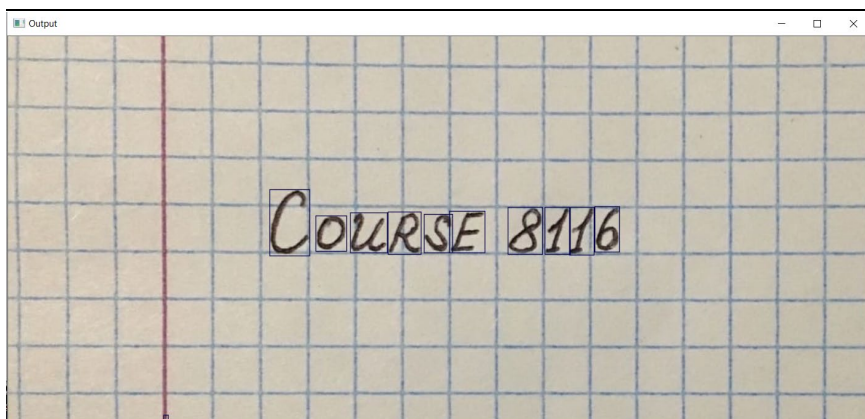


Рис. 5. Выделение контуров

После выделения контуров каждый сектор становится самостоятельным изображением. То есть из строки HELLO WORLD получается десять изображений определенного размера. Таким образом можно сохранить эти изображения для дальнейшего побуквенного распознавания.

Разработка программы распознавания символов

Использование нейронных сетей для распознавания символов требует выделения характерных черт. Эти черты затем поступают на вход нейронной сети для определения класса символа. Определение числа слов и нейронов в сети – нетривиальная задача, влияющая на результативность и трудоемкость процесса.

Обучение нейронной сети

Для обучения нейронной сети необходимо установить библиотеки TensorFlow и Keras. Важны совместимость версий TensorFlow и Python, а также подключение дополнительных библиотек для корректной работы TensorFlow.

Создание модели сети начинается с класса Sequential. В сверточной нейронной сети используется слой Convolution2D для свертки с ядром 3 x 3 и MaxPooling2D для уменьшения размеров (рис. 6). Два слоя Dense задают количество входов (784) и выходов (число цифр и букв).

```
def create_model():
    model = Sequential()
    model.add(Convolution2D(filters=32, kernel_size=(3, 3), padding='valid', input_shape=(28, 28, 1), activation='relu'))
    model.add(Convolution2D(filters=64, kernel_size=(3, 3), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.25))
    model.add(Flatten())
    model.add(Dense(784, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(len(position), activation='softmax'))
    model.compile(loss='categorical_crossentropy', optimizer='adadelta', metrics=['accuracy'])
    return model
```

Рис. 6. Создание модели

Для обучения используется набор данных EMNIST, содержащий рукописные символы, аналогичный набору MNIST. Набор включает 697932 блока данных для обучения и 116323 для тестирования, считываемых с помощью idx2numpy (рис. 7).

```
X_train = idx2numpy.convert_from_file(way + 'emnist-byclass-train-images-idx3-ubyte')
y_train = idx2numpy.convert_from_file(way + 'emnist-byclass-train-labels-idx1-ubyte')

X_test = idx2numpy.convert_from_file(way + 'emnist-byclass-test-images-idx3-ubyte')
y_test = idx2numpy.convert_from_file(way + 'emnist-byclass-test-labels-idx1-ubyte')

X_train = np.reshape(X_train, (X_train.shape[0], 28, 28, 1))
X_test = np.reshape(X_test, (X_test.shape[0], 28, 28, 1))
```

Рис. 7. Считывание данных для обучения нейронной сети

Технические параметры, такие как количество эпох, задаются для обучения. Процесс запускается функцией `fit`, а обученная модель сохраняется в файл `mosh.h5` (рис. 8 и 9).

```
model = create_model()
model.fit(X_train, x_train_cat, validation_data=(X_test, y_test_cat), callbacks=[learning_rate_reduction], batch_size=64, epochs=28)
model.save('mosh.h5')
```

