НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
январь–февраль 2024 Том 24 № 1 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
January–February 2024 Vol. 24 No 1 http://ntv.ifmo.ru/en/
ISSN 2226-1494 (print) ISSN 2500-0373 (online)

НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

doi: 10.17586/2226-1494-2024-24-1-90-100
УДК 004.8, 004.93

# Автоматизация распознавания сложной текстовой CAPTCHA с применением условной генеративно-состязательной нейронной сети

**Александр Сергеевич Задорожный[1], Анастасия Андреевна Корепанова[2], Максим Викторович Абрамов[3]✉, Артем Азатович Сабреков[4]**

[1] Санкт-Петербургский государственный университет, Санкт-Петербург, 199034, Российская Федерация
[2,3,4] Санкт-Петербургский Федеральный исследовательский центр Российской академии наук, Санкт-Петербург, 199178, Российская Федерация

[1] alexander.zadorozhnyy@yandex.ru, https://orcid.org/0009-0004-8903-7244
[2] aak@mail.ru, https://orcid.org/0000-0003-2962-8670
[3] mva@dscs.pro✉, https://orcid.org/0000-0002-5476-3025
[4] aas@dscs.pro, https://orcid.org/0009-0000-8378-563X

**Аннотация**

**Введение.** С быстрым развитием интернет-технологий проблемы сетевой безопасности продолжают обостряться. Так одним из наиболее распространенных методов поддержания безопасности и предотвращения вредоносных атак на интернет-ресурсы является полностью автоматизированный публичный тест Тьюринга (Completely Automated Public Turing test to tell Computers and Humans Apart, CAPTCHA). CAPTCHA чаще всего состоит из некоторого защитного кода, для обхода которого необходимо выполнить простую задачу. Однако наиболее широко используемым видом CAPTCHA до сих пор остается текстовый тип. В последние годы развитие компьютерного зрения и, в частности, нейронных сетей позволило снизить устойчивость к взлому текстовых CAPTCHA. Однако безопасность и надежность сложных CAPTCHA, содержащих много шума и искажений, все еще недостаточно изучена. Предметом данного исследования является устойчивость к распознаванию CAPTCHA. Особенность CAPTCHA — использование большого количества разнообразных искажений, причем на каждом отдельном изображении применяется свой набор искажений. При наличии искажений человеческий глаз не всегда может распознать, что изображено на фотографии. Данная работа состоит в тестировании устойчивости сайтов, использующих исследуемую CAPTCHA, к автоматизированному решению. Полученная методика тестирования может быть применена для последующей разработки рекомендаций по повышению эффективности механизмов защиты. **Метод.** В работе продемонстрирован новый метод к распознаванию CAPTCHA с использованием синтетического генератора и дискриминатора условно генеративно-состязательной архитектуры, а также программы-декодера, представляющей собой обученную сверточную нейронную сеть. **Основные результаты.** Результатом работы являются реализованные генератор и дискриминатор совместно с программой-декодером, решающей данный тип CAPTCHA. Точность распознавания построенной модели составила 63 % на изначально очень ограниченном наборе данных, что показывает риски информационной безопасности, которые могут нести сайты, использующие подобный вид CAPTCHA. **Обсуждение.** Несмотря на то, что получена точность распознавания ниже, чем у некоторых существующих методов (70–99 %), данный результат обладает следующей значимостью: продемонстрирован неизученный ранее в подобных работах вид CAPTCHA, а также предложен новый метод по распознаванию текстовых CAPTCHA.

**Ключевые слова**

текстовая CAPTCHA, глубокое обучение, условная генеративно-состязательная нейронная сеть, CGAN, CNN, информационная безопасность

**Благодарности**

Работа выполнена в рамках проекта по государственному заданию Санкт-Петербургского Федерального исследовательского центра Российской академии наук «Санкт-Петербургский институт информатики и автоматизации Российской академии наук» № FFZF-2022-0003.

**Ссылка для цитирования:** Задорожный А.С., Корепанова А.А., Абрамов М.В., Сабреков А.А. Автоматизация распознавания сложной текстовой CAPTCHA с применением условной генеративно-состязательной нейронной сети // Научно-технический вестник информационных технологий, механики и оптики. 2024. Т. 24, № 1. С. 90–100. doi: 10.17586/2226-1494-2024-24-1-90-100

# Automation of complex text CAPTCHA recognition using conditional generative adversarial networks

**Alexander S. Zadorozhnyy[1], Anastasia A. Korepanova[2], Maxim V. Abramov[3]✉, Artem A. Sabrekov[4]**

[1] St. Petersburg State University (SPbSU), Saint Petersburg, 199034, Russian Federation
[2,3,4] Saint Petersburg Federal Research Center of the Russian Academy of Sciences, Saint Petersburg, 199178, Russian Federation

[1] alexander.zadorozhnyy@yandex.ru, https://orcid.org/0009-0004-8903-7244
[2] aak@mail.ru, https://orcid.org/0000-0003-2962-8670
[3] mva@dscs.pro✉, https://orcid.org/0000-0002-5476-3025
[4] aas@dscs.pro, https://orcid.org/0009-0000-8378-563X

**Abstract**
With the rapid development of Internet technologies, the problems of network security continue to worsen. So, one of the most common methods of maintaining security and preventing malicious attacks is CAPTCHA (fully automated public Turing test). CAPTCHA most often consists of some kind of security code, to bypass which it is necessary to perform a simple task, such as entering a word displayed in an image, solving a basic arithmetic equation, etc. However, the most widely used type of CAPTCHA is still the text type. In the recent years, the development of computer vision and, in particular, neural networks has contributed to a decrease in the resistance to hacking of text CAPTCHA. However, the security and resistance to recognition of complex CAPTCHA containing a lot of noise and distortion is still insufficiently studied. This study examines CAPTCHA, the distinctive feature of which is the use of a large number of different distortions, and each individual image uses its own different set of distortions, that is why even the human eye cannot always recognize what is depicted in the photo. The purpose of this work is to assess the security of sites using the CAPTCHA text type by testing their resistance to an automated solution. This testing will be used for the subsequent development of recommendations for improving the effectiveness of protection mechanisms. The result of the work is an implemented synthetic generator and discriminator of the CGAN architecture, as well as a decoder program, which is a trained convolutional neural network that solves this type of CAPTCHA. The recognition accuracy of the model constructed in the article was 63 % on an initially very limited data set, which shows the information security risks that sites using a similar type of CAPTCHA can carry.

**Keywords**
text-based CAPTCHAs, deep learning, conditional generative adversarial network, CGAN, CNN, information security

**Acknowledgements**
The work was carried out within the framework of the project under the state assignment of SPC RAS SPIIRAS no. FFZF-2022-0003.

**For citation:** Zadorozhnyy A.S., Korepanova A.A., Abramov M.V., Sabrekov A.A. Automation of complex text CAPTCHA recognition using conditional generative adversarial networks. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2024, vol. 24, no. 1, pp. 90–100 (in Russian). doi: 10.17586/2226-1494-2024-24-1-90-100

## Введение

В области информационной безопасности используется множество методов защиты от автоматизированного сбора данных [1], включая полностью автоматизированный публичный тест Тьюринга (Completely Automated Public Turing Tests to Tell Computers and Humans Apart, CAPTCHA). CAPTCHA представляет собой защитный код, обход которого требует выполнения простого для человека задания, такого как ввод искаженных символов с изображения, решение арифметических уравнений или другие подобные задачи. Текстовая CAPTCHA, содержащая изображение с искаженными символами для ввода, является одним из наиболее распространенных типов.

