УДК 519.633.6

А.В. Лаптев

АВТОМАТИЗАЦИЯ РЕШЕНИЯ CAPTCHA В ТЕКСТОВОМ ФОРМАТЕ

Автоматизация тестирования web-приложений нередко осложняется необходимостью взаимодействия с защитными механизмами, такими как текстовые CAPTCHA. Для преодоления этого препятствия в данной работе предлагается метод автоматического распознавания CAPTCHA с использованием нейросетевых моделей. Разработанная архитектура ориентирована на устойчивое определение символов в искажённых и зашумлённых изображениях, характерных для CAPTCHA, что позволяет повысить надёжность и полноту автоматизированных тестов. Представлены результаты обучения модели на специализированном датасете и проведена оценка её точности в различных условиях. Полученные данные демонстрируют потенциал предлагаемого подхода в рамках задач обеспечения бесперебойной работы автоматизированных средств тестирования.

***Ключевые слова:****CAPTCHA, автоматизированное распознавание, нейронная сеть, TensorFlow, OCR, Tesseract, CRNN, Seq-to-Seq, Python.*

Введение

CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) представляет собой широко используемый механизм защиты web-ресурсов от автоматизированного доступа, спама и несанкционированного извлечения данных. На сегодняшний день текстовые CAPTCHA, несмотря на появление более продвинутых и «невидимых» форм, остаются распространёнными, в том числе в процессах регистрации, подтверждения действий и защиты от автоматизированных атак.

Автоматизация тестирования веб-приложений требует полноценной имитации пользовательского поведения, включая прохождение защитных механизмов. Одной из актуальных задач является разработка методов, способных автоматически распознавать символы, представленные в виде искажённого текста в CAPTCHA-изображениях. Решение данной задачи позволяет существенно повысить эффективность средств автоматизированного тестирования, снижая зависимость от ручных проверок.

Настоящая работа направлена на разработку и реализацию нейросетевой модели, способной эффективно распознавать текстовые CAPTCHA. Для достижения поставленной цели были сформулированы и решены следующие задачи:

1. Проведён анализ структуры текстовых CAPTCHA на основе открытых источников, включая допустимые символы и типичные