Рис. 8. Запуск процесса обучения и сохранение файла

```
Instructions for updating:
If using Keras pass *constraint arguments to layers.
Train on 697932 samples, validate on 116323 samples
Epoch 1/28
72192/697932 [==>.....] - ETA: 45:38 - loss: 0.9545 - accuracy: 0.7207
```

Рис. 9. Процесс обучения нейронной сети

Во время работы программы консоль отображает состояние обучения, включая число образцов, тестов, текущую эпоху и время прохождения эпохи.

Для распознавания символов необходима библиотека `OpenCV` и ее функция `findContours`, позволяющая определить контуры. `OpenCV` содержит более 2500 алгоритмов для компьютерного зрения и машинного обучения, поддерживает интерфейсы на `Python`, `Java`, `C++` и `Matlab`. Для работы с `OpenCV` необходимо установить библиотеки (рис. 10).

```
import os
from cv2 import cv2
import matplotlib.pyplot as plt
from typing import List
from keras import backend as K
```

Рис. 10. Дополнение к библиотекам

Программа начинается с загрузки изображения командой `cv2.imread`, которая возвращает массив данных изображения. Команда `cv2.imshow` открывает окно с изображением, а следующая команда удерживает это окно открытым (рис. 11).

```

picture = input("Введите наименование изображения: ")
img = cv2.imread(picture)
cv2.imshow("Uploaded image", img)
cv2.waitKey(0)

```

Рис. 11. Считывание изображения

Предварительная обработка изображения

Изображение переводится в оттенки серого, затем бинаризуется, увеличивается, определяются контуры. Переход из серого в черно-белое: пиксели ярче значения $\text{int}(\text{light})$ становятся белыми, а темнее – черными (рис. 12).

```

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
ret, thresh = cv2.threshold(gray, int(light), 255, cv2.THRESH_BINARY)
img_erode = cv2.erode(thresh, np.ones((3, 3), np.uint8), iterations=1)
contours, hierarchy = cv2.findContours(img_erode, cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)

```

Рис. 12. Предварительная обработка изображения

Определение контуров

Создается список контуров букв. Определяются их внешние и внутренние контуры, оттенок и расположение, после этого они добавляются в список в размере 28 x 28. Проверяется правильность бинаризации, чтобы избежать ошибок распознавания. Далее контуры сортируются по координате X (рис. 13).

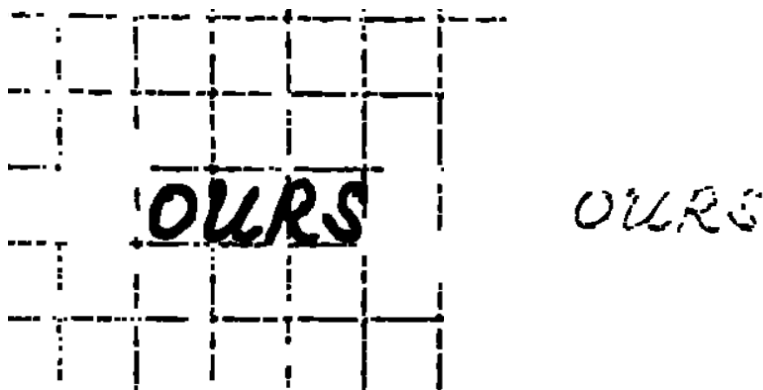


Рис. 13. Результат бинаризации при значении освещенности 100 и 45

Вывод символов

Можно вывести символы в формате изображения для проверки правильности подготовки и распознавания (рис. 14).

```
def treatment_img(model, img):  
    img_arr = np.expand_dims(img, axis=0)  
    img_arr = 1 - img_arr/255.0  
    img_arr[0] = np.rot90(img_arr[0], 3)  
    img_arr[0] = np.fliplr(img_arr[0])  
    img_arr = img_arr.reshape((1, 28, 28, 1))  
  
    result = model.predict_classes([img_arr])  
    return chr(position[result[0]])
```