Однако уровень безопасности сайтов с текстовой CAPTCHA снижается из-за развития технологий компьютерного зрения. Исследования показывают уязвимость многих видов искажений CAPTCHA перед современными системами. Однако устойчивость к распознаванию сложных CAPTCHA с большим числом шумов и различных искажений все еще неисследована до конца [2]. Сложные CAPTCHA состоят из изображений с разнообразными символами, подвергающимися различным искажениям и шумам, но остающимися решаемыми для человека. Рассматриваемый в данном исследовании вид CAPTCHA относится именно к такому типу. Несмотря на то, что она состоит из четырех случайных символов, каждый из которых может быть как одной из букв латинского алфавита, так и цифрой, CAPTCHA обладает большой вариативностью: от расстояния между буквами до наложенного поверх паттерна. К разным изображениям применяется суммарно более 30 видов искажений, включая искажение перспективы, растяжение или другие методы, которые делают CAPTCHA практически невозможной для распознания. Следствием такого разнообразия является не только трудность различить, что представлено на изображении даже для человеческого глаза, но и сложность в обучении модели распознавания CAPTCHA, которая бы одинаково хорошо распознавала все изображения, подвергшиеся разным видам искажений. Вопрос, насколько действительно сложно злоумышленнику построить систему прохождения такой системы защиты, ранее не исследовался.

Цель работы — исследование устойчивости к распознаванию сложной CAPTCHA с большим числом различных искажений. В перспективе полученные ре-

зультаты помогут сформировать рекомендации для повышения защищенности сайтов, использующих текстовые CAPTCHA. В настоящей работе изучен первый этап в достижении данной цели — из исследуемого множества возможных изображений CAPTCHA выбрано подмножество, соответствующее одному типу искажений и исследованы способы обучения решателя CAPTCHA такого вида. В дальнейшем полученные результаты будут использованы для построения решателя для полного набора изображений CAPTCHA. Теоретическая значимость работы состоит в предложении нового метода распознавания символов на изображении CAPTCHA. Например, метод может быть полезен в ситуациях, когда необходимо обеспечить баланс между сложностью распознавания для машины и удобством для человека. Отметим, что значимость работы заключается в потенциале использования полученных результатов для формулирования рекомендаций по повышению защищенности сайтов, использующих текстовые CAPTCHA.

## Обзор работ

В ходе работы были изучены ведущие сервисы для решения CAPTCHA: CapMonster[1], Azcaptcha[2], RuCaptcha[3], Captcha.Guru[4], XEvil[5], BestCaptchaSolver[6], SolveCaptcha[7], AntiCaptcha[8]. Данные сервисы разделены на автоматизированные решатели (XEvil, CapMonster, Azcaptcha, Captcha.Guru) с использованием оптического распознавания символов (Optical Character Recognition, OCR) и на привлекающие наемных сотрудников (BestCaptchaSolver, SolveCaptcha, RuCaptcha, AntiCaptcha), тратя до 20 с на одну CAPTCHA. Подробные результаты сравнения сервисов, включая их стоимость и методы, представлены в табл. 1.

Таким образом, как это видно из табл. 1, имеющиеся сервисы либо не способны распознать имеющийся набор данных, либо прибегают к помощи человека и не способны делать это самостоятельно, что показывает защищенность данной CAPTCHA от простейших атак с использованием готовых инструментов.

В научных исследованиях последних лет устоявшимся подходом к разработке инструмента для распознавания символьных CAPTCHA является применение нейронных сетей [3–5]. В работе [3] предложена архитектура и методика обучения сверточной нейронной сети (Convolutional Neural Network, CNN) для распознавания CAPTCHA, сгенерированной с помощью библиотеки Python ImageCaptcha Library. Точность распознавания предложенного решения по метрике accuracy составила 98,9 %. В [2] разработана модель, основанная на долгой краткосрочной памяти, для распознавания моделей CAPTCHA разных видов, точность модели на CAPTCHA с 50 популярных сайтов находится в промежутке от 74,8 % до 97,3 %. Работа [4] посвящена распознаванию CAPTCHA, составленных из символов индийских языков хинди, бенгали и тамильского, с помощью модели CNN и достигнутой точностью 81 % на сгенерированном вручную наборе данных. Также для решения подобной задачи была предложена архитектура CNN с соединениями быстрого доступа (shortcut connections или skip-connections) [5] и итоговой точностью до 98,82 %. В работе [6] была применена функция потерь Focal loss для повышения точности распознавания на популярных примерах CAPTCHA и получена точность до 99 %.

Обучение модели CNN с различными модификациями демонстрирует высокую точность в данной задаче, однако требует большого объема данных для обучения: в работах [3–6] количество изображений CAPTCHA с известными ответами, использованных для обучения моделей, превышает 10 000. В [2] отмечено, что в зависимости от сложности CAPTCHA, для обучения модели, способной показывать точность более 80 %, может потребоваться несколько тысяч размеченных изображений.

Для уменьшения числа требуемых данных, размеченных вручную, в изученных научных работах были предложены следующие подходы: использование генеративно-состязательных сетей (Generative Adversarial Nets, GAN) для увеличения размера набора данных и использование алгоритма neural style transfer [7, 8] для упрощения изображения CAPTCHA. В работе [7] предложена модель, которая с помощью механизма neural style transfer упрощает изображения CAPTCHA, убирая искажения, после чего символы становится гораздо проще распознать с помощью другой модели. Точность распознавания после упрощения изображений с помощью предложенного решения составила 66–89 % на разных типах CAPTCHA. В [9–11] проведены исследования, направленные на изучение возможности применения GAN для решения CAPTCHA. Результаты позволили получить на небольшом обучающем наборе данных в количестве 500 размеченных вручную изображений точность от 40 до 100 % распознавания в зависимости от сложности CAPTCHA.

Таким образом, обзор показал, что существующие аналоги либо требуют ручного решения, что затратно по времени, либо используют OCR и решают только

[1] Capmonster.cloud: сервис для распознавания CAPTCHA [Электронный ресурс]. URL: https://capmonster.cloud (дата обращения: 10.11.2023).

[2] Azcaptcha: сервис для распознавания CAPTCHA [Электронный ресурс]. URL: https://azcaptcha.com (дата обращения: 10.11.2023).

[3] ruCaptcha: сервис для распознавания CAPTCHA [Электронный ресурс]. URL: https://rucaptcha.com (дата обращения: 10.11.2023).

[4] Captcha.Guru: сервис для распознавания CAPTCHA [Электронный ресурс]. URL: https://captcha.guru (дата обращения: 10.11.2023).

[5] XEvil: сервис для распознавания CAPTCHA [Электронный ресурс]. URL: https://xevil.net (дата обращения: 11.11.2023).

[6] BestCaptchaSolver: сервис для распознавания CAPTCHA [Электронный ресурс]. URL: https://bestcaptchasolver.com (дата обращения: 11.11.2023).