Рис. 14. Редактирование результатов обработки

Запуск распознавания

Загружается модель, и вызывается функция `predict_classes`. Изображения из базы данных требуется повернуть. После этого вместо изображения получаем строку с результатом распознавания. Пробелы между символами расставляются, если расстояние между ними больше трети ширины символа. Все символы обрабатываются с использованием обученной нейронной сети (рис. 15).

```
model = keras.models.load_model('mosh.h5', compile=False)  
detective = img_to_str(model, 'picture')  
print("\nРаспознанный текст: " + detective)
```

Рис. 15. Подключение нейронной сети к распознаванию

Обучение нейронной сети является достаточно сложным и долгим процессом, хотя сам пользователь почти не принимает участия. В течение всего времени изучения было создано пять тестовых нейронных сетей. Первая была пробной: она проходила не всю базу данных, а только

десятую часть (коэффициент $k = 10$), при этом число нейронов было 512, а эпох – 30. Результатом было многократное допущение ошибок, самые частые из которых возникали при определении символов «1» и «7», «0» и «U», а в большинстве случаев выдавался результат, который вообще не сочетался с реальностью, например, при рукописных буквах «А В Н» результатом был «3 1 1». Такая сеть обучалась 2 часа.

Следующие три попытки давали одинаковый результат, но лучше первого. При этом увеличивался объем исследуемой базы данных. Пятая и последняя версия нейронной сети обучалась 23 часа 45 минут, при этом она проходила полностью весь датасет, имела 784 нейрона и 28 эпох (на каждую уходило 45–48 минут). Результат стал во много раз лучше, но некие погрешности также появлялись. В большинстве случаев программа путает букву O и цифру 0 (рис. 16).

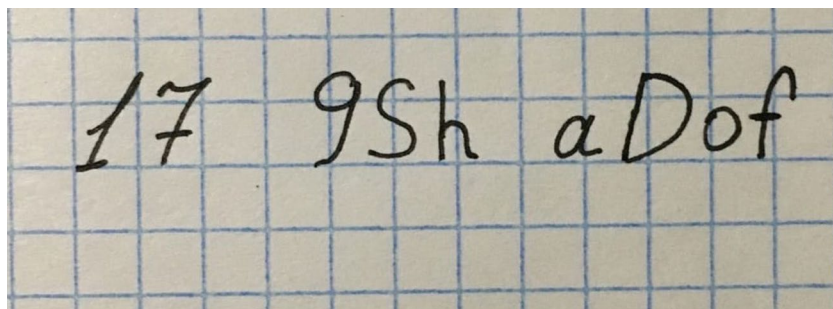


Рис. 16. Набор символов с наибольшей погрешностью при распознавании

На протяжении всей работы были использованы изображения как рукописного текста, так и печатного. Присутствуют погрешности, но от этого текст не перестает быть читабельным (рис. 17).

Были протестированы, а также сравнены с результатом рукописного текста популярные шрифты (рис. 17). Набор символов (рис. 16) был прочитан как «17 9Sh aDOf». Этот же набор был изучен на примере шести шрифтов. В ходе теста было обнаружено, что между размером загружаемого изображения и результата есть небольшая зависимость. Сведения о проделанной работе предоставлены в таблице.



Рис. 17. Тестируемые шрифты

Результат тестирования распознавания различных шрифтов

Шрифт	Размер пиксель)		
	1500 x 1500	1000 x 1000	500 x 500
Bodoni	17 ySh dUUF	17 ySh aDUF	17 9Sh UD0F
Calibri	17 9Sh 0D0t	17 gSh aU0t	17 9Sh aD0F
Comic Sans	11 9Sh a D0t	17 9Sh 0D0t	17 9Sh 0D0t
Futura	1 7 9Sh 0D0F	1 7 9Sh 0D0F	1 7 9Sh 0D0f
Helvetica	7 7 gSn aU0t	7 7 gSn aU0t	7 7 gSn aD0t
Times New Roman	1 7 9Sh gap0F	1 7 9Sh dD0F	17 9Sh aD0F

Как можно заметить, программе сложно отличить 0 от 0, строчная буква *f* почти не поддается прочтению. Есть ненужные пробелы, которые зависят от ширины символа, или постоянное неправильное написание одного и того же символа, как буква *a* у Futura. Но все это определяется строго самим шрифтом, к которому нейросеть не готова.

По результатам данного теста стоит сделать вывод, что нейронная сеть, обученная на примерах рукописных символов, подходит для распознавания печатных только для определенных шрифтов (например, Futura

или Times New Roman) или для некоторых символов. В ином случае ее стоит обучить дополнительной информацией. Также стоит отметить, что Helvetica хуже всех распознается, но входит в тройку самых популярных шрифтов.

Другой вывод заключается в обнаруженной зависимости распознавания текста от размера изображения. Чем меньше размер, тем меньше число пикселей, тем легче обнаружить каждый из них нейроном (рис. 18).

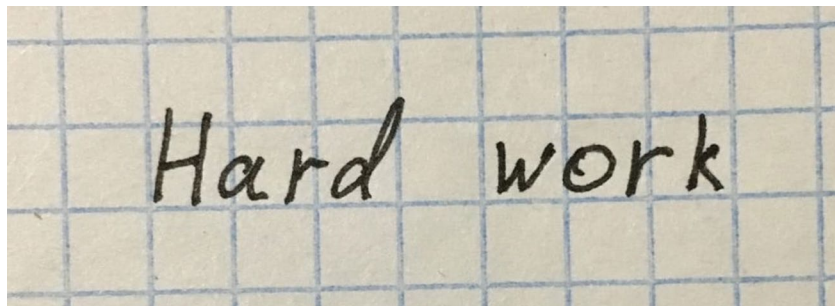


Рис. 18. Текст для распознавания

При всем этом распознавание рукописного текста дает положительные результаты. Например, изображение на рис. 18 распознано программой как «Hard WOrk».

Заключение

Проведенные исследования показывают, что метод распознавания рукописных текстов на основе нейронных сетей гораздо более точен, чем традиционные методы, основанные на классификаторах символов структурирования, и может стать основой для создания коммерческих продуктов в данной области.

Корпорация ABBYY с программой FineReader остается единственным серьезным производителем на этом рынке. Однако продукты ABBYY не обеспечивают высокой точности в нестандартных форматах, а уровень надежности распознавания рукописей невысок. Например, если пользователь попытается распознать несколько разных образцов почерка от разных людей в одной и той же версии программы, программа не сможет выдать результат.

Использование нейронных сетей в обработке рукописного и печатного текста – современное и действенное средство. Оно оптимизирует, улучшает и ускоряет процессы обработки данных и их дальнейший анализ, редактирование и т. д. При этом необходимо учитывать, что использование нейронных сетей требует наличия квалифицированных специалистов и высококачественных данных.