[7] SolveCaptcha: сервис для распознавания CAPTCHA [Электронный ресурс]. URL: https://solvecaptcha.com (дата обращения: 11.11.2023).

[8] AntiCaptcha: сервис для распознавания CAPTCHA. URL: https://anti-captcha.com (дата обращения: 11.11.2023).

*Таблица 1*. Сравнение сервисов, предоставляющих услуги решения CAPTCHA
*Table 1*. Comparison of services providing CAPTCHA solution

| Сервис | Метод | Стоимость, долл. США за 1000 шт. | Решает ли рассматриваемую CAPTCHA |
|---|---|---|---|
| XEvil | OCR | 0 | нет |
| CapMonster | OCR | 0,3 | нет |
| Azcaptcha | OCR | 0,4 | нет |
| Captcha.Guru | OCR | 0,1 | да |
| BestCaptchaSolver | человек | 0,6 | да |
| SolveCaptcha | человек | 0,5–1,0 | да |
| RuCaptcha | человек | 0,6 | да |
| AntiCaptcha | человек | 0,5 | да |

простые текстовые CAPTCHA, не решая конкретно задачу, поставленную в настоящей работе. А также, что ни одно из исследований не посвящено данному типу CAPTCHA с его разнообразием шумов и искажений. Исходя из этого, было предложено обучить собственную модель, основанную на существующих решениях, для оценки защищенности этого типа CAPTCHA от специально обученных моделей.

## Постановка задачи

Задача, решаемая в данной работе, состоит в оценке защищенности к распознаванию сложной CAPTCHA с большим числом возможных искажений. Цель достигается через разработку модели для автоматического распознавания.

Исследуемая CAPTCHA представляет собой изображение с набором символов, каждый из которых может быть буквой из латинского алфавита или цифрой, с различными искажениями и шумами. Исходный набор данных состоял из 18 020 пар изображений и их расшифровок, собранных вручную (рис. 1). Как видно, набор CAPTCHA обладает существенной вариативностью: перечеркнутые линии, сильный шум, наложенный поверх узор, переменное расстояние между символами, а также разная ширина каждой отдельной буквы. При этом не было обнаружено никакой зависимости между размером символов и расстоянием между ними — все это сильно затрудняет дальнейшую обработку изображений и обучение единой модели, которая могла бы распознать все CAPTCHA.

Для упрощения обучения решателя в рамках исследования использована кластеризация набора данных на группы изображений с однородными шумами. Применена модель VGG16 [12] в сочетании с алгоритмом кластеризации KMeans [13]. Эксперимент определил оптимальное количество кластеров — 30. Во время эксперимента была обучена модель на одном из кластеров, включающем 600 изображений (рис. 2). Кластер выбран как наиболее визуально простой для тестирования обучаемости решателя, но содержащий сложные шумы, такие как нечеткие контуры и слипание символов. В ходе работы было проверено обучение модели на данном типе изображений и выполнена оценка возможности адаптации решения для других кластеров в дальнейшем.

## Методы

В настоящей работе, из-за отсутствия готовых инструментов, было решено применить машинное обучение. В большинстве существующих решений для поставленной задачи используются CNN [14–16]. По этой причине, на основании результатов анализа существующих архитектур [14], были выбраны модели ResNet [17, 18] и Inception, обладающие большой рас-

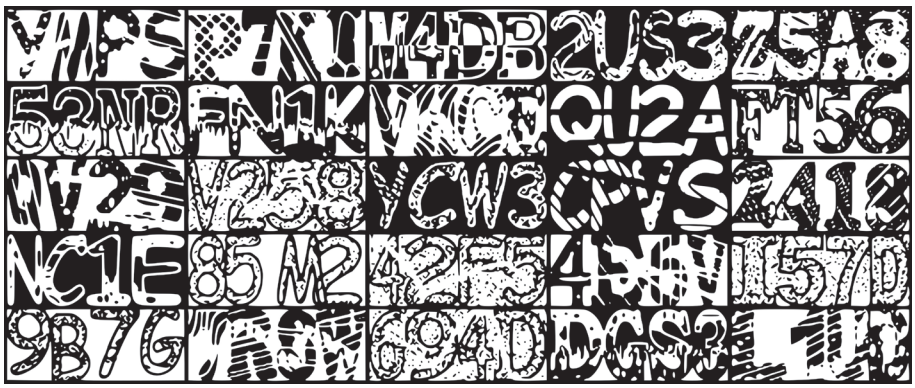

*Рис. 1*. Примеры CAPTCHA
*Fig. 1*. CAPTCHA examples

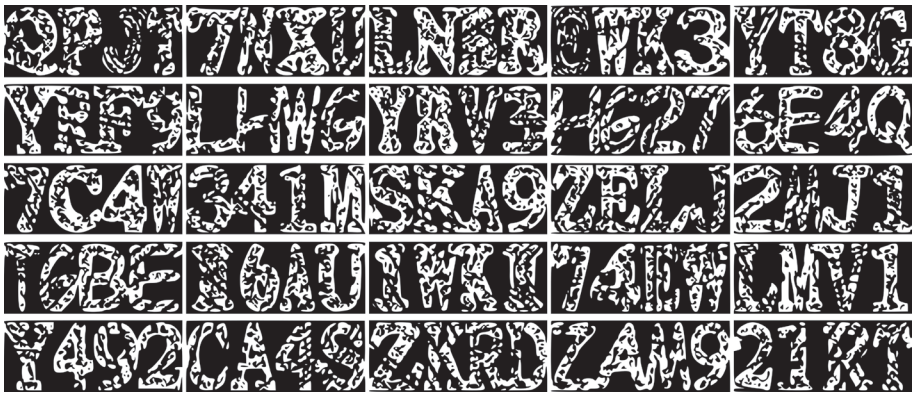

*Рис. 2*. Кластер, выбранный для обучения
*Fig. 2*. Cluster selected for training

пространенностью и хорошей точностью для решения различных задач [19, 20].

В работе [2] отмечено, что для корректного и эффективного обучения нейронной сети может потребоваться более 10 000 размеченных данных. А сбор и ручная маркировка такого количества реальных CAPTCHA является очень трудозатратным процессом. Таким образом, несмотря на то что CNN требуют большого количества образцов для обучения, они являются наиболее эффективными как для задач распознавания объектов, так и для решения задачи распознавания CAPTCHA. В то же время, чтобы устранить проблему с нехваткой обучающих данных, было решено использовать архитектуру GAN.