Список источников

1. **Распознавание** символов. URL: https://ru.wikipedia.org/wiki/Оптическое_распознавание_символов.
2. **ABBYY** FineReader. URL: <https://www.abbyy.com/ru-ru/>.
3. **SimpleOCR** system. URL: <http://freeanalogs.ru/SimpleOCR>.
4. **Free-OCR** system. URL: <http://www.free-ocr.com/>.
5. **Tesseract-OCR** source code. URL: <https://github.com/tesseract-ocr>
6. **OCRFeeder** – Распознавание текста. URL: <https://pingvinus.ru/program/ocrfeeder>.
7. **Методы** распознавания текста. URL: <https://habr.com/ru/post/220077/>.
8. **Алгоритмы** постобработки результатов распознавания текстовых полей. URL : <https://habr.com/ru/companies/smartengines/articles/498520/>.
9. **Шпаргалка** по OpenCV – Python. URL: <https://tproger.ru/translations/opencv-python-guide/>.
10. **Основные** операции с изображениями OpenCV. URL: <https://arboook.com/kompyuternoe-zrenie/osnovnye-operatsii-s-izobrazheniyami-v-opencv-3-python/>.
11. **Keras**. URL: <https://keras.io>.
12. **Фролов А. В., Титова А. А., Верещагина Е. А.** Мониторинг рисков и безопасности перевозок нефти-газа железнодорожным транспортом // Экологические системы и приборы. 2022. № 9. С. 51–53. DOI 10.25791/esip.9.2022.1319. EDN IHZQEE.
13. **Использование** графов знаний для формирования интеллектуального капитала компании / Гореликов Р. С., Ярмонов А. С., Добржинский Ю. В. и др. // Промышленные АСУ и контроллеры. 2024. № 1. С. 23–27. DOI 10.25791/asu.1.2024.1481. EDN BBFJHA.
14. **Ярмонов А. С., Верещагина Е. А., Фролов А. В.** Тестирование программ и математическая модель поиска ошибок в программном комплексе // Промышленные АСУ и контроллеры. 2024. № 2. С. 42–44. DOI 10.25791/asu.2.2024.1489. EDN XHGXHA.
15. **Оптимизация** обнаружения вредоносного ПО в сетях IoT: Использование распределенных вычислений с учетом ресурсов для повышения безопасности / Ярмонов А. С., Гореликов Р. С., Явтуховский Е. Ю. и др. // Промышленные АСУ и контроллеры. 2024. № 5. С. 14–22. EDN WHUOWI.

16. **Костыркин Н. В., Верещагина Е. А., Фролов А. В.** Анализ влияния DRM-системы на производительность в компьютерных играх // Промышленные АСУ и контроллеры. 2023. № 10. С. 42–50. DOI 10.25791/asu.10.2023.1467. EDN HLLARC.
17. **Самоидентификация** узлов сенсорной сети на основе интернета вещей с помощью алгоритма изоморфизма графов / Ярмонов А. С., Гореликов Р. С., Верещагина Е. А. и др. // Промышленные АСУ и контроллеры. 2023. № 11. С. 3–14. DOI 10.25791/asu.11.2023.1468. EDN LKTZYD.
18. **Фролов А. В., Дымченко Ю. В., Верещагина Е. А.** Защита облачных корпоративных данных // Промышленные АСУ и контроллеры. 2023. № 3. С. 37–40. DOI 10.25791/asu.3.2023.1425. EDN EONGXQ.
19. **Разработка** системы дистанционного управления с идентификацией личности по отпечатку пальца / Фролов А. В., Дымченко Ю. В., Янголь М. С., Верещагина Е. А. // Промышленные АСУ и контроллеры. 2023. № 4. С. 18–26. DOI 10.25791/asu.4.2023.1432. EDN HKRLIS.
20. **Интерпретация** работы Internet of Things на примере строительного объекта / Ярмонов А. С., Гореликов Р. С., Верещагина Е. А., Фролов А. В. // Промышленные АСУ и контроллеры. 2023. № 6. С. 16–23. DOI 10.25791/asu.6.2023.1441. EDN ALKZQS.
21. **Использование** технологии блокчейн в системах электронного документооборота / Гореликов Р. С., Ярмонов А. С., Добржинский Ю. В. и др. // Промышленные АСУ и контроллеры. 2023. № 7. С. 41–46. DOI 10.25791/asu.7.2023.1449. EDN BUOGJH.
22. **Фролов А. В., Верещагина Е. А., Золкин А. Л.** Big Data в библиотеках и научных исследованиях // Вестник Российского нового университета. Серия: Сложные системы: модели, анализ и управление. 2023. № 2. С. 167–172. DOI 10.18137/RNU.V9I187.23.02.P.167. EDN RRFXYK.
23. **Фролов А. В., Титова А. А., Верещагина Е. А.** Big Data и виртуальные ЦОД // Промышленные АСУ и контроллеры. 2022. № 2. С. 25–29. DOI 10.25791/asu.2.2022.1347. EDN AJUXPV.
24. **Фролов А. В., Титова А. А., Верещагина Е. А.** Электронное голосование без рассекречивания конфиденциальной информации // Промышленные АСУ и контроллеры. 2022. № 6. С. 41–48. DOI 10.25791/asu.6.2022.1372. EDN JKTTKF.

References

1. **Raspoznavanie** simvolov. URL: https://ru.wikipedia.org/wiki/Оптическое_распознавание_символов.
2. **ABBYY** FineReader. URL: <https://www.abby.com/ru-ru/>.
3. **SimpleOCR** system. URL: <http://freeanalogs.ru/SimpleOCR>.
4. **Free-OCR** system. URL: <http://www.free-ocr.com/>.
5. **Tesseract-OCR** source code. URL: <https://github.com/tesseract-ocr>

6. **OCRFeeder** – Rozpoznávanie teksta. URL: <https://pingvinus.ru/program/ocrfeeder>.
7. **Metody** rozpoznávaníia teksta. URL: <https://habr.com/ru/post/220077/>.
8. **Algoritmy** postobrabotki rezul'tatov rozpoznávaníia tekstovy'kh polei'. URL: <https://habr.com/ru/companies/smartengines/articles/498520/>.
9. **Shpargalka** po OpenCV – Python. URL: <https://tproger.ru/translations/opencv-python-guide/>.
10. **Osnovny'e** operacii s izobrazheníiami OpenCV. URL: <https://arboook.com/kompyuternoie-zrenie/osnovnyie-operatsii-s-izobrazheniyami-v-opencv-3-python/>.
11. **Keras**. URL: <https://keras.io>.
12. **Frolov A. V., Titova A. A., Vereshchagina E. A.** Monitoring riskov i bezopasnosti perevozok nefiti-gaza zheleznodorozhny'm transportom // E'kologicheskie sistemy i pribory'. 2022. № 9. S. 51–53. DOI 10.25791/esip.9.2022.1319. EDN IHZQEE.
13. **Ispol'zovanie** grafov znanií` dlia formirovaniia intellektual'nogo kapitala kompanii / Gorelikov R. S., Iarmonov A. S., Dobrzhinskii` Iu. V. i dr. // Promy'shlenny'e ASU i kontrolyery'. 2024. № 1. S. 23–27. DOI 10.25791/asu.1.2024.1481. EDN BBFJHA.
14. **Iarmonov A. S., Vereshchagina E. A., Frolov A. V.** Testirovanie programm i matematicheskaia model' poiska oshibok v programmnom komplekse // Promy'shlenny'e ASU i kontrolyery'. 2024. № 2. S. 42–44. DOI 10.25791/asu.2.2024.1489. EDN XIHGXA.
15. **Optimizatciia** obnaruzheniia vredonosnogo PO v setiakh IoT: Ispol'zovanie raspredelenny'kh vy'chislenii' s uchetoм resursov dlia povy'sheniia bezopasnosti / Iarmonov A. S., Gorelikov R. S., Iavtuhovskii` E. Iu. i dr. // Promy'shlenny'e ASU i kontrolyery'. 2024. № 5. S. 14–22. EDN WHUOWI.
16. **Kosty'rkín N. V., Vereshchagina E. A., Frolov A. V.** Analiz vliianiia DRM-sistemy` na proizvoditel'nost' v komp'uterny'kh igrakh // Promy'shlenny'e ASU i kontrolyery'. 2023. № 10. S. 42–50. DOI 10.25791/asu.10.2023.1467. EDN HLLARC.
17. **Samoidentifikatciia** uzlov sensornoi` seti na osnove interneta veshchei` s pomoshch'iu algoritma izomorfizma grafov / Iarmonov A. S., Gorelikov R. S., Vereshchagina E. A. i dr. // Promy'shlenny'e ASU i kontrolyery'. 2023. № 11. S. 3–14. DOI 10.25791/asu.11.2023.1468. EDN LKTZYD.
18. **Frolov A. V., Dy'mchenko Iu. V., Vereshchagina E. A.** Zashchita oblachny'kh korporativny'kh danny'kh // Promy'shlenny'e ASU i kontrolyery'. 2023. № 3. S. 37–40. DOI 10.25791/asu.3.2023.1425. EDN EONGXQ.
19. **Razrabotka** sistemy` distantsionnogo upravleniia s identifikatciei` lichnosti po oтpechatku pal'tca / Frolov A. V., Dy'mchenko Iu. V., Iangoľ M. C., Vereshchagina E. A. // Promy'shlenny'e ASU i kontrolyery'. 2023. № 4. S. 18–26. DOI 10.25791/asu.4.2023.1432. EDN HKRLIS.
20. **Interpretatciia** raboty` Internet of Things na primere stroitel'nogo ob'ekta / Iarmonov A. S., Gorelikov R. S., Vereshchagina E. A., Frolov A. V. // Promy'shlenny'e ASU i kontrolyery'. 2023. № 6. S. 16–23. DOI 10.25791/asu.6.2023.1441. EDN ALKZQS.
21. **Ispol'zovanie** tekhnologii blokchei'n v sistemakh e'lektronnogo dokumentooborota / Gorelikov R. S., Iarmonov A. S., Dobrzhinskii` Iu. V. i dr. // Promy'shlenny'e ASU i kontrolyery'. 2023. № 7. S. 41–46. DOI 10.25791/asu.7.2023.1449. EDN BUOGJH.