Архитектура GAN [21] представляет собой две ключевые модели: генеративную сеть, создающую синтетические примеры, и дискриминативную сеть, различающую синтезированные и реальные примеры. Однако в отличии от существующих работ, которые также основаны на архитектуре GAN [7, 8], предлагается не просто генерировать похожие изображения, а создавать полностью уникальные примеры из текстовой расшифровки. Тем самым модель становится независимой от разнообразия начальных данных, воссоздавая форму и стиль любых входных данных. Для этого возможно воспользоваться условно генеративно-состязательной архитектурой (CGAN). CGAN позволяет не только генерировать CAPTCHA с заданными искажениями и случайным набором букв, но и указывать требуемую расшифровку. В архитектуре CGAN генератор (Generator) и дискриминатор (Discriminator) настраиваются с использованием дополнительной информации, такой как метка класса (рис. 3). На данной схеме $y$, $y_1$, $y_2$, ..., $y_n$ — некоторая метка, которую нужно преобразовать в изображение, необработанный вход (шум), представлен в виде вектора случайных чисел — **z**. Выходное значение, которое получается в результате работы генератора, — $x^*$, а $x^*|y$ — $x^*$ с учетом или при условии $y$, σ обозначена некоторая функция активации, Training dataset — датасет с тренировочными данными, где настоящие примеры с метками показаны как $(x, y)$.

В результате итоговая архитектура решения состоит из двух отдельных компонентов: CGAN, который генерирует реалистичные примеры, неотличимые от оригинальных CAPTCHA, причем с известными расшифровками, и CNN, который, имея достаточное количество пар пример-метка от предыдущего компонента после обучения на искусственных данных, сможет эффективно распознавать уже реальные CAPTCHA.

### Программная реализация

Опишем реализацию двух ключевых компонент предлагаемого подхода к распознаванию CAPTCHA. Для этого в качестве основного языка выберем Python. Используем стандартные библиотеки для работы с изображениями Pillow и OpenCV, первая из которых позволяет удобно работать с изображениями, а вторая

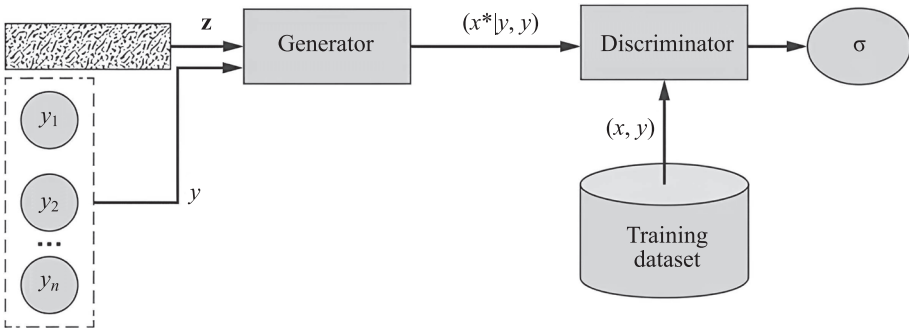


*Рис. 3*. CGAN архитектура
*Fig. 3*. CGAN architecture

применяет алгоритмы компьютерного зрения для распознавания и описания объектов на фотографии. Для работы непосредственно с обучением нейронных сетей применим библиотеку TensorFlow.

В первую очередь рассмотрим процесс обучения модели CGAN, которая позволяет генерировать синтетические образцы CAPTCHA с соответствующими расшифровками. Во-вторых, продемонстрируем решатель CAPTCHA на основе архитектуры ResNet-34, адаптированной под ее распознавание.

**Реализация CGAN-архитектуры.** Рассмотрим процесс обучения предложенной модели генерации новых CAPTCHA. В качестве меток для генерируемых примеров CGAN-архитектурой можно было бы использовать расшифровки CAPTCHA целиком. Однако данный тип CAPTCHA представляет собой последовательность из четырех символов, где каждый символ является буквой латинского алфавита или цифрой. Таким образом, всего возможных последовательностей символов $34^4$ или 1 336 336. Чтобы CGAN научилась генерировать все возможные изображения CAPTCHA, набор данных должен содержать несколько примеров каждой метки, т. е. быть в разы больше этого числа. Из чего можно сделать вывод, что обучить CGAN модель на количестве исходных данных в 600 изображений с метками такого типа невозможно. По этой причине было решено научиться отдельно генерировать каждый из 34 символов и далее собрать из них CAPTCHA для обучения модели. Таким образом, вместо $34^4$ используем всего 34 метки. Вручную был размечен набор данных, в котором содержится по 32 образца каждого символа.

Процесс построения CGAN сети состоит из трех основных этапов.

Этап 1. Генератор CAPTCHA. Чтобы формализовать определение, назовем меткой текстовый символ, который нужно преобразовать в изображение — $y$. А необработанный вход (шум), представленный в виде вектора случайных чисел, которые генератор использует в качестве отправной точки для синтеза поддельных примеров — **z**. Выходное значение, которое получается в результате работы генератора, — $x^*$. Тогда их комбинация дает условный $G(z, y) = x^*|y$ (читается как «$x^*$ с учетом или при условии $y$»). Цель этого поддельного примера — выглядеть для дискриминатора как можно ближе к реальному примеру для данной метки (в нашем случае в качестве метки выступают символы латинского алфавита или цифры). Наилучшего результата удалось добиться при помощи CNN, состоящей из трех сверточных слоев и функцией активации выходного слоя — Sigmoid. Отметим, что генератор обновляет свои параметры, минимизируя следующую функцию потерь:

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log_2(p(y_i)) + (1 - y_i)\log_2(1 - (p(y_i)), \qquad (1)$$

где $N$ — количество данных, на основе которых вычисляется функция потерь; $p(x)$ — функция вероятности; $H_p(q)$ — логистическая функция потерь (binary cross-entropy).

Этап 2. Дискриминатор CAPTCHA. Дискриминатор получает на вход как настоящие примеры с метками $(x, y)$, так и поддельные примеры с меткой, которая использовалась для их генерации $(x^*|y, y)$. На синтетических примерах он учится распознавать фальшивые изображения. Метки $y$ ему необходимы, чтобы научиться проверять, действительно ли сгенерированное изображение соответствует изначальному заданному символу. Таким образом, цель дискриминатора заключается в том, чтобы определить, не только является ли образ, полученный генератором, реальным или сгенерированным, но и в том, совпадает ли полученный результат с указанной меткой. Для этого была использована нейронная сеть, состоящая из трех сверточных слоев с функцией активации LeakyRelu и одного полносвязного слоя с той же функцией активации (1) для BackPropogation, что и для генератора. В качестве ответа дискриминатор выдает одно число — вероятность того, что входные данные являются реальной совпадающей парой, т. е. символом из реального набора данных, соответствующего его метке. Вероятность вычисляется при помощи сигмоидальной функции активации.

Этап 3. Обучение. Был использован оптимизатор Adam с коэффициентом скорости обучения 0,0003 для обучения генератора CAPTCHA. Общая цель обучения соответствует общему подходу GAN [22], однако с некоторыми небольшими изменениями:

$$\min_G \max_D V(D, G) =$$
$$= E_{x\sim p_{data(x)}}[\log_2 D(x, y)] + E_{z\sim p_z(z)}[\log_2(1 - D(G(z, y)))],$$

где $E_x$ представляет собой математическое ожидание; $p_{data(x)}$ — распределение реальных данных, а следовательно $E_{x\sim p_{data(x)}}$ — математическое ожидание логарифма вероятности того, что дискриминатор правильно классифицирует некоторые данные $x$, где $x$ выбирается из определенного распределения; $V(D, G)$ — функция потерь для обучения CGAN. Таким образом, генератор $G$ пытается минимизировать разницу между сгенерированной парой изображения и метки и реальной, в то время как дискриминатор $D$ пытается максимизировать ее. Во время обновления параметров одного из них параметры другого фиксировались.