22. **Frolov A. V., Vereshchagina E. A., Zolkin A. L.** Big Data v bibliotekakh i nauchny`kh issledovaniyakh // Vestnyk Rossijskogo novogo universiteta. Seriya: Slozhny`e sistemy`: modeli, analiz i upravlenie. 2023. № 2. S. 167–172. DOI 10.18137/RNU.V9187.23.02.P.167. EDN RRFXY.
23. **Frolov A. V., Titova A. A., Vereshchagina E. A.** Big Data i virtual`ny`e TCOД // Promy`shlenny`e ASU i kontrolyery`. 2022. № 2. S. 25–29. DOI 10.25791/asu.2.2022.1347. EDN AJXPV.
24. **Frolov A. V., Titova A. A., Vereshchagina E. A.** E`lektronnoe golosovanie bez rassekrechivaniia konfidentcial`noi` informacii // Promy`shlenny`e ASU i kontrolyery`. 2022. № 6. S. 41–48. DOI 10.25791/asu.6.2022.1372. EDN JKTKF.

Информация об авторах / Authors

Фролов Александр Владимирович – начальник отдела информационных технологий Морского государственного университета им. адмирала Г. И. Невельского, Владивосток, Российская Федерация
alekandr.911@mail.ru

Alexander V. Frolov – Head, Information Technologies Department, Admiral Nevelskoy Maritime State University, Vladivostok, Russian Federation
alekandr.911@mail.ru

Верещагина Елена Александровна – канд. техн. наук, доцент, доцент департамента информационной безопасности Института математики и компьютерных технологий Дальневосточного Федерального университета, Владивосток, Российская Федерация
vereschagina.ea@dfvu.ru

Elena A. Vereshchagina – Cand. Sc. (Engineering), Associate Professor, Information Security Department, Institute of Mathematics and Computer Technology, Far Eastern Federal University, Vladivostok, Russian Federation
vereschagina.ea@dfvu.ru