**Пример.** Рассмотрим в качестве примера одно из изображений CAPTCHA, которое было составлено из сгенерированных символов. На рис. 4 видно, что стиль сгенерированной CAPTCHA соответствует оригинальной, при этом начертания букв различаются. Это позволяет воссоздать неограниченное количество похожих друг на друга, но не идентичных экземпляров для дальнейшего обучения, тем самым сократив количество времени на сбор и структуризацию данных.

**Создание решателя CAPTCHA.** Процесс создания решателя разделим на три этапа.

Этап 1. Подготовка обучающих данных. На том основании, что генерируются отдельные символы, а не вся CAPTCHA целиком, необходимо научиться соединять их обратно в одно изображение. В связи с этим было решено получившиеся изображения символов случайным образом расширять или сужать, а после

*Рис. 4*. Сравнение оригинальной (*a*) и сгенерированной нейросетью (*b*) CAPTCHA
*Fig. 4*. Comparison of the original (*a*) and generated by neural network (*b*) CAPTCHA

соединять в одну CAPTCHA. Для этого было применено полиномиальное распределение, которое позволяет выбрать $n$ чисел суммы $m$. Ширина изображений CAPTCHA исследуемого вида составляет 96 пикселов и содержит 4 символа, вследствие чего, каждый из сгенерированных символов имеет ширину в 24 пиксела. Далее случайным образом выбиралась ширина каждого нового изображения символа таким образом, чтобы в сумме они давали 96, т. е. $n = 4$, а $m = 96$. После применения данного подхода к полученным ранее синтетическим одиночным изображениям $40 \times 24 \times 1$ и последовательного их объединения, были получены изображения размером $40 \times 96 \times 1$, содержащие нужное количество стилистически неотличимых от оригинала символов, при этом находящихся на случайном расстоянии друг от друга и имеющих случайную ширину.

Этап 2. Выбор способа хранения. Существует два варианта работы с данными: можно непрерывно генерировать образцы во время обучения или предварительно создать достаточное количество. В данной работе выбран второй подход из-за значительного увеличения скорости обучения. При этом использовано: для обучения — 5000 оригинальных изображений, полученных путем клонирования нескольких сотен исходных экземпляров, совместно с 40 000 синтетических CAPTCHA, для валидации — 2000 оригинальных и 10 000 сгенерированных изображений.

Этап 3. Поиск оптимальной архитектуры. На данном этапе выбрана архитектура CNN для распознавания полученных CAPTCHA. Сначала были протестированы архитектуры AlexNet [23], VGG19 [12], DenseNet [24], Unet [25]. Выбор данных архитектур обоснован их популярностью в области компьютерного зрения и распознавания образов, высокой точностью распознавания и результатами на известных наборах данных, таких как ImageNet. Однако на данных исследуемого набора вышеперечисленные архитектуры за различное количество эпох не смогли достичь точности выше 10 %, несмотря на то что на синтетических CAPTCHA показатели спустя несколько первых эпох стремились к 100 %.

Для решения поставленной задачи была выбрана архитектура XCeption [26], которая справилась с задачей намного лучше, но наилучшие результаты были получены при помощи архитектуры ResNet-34 [27]. Результаты исследования архитектур приведены в разделе «Вычислительный эксперимент». Приведем процесс обучения более детально. Сеть состоит из шести частей: входного уровня, четырех остаточных блоков и выходного уровня. Внутри остаточных блоков содержится по три, четыре, шесть и три сверточных блока, каждый из которых, в свою очередь, состоит из двух сверточных слоев. Особенность этой сети состоит в том, что у каждого такого блока есть свой «обходной путь», который напрямую связывает вход с выходом. Код реализованной модели выложен в GitHub-репозитории[1].

### Вычислительный эксперимент

Выполним анализ результатов как метода синтеза обучающего множества, так и метода распознавания реализованных нейронных сетей на одном из кластеров исследуемой CAPTCHA.

**Синтез обучающего множества.** В данной работе использованы различные методы для генерации синтетических изображений, однако наилучших результатов удалось добиться при помощи архитектур GAN и CGAN. Синтетические изображения CAPTCHA, полученные с помощью GAN, сохраняют контуры букв, однако, в свою очередь не обладают схожей текстурой: серый фон, инверсия цветов и слияние символов. Все эти особенности подчеркивают нарушение структуры синтезированных образцов по сравнению с оригинальной CAPTCHA (рис. 5).

В настоящем исследовании использованы стандартные параметры обучения архитектуры GAN. Отметим, что применение данной архитектуры и различных экспериментов с варьированием гиперпараметров в разных диапазонах ($50 \leq$ latent space(размерность скрытого пространства) $\leq 200$; $3 \leq$ количество слоев $\leq 10$; $10^{-6} \leq$ $\leq$ коэффициент скорости обучения $\leq 10^{-2}$), не привели к существенным изменениям результатов. Для генератора изображений и дискриминатора, ответственного за распознавание CAPTCHA, использованы следующие гиперпараметры: размерность входных данных для генератора равна 100, а для дискриминатора — $40 \times 96 \times 1$; количество слоев генератора — 3. Применены следующие функции активации генератора: ReLU для скрытых слоев; Sigmoid для выходного слоя; коэффициент скорости обучения генератора равен $2 \cdot 10^{-5}$.

Как видно на представленных изображениях (рис. 4, *b* и рис. 5), CGAN преуспевает в создании более реалистичных образцов CAPTCHA, которые при этом полностью сохраняют текстуру оригинала. А полученные результаты подчеркивают не только возможность автоматизированного распознавания сложных текстовых CAPTCHA с использованием CGAN, но и полез-

[1] GitHub репозиторий с реализованным приложением [Электронный ресурс]. URL: https://github.com/ticslab/CGAN_CAPTCHA_SOLVER (дата обращения: 20.12.2023).

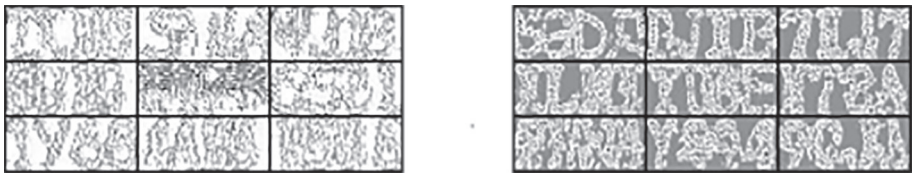

*Рис. 5*. Примеры CAPTCHA, сгенерированные при помощи GAN, на разных промежуточных этапах обучения
*Fig. 5*. CAPTCHA examples generated using GAN at various intermediate stages of training

*Таблица 2*. Наивысшие показатели метрик решателя CAPTCHA в зависимости от выбранной архитектуры
*Table 2.* The highest metrics of the CAPTCHA solver depending on the chosen architecture

| Архитектура | Точность работы генератора, % | Точность работы распознавателя, % | Скорость, с |
|---|---|---|---|
| AlexNet | 99,2 | 3,1 | 320,4 |
| VGG19 | 98,7 | 5,8 | 301,2 |
| DenseNet | 99,3 | 7,4 | 334,3 |
| Unet | 97,8 | 9,4 | 320,8 |
| Xception | 98,1 | 58,1 | 430,2 |
| ResNet-34 | 100 | 63,1 | 419,1 |

ность данного метода в контексте создания обучающего множества с высококачественными синтезированными образцами.

**Распознавание CAPTCHA**

**Условия эксперимента.** Характеристики машины, на которой проводилось тестирование: процессор Intel(R) Core (TM) i5-10300H, видеокарта NVIDIA GeForce GTX 1650 Ti, размер оперативной памяти 8GB.

**Метрики.** В рамках эксперимента была замерена точность (accuracy) распознавания 1000 сгенерированных (точность работы генератора CAPTCHA, %) и 1000 оригинальных (точность работы распознавателя CAPTCHA, %) образцов, а также скорость распознавания 5000 произвольных CAPTCHA (скорость, с).

Для вычисления точности распознавания использована следующая формула:

$$\text{Точность (Accuracy)} = \frac{\text{Количество верно распознанных CAPTCHA}}{\text{Общее количество тестовых CAPTCHA}} \cdot 100\ \%.$$

Результаты тестирования разных архитектур нейронных сетей для распознавания CAPTCHA представлены в табл. 2.

Таким образом, наилучшие показатели точности на исследуемых данных получены при использовании архитектуры ResNet-34 за 30 эпох с оптимизатором Adam, шагом обучения $10^{-6}$ и функцией активации Categorical Cross-Entropy.

## Обсуждение

В работе разработано решение для текстовых CAPTCHA на основе технологии CGAN. Результаты исследования доказали потенциал эффективного применения генеративных нейронных сетей в данной области, особенно можно выделить высокую конфигурируемость и гибкость CGAN. Это позволяет настраивать распознавание CAPTCHA для различных изображений и контекстов, а также обеспечивает преимущества в адаптации к разнообразным стилям CAPTCHA. Эффективное использование CGAN также упрощает адаптацию к изменяющимся условиям и стилям CAPTCHA, сокращая время и ресурсы, затрачиваемые на обучение, по сравнению с традиционными методами. Таким образом, применение CGAN в области распознавания текстовых CAPTCHA открывает новые перспективы для автоматизированного анализа и обработки визуальных данных.

Дальнейшие исследования будут направлены на улучшение способности распознавателя CAPTCHA справляться с различными шумами и искажениями. Это включает в себя изучение различных методов обработки изображений и генерации данных. Также будет исследовано совместное обучение генератора синтетических образцов и распознавателя CAPTCHA для улучшения точности и сокращения времени обучения. В будущем планируется использование других моделей машинного обучения, включая алгебраические байесовские сети [28–30]. Реализованный инструмент обеспечивает легкое расширение набора распознаваемых CAPTCHA, независимо от их сложности и зашумленности, что позволяет адаптироваться к постоянно меняющимся схемам CAPTCHA.

## Заключение

В работе изучена защищенность компаний, использующих конкретный тип CAPTCHA сложного вида от автоматического решения. В исследуемом типе CAPTCHA есть более 30 возможных искажений, что затрудняет использование готовых инструментов и обучение модели для ее распознавания. Рассмотрен новый подход к решению текстовых CAPTCHA на основе условной генеративно-состязательной и сверточной нейронных сетей. Несмотря на то, что точность рас-

познавания составляет 63 %, что ниже, чем у некоторых существующих методов (70–99 %), предложенное решение имеет ряд преимуществ. Во-первых, оно демонстрирует эффективное использование генеративных нейронных сетей в этой области. Другими словами, в отличие от аналогов, которые основаны на архитектуре генеративно-состязательных сетей, реализованный инструмент позволяет не просто генерировать незначительно отличающиеся внешне от начальных данных группы изображений, а создавать полностью уникальные образцы с заранее заданным стилем и расшифровкой. Во-вторых, новый подход требует значительно меньшего количества реальных CAPTCHA, что снижает необходимость в человеческом участии в процессе обучения нейронной сети и делает подход более практичным и экономически эффективным.

Таким образом, реализованный инструмент, несмотря на текущую точность в 63 %, обладает значительным потенциалом для улучшения и расширения, так как он способен адаптироваться к постоянно меняющимся схемам CAPTCHA, что делает его гибким и универсальным решением. Полученные результаты показывают, что CAPTCHA представленного формата является потенциально уязвимой к автоматическому распознаванию. В связи с этим в рамках дальнейшей работы будет исследоваться, сколько ресурсов требуется для повышения уязвимости и распознавания CAPTCHA со всеми типами искажений, чтобы на основе полученных результатов сформулировать рекомендации для сайтов, использующих текстовые CAPTCHA, для повышения их защищенности.

**Литература**


1. Корепанова А.А., Бушмелев Ф.В., Сабреков А.А. Технологии парсинга на Node.js в задаче агрегации сведений и оценки параметров грузовых маршрутов посредством извлечения данных из открытых источников // Компьютерные инструменты в образовании. 2021. № 3. С. 41–56. https://doi.org/10.32603/2071-2340-2021-3-41-56
2. Zi Y., Gao H., Cheng Z., Liu Y. An end-to-end attack on text CAPTCHAs // IEEE Transactions on Information Forensics and Security. 2019. V. 15. P. 753–766. https://doi.org/10.1109/TIFS.2019.2928622
3. Noury Z., Rezaei M. Deep-CAPTCHA: A deep learning based CAPTCHA solver for vulnerability assessment // ERN: Neural Networks & Related Topics (Topic). 2020. https://doi.org/10.2139/ssrn.3633354
4. Sahil Ahmed S., Anand K.M. Convolution neural network-based CAPTCHA recognition for indic languages // Advances in Intelligent Systems and Computing. 2021. V. 1407. P. 493–502. https://doi.org/10.1007/978-981-16-0171-2_46
5. Lu S., Huang K., Meraj T., Rauf H.T. A novel CAPTCHA solver framework using deep skipping Convolutional Neural Networks // PeerJ Computer Science. 2022. V. 8. P. e879. https://doi.org/10.7717/peerj-cs.879
6. Wang Z., Shi P. CAPTCHA recognition method based on CNN with focal loss // Complexity. 2021. V. 2021. P. 6641329. https://doi.org/10.1155/2021/6641329
7. Chen J., Luo X., Zhu L., Zhang Q., Gan Y. Handwritten CAPTCHA recognizer: a text CAPTCHA breaking method based on style transfer network // Multimedia Tools and Applications. 2023. V. 82. N 9. P. 13025–13043. https://doi.org/10.1007/s11042-021-11485-9
8. Bostik O., Horak K., Kratochvila L., Zemcik T., Bilik S. Semi-supervised deep learning approach to break common CAPTCHAs // Neural Computing and Applications. 2021. V. 33. N 20. P. 13333–13343. https://doi.org/10.1007/s00521-021-05957-0
9. Le T.A., Baydin A.G., Zinkov R., Wood F. Using synthetic data to train neural networks is model-based reasoning // Proc. of the 2017 International Joint Conference on Neural Networks (IJCNN). 2017. P. 3514–3521. https://doi.org/10.1109/IJCNN.2017.7966298
10. Wang Y., Wei Y., Zhang M., Liu Y., Wang B. Make complex captchas simple: a fast text CAPTCHA solver based on a small number of samples // Information Sciences. 2021. V. 578. P. 181–194. https://doi.org/10.1016/j.ins.2021.07.040
11. Li C., Chen X., Wang H., Wang P., Zhang Y., Wang W. End-to-end attack on text-based CAPTCHAs based on cycle-consistent generative adversarial network // Neurocomputing. 2021. V. 433. P. 223–236. https://doi.org/10.1016/j.neucom.2020.11.057
12. Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition // arXiv. 2014. arXiv:1409.1556. https://doi.org/10.48550/arXiv.1409.1556
13. Hartigan J.A., Wong M.A. Algorithm AS 136: A k-means clustering algorithm // Journal of the Royal Statistical Society. Series C (Applied Statistics). 1979. V. 28. N 1. P. 100–108. https://doi.org/10.2307/2346830


**References**


1. Korepanova A.A., Bushmelev F.V., Sabrekov A.A. Node.js parsing technologies in the task of aggregating information and evaluating the parameters of cargo routes by extracting data from open sources. *Computer Tools in Education Journal*, 2021, no. 3, pp. 41–56. (in Russian). https://doi.org/10.32603/2071-2340-2021-3-41-56
2. Zi Y., Gao H., Cheng Z., Liu Y. An end-to-end attack on text CAPTCHAs. *IEEE Transactions on Information Forensics and Security*, 2019, vol. 15, pp. 753–766. https://doi.org/10.1109/TIFS.2019.2928622
3. Noury Z., Rezaei M. Deep-CAPTCHA: A deep learning based CAPTCHA solver for vulnerability assessment. *ERN: Neural Networks & Related Topics (Topic)*, 2020. https://doi.org/10.2139/ssrn.3633354
4. Sahil Ahmed S., Anand K.M. Convolution neural network-based CAPTCHA recognition for indic languages. *Advances in Intelligent Systems and Computing*, 2021, vol. 1407, pp. 493–502. https://doi.org/10.1007/978-981-16-0171-2_46
5. Lu S., Huang K., Meraj T., Rauf H.T. A novel CAPTCHA solver framework using deep skipping Convolutional Neural Networks. *PeerJ Computer Science*, 2022, vol. 8, pp. e879. https://doi.org/10.7717/peerj-cs.879
6. Wang Z., Shi P. CAPTCHA recognition method based on CNN with focal loss. *Complexity*, 2021, vol. 2021, pp. 6641329. https://doi.org/10.1155/2021/6641329
7. Chen J., Luo X., Zhu L., Zhang Q., Gan Y. Handwritten CAPTCHA recognizer: a text CAPTCHA breaking method based on style transfer network. *Multimedia Tools and Applications*, 2023, vol. 82, no. 9, pp. 13025–13043. https://doi.org/10.1007/s11042-021-11485-9
8. Bostik O., Horak K., Kratochvila L., Zemcik T., Bilik S. Semi-supervised deep learning approach to break common CAPTCHAs. *Neural Computing and Applications*, 2021, vol. 33, no. 20, pp. 13333–13343. https://doi.org/10.1007/s00521-021-05957-0
9. Le T.A., Baydin A.G., Zinkov R., Wood F. Using synthetic data to train neural networks is model-based reasoning. *Proc. of the 2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 3514–3521. https://doi.org/10.1109/IJCNN.2017.7966298
10. Wang Y., Wei Y., Zhang M., Liu Y., Wang B. Make complex captchas simple: a fast text CAPTCHA solver based on a small number of samples. *Information Sciences*, 2021, vol. 578, pp. 181–194. https://doi.org/10.1016/j.ins.2021.07.040
11. Li C., Chen X., Wang H., Wang P., Zhang Y., Wang W. End-to-end attack on text-based CAPTCHAs based on cycle-consistent generative adversarial network. *Neurocomputing*, 2021, vol. 433, pp. 223–236. https://doi.org/10.1016/j.neucom.2020.11.057
12. Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv*, 2014, arXiv:1409.1556. https://doi.org/10.48550/arXiv.1409.1556
13. Hartigan J.A., Wong M.A. Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 1979, vol. 28, no. 1, pp. 100–108. https://doi.org/10.2307/2346830

14. Khan A., Sohail A., Zahoora U., Qureshi A.S. A survey of the recent architectures of deep convolutional neural networks // Artificial Intelligence Review. 2020. V. 53. N 8. P. 5455–5516. https://doi.org/10.1007/s10462-020-09825-6
15. Oliseenko V., Abramov M. Identification of user profiles in online social networks: a combined approach with face recognition // Journal of Physics: Conference Series. 2021. V. 1864. P. 012119. https://doi.org/10.1088/1742-6596/1864/1/012119
16. Bushmelev F., Khlobystova A., Abramov M., Livshits L. Deep machine learning techniques in the problem of estimating the expression of psychological characteristics of a social media user // Studies in Systems, Decision and Control. 2023. V. 457. P. 315–324. https://doi.org/10.1007/978-3-031-22938-1_22
17. Shafiq M., Gu Z. Deep residual learning for image recognition: a survey // Applied Sciences. 2022. V. 12. N 18. P. 8972. https://doi.org/10.3390/app12188972
18. Hossen M.I., Hei X. A low-cost attack against the hcaptcha system // Proc. of the 2021 IEEE Security and Privacy Workshops (SPW). 2021. P. 422–431. https://doi.org/10.1109/SPW53761.2021.00061
19. Kapoor A., Shah R., Bhuva R., Pandit T. Understanding inception network architecture for image classification: Technical Report. 2020. https://doi.org/10.13140/RG.2.2.16212.35204
20. Mittal S., Kaushik P., Hashmi S., Kumar K. Robust real time breaking of image CAPTCHAs using inception v3 model // Proc. of the 2018 Eleventh International Conference on Contemporary Computing (IC3). 2018. P. 1–5. https://doi.org/10.1109/IC3.2018.8530607
21. Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair Sh., Courville A., Bengio Y. Generative Adversarial Networks // Communications of the ACM. 2020. V. 63. N 11. P. 139–144. https://doi.org/10.1145/3422622
22. Mirza M., Osindero S. Conditional generative Adversarial Nets // arXiv. 2014. arXiv:1411.1784. https://doi.org/10.48550/arXiv.1411.1784
23. Krizhevsky A., Sutskever I., Hinton G.E. ImageNet classification with deep convolutional neural networks // Communications of the ACM. 2017. V. 60. N 6. P. 84–90. https://doi.org/10.1145/3065386
24. Huang G., Liu Z., Van Der Maaten L., Weinberger K.Q. Densely connected convolutional networks // Proc. of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017. P. 2261–2269. https://doi.org/10.1109/CVPR.2017.243
25. Ronneberger O., Fischer P., Brox T. U-Net: Convolutional networks for biomedical image segmentation // Lecture Notes in Computer Science. 2015. V. 9351. P. 234–241. https://doi.org/10.1007/978-3-319-24574-4_28
26. Chollet F. Xception: Deep learning with depthwise separable convolutions // Proc. of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017. P. 1800–1807. https://doi.org/10.1109/CVPR.2017.195
27. He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition // Proc. of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. P. 770–778. https://doi.org/10.1109/CVPR.2016.90
28. Вяткин А.А., Тулупьев А.Л. Автоматизация проверки непротиворечивости идеалов конъюнктов с оценками вероятности истинности // Информационная безопасность регионов России (ИБРР-2021): материалы XII Санкт-Петербургской межрегиональной конференции. 2021. С. 330–332.
29. Вяткин А.А., Абрамов М.В., Харитонов Н.А., Тулупьев А.Л. Применение третичной структуры алгебраической байесовской сети в задаче апостериорного вывода // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. 2023. Т. 12. № 1. С. 61–88. https://doi.org/10.14529/cmse230104
30. Вяткин А.А., Харитонов Н.А., Тулупьев А.Л. Применение алгебраических байесовских сетей в задаче распознавания рукописных символов // Региональная информатика и информационная безопасность: сборник трудов Юбилейной XVIII Санкт-Петербургской международной конференции. 2022. С. 538–542.

14. Khan A., Sohail A., Zahoora U., Qureshi A.S. A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 2020, vol. 53, no. 8, pp. 5455–5516. https://doi.org/10.1007/s10462-020-09825-6
15. Oliseenko V., Abramov M. Identification of user profiles in online social networks: a combined approach with face recognition. *Journal of Physics: Conference Series*, 2021, vol. 1864, pp. 012119. https://doi.org/10.1088/1742-6596/1864/1/012119
16. Bushmelev F., Khlobystova A., Abramov M., Livshits L. Deep machine learning techniques in the problem of estimating the expression of psychological characteristics of a social media user. *Studies in Systems, Decision and Control*, 2023, vol. 457, pp. 315–324. https://doi.org/10.1007/978-3-031-22938-1_22
17. Shafiq M., Gu Z. Deep residual learning for image recognition: a survey. *Applied Sciences*, 2022, vol. 12, no. 18, pp. 8972. https://doi.org/10.3390/app12188972
18. Hossen M.I., Hei X. A low-cost attack against the hcaptcha system. *Proc. of the 2021 IEEE Security and Privacy Workshops (SPW)*, 2021, pp. 422–431. https://doi.org/10.1109/SPW53761.2021.00061
19. Kapoor A., Shah R., Bhuva R., Pandit T. *Understanding inception network architecture for image classification: Technical Report*, 2020. https://doi.org/10.13140/RG.2.2.16212.35204
20. Mittal S., Kaushik P., Hashmi S., Kumar K. Robust real time breaking of image CAPTCHAs using inception v3 model. *Proc. of the 2018 Eleventh International Conference on Contemporary Computing (IC3)*, 2018, pp. 1–5. https://doi.org/10.1109/IC3.2018.8530607
21. Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair Sh., Courville A., Bengio Y. Generative Adversarial Networks. *Communications of the ACM*, 2020, vol. 63, no. 11, pp. 139–144. https://doi.org/10.1145/3422622
22. Mirza M., Osindero S. Conditional generative Adversarial Nets. *arXiv*, 2014, arXiv:1411.1784. https://doi.org/10.48550/arXiv.1411.1784
23. Krizhevsky A., Sutskever I., Hinton G.E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 2017, vol. 60, no. 6, pp. 84–90. https://doi.org/10.1145/3065386
24. Huang G., Liu Z., Van Der Maaten L., Weinberger K.Q. Densely connected convolutional networks. *Proc. of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269. https://doi.org/10.1109/CVPR.2017.243
25. Ronneberger O., Fischer P., Brox T. U-Net: Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science*, 2015, vol. 9351, pp. 234–241. https://doi.org/10.1007/978-3-319-24574-4_28
26. Chollet F. Xception: Deep learning with depthwise separable convolutions. *Proc. of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1800–1807. https://doi.org/10.1109/CVPR.2017.195
27. He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition. *Proc. of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90
28. Vyatkin A., Tulupyev A. Automation of consistency checking of ideals of conjuncts with truth probability estimates. *Information Security of Russian Regions (ISRR-2021). Proc. of the XII St. Petersburg Interregional Conference*, 2021, pp. 330–332. (in Russian).
29. Vyatkin A., Abramov M., Kharitonov N., Tulupyev A. Application of tertiary structure of algebraic bayesian network in the problem of a posteriori inference. *Bulletin of the South Ural State University. Series "Computational Mathematics and Computer Science"*, 2023, vol. 12, no. 1, pp. 61–88. (in Russian). https://doi.org/10.14529/cmse230104
30. Vyatkin A., Kharitonov N., Tulupyev A. Application of algebraic bayesian networks in handwritten character recognition. *Regional Informatics and Information Security. Proc. of the Anniversary XVIII St. Petersburg International Conference*, 2022, pp. 538–542. (in Russian).

**Авторы**

**Задорожный Александр Сергеевич** — студент, Санкт-Петербургский государственный университет, Санкт-Петербург, 199034, Российская Федерация, https://orcid.org/0009-0004-8903-7244, alexander.zadorozhnyy@yandex.ru

**Корепанова Анастасия Андреевна** — младший научный сотрудник, Санкт-Петербургский Федеральный исследовательский центр Российской академии наук, Санкт-Петербург, 199178, Российская Федерация, SC 57218191916, https://orcid.org/0000-0003-2962-8670, aak@mail.ru

**Абрамов Максим Викторович** — кандидат технических наук, старший научный сотрудник, Санкт-Петербургский Федеральный исследовательский центр Российской академии наук, Санкт-Петербург, 199178, Российская Федерация, SC 56938320500, https://orcid.org/0000-0002-5476-3025, mva@dscs.pro

**Сабреков Артем Азатович** — младший научный сотрудник, Санкт-Петербургский Федеральный исследовательский центр Российской академии наук, Санкт-Петербург, 199178, Российская Федерация, SC 56938320500, https://orcid.org/0009-0000-8378-563X, aas@dscs.pro

**Authors**

**Alexander S. Zadorozhnyy** — Student, St. Petersburg State University (SPbSU), Saint Petersburg, 199034, Russian Federation, https://orcid.org/0009-0004-8903-7244, alexander.zadorozhnyy@yandex.ru

**Anastasia A. Korepanova** — Junior Researcher, Saint Petersburg Federal Research Center of the Russian Academy of Sciences, Saint Petersburg, 199178, Russian Federation, SC 57218191916, https://orcid.org/0000-0003-2962-8670, aak@mail.ru

**Maxim V. Abramov** — PhD, Senior Researcher, Saint Petersburg Federal Research Center of the Russian Academy of Sciences, Saint Petersburg, 199178, Russian Federation, SC 56938320500, https://orcid.org/0000-0002-5476-3025, mva@dscs.pro

**Artem A. Sabrekov** — Junior Researcher, Saint Petersburg Federal Research Center of the Russian Academy of Sciences, Saint Petersburg, 199178, Russian Federation, SC 56938320500, https://orcid.org/0009-0000-8378-563X, aas@dscs.pro

*Статья поступила в редакцию 12.10.2023*
*Одобрена после рецензирования 14.12.2023*
*Принята к печати 24.01.2024*

*Received 12.10.2023*
*Approved after reviewing 14.12.2023*
*Accepted 24.01.2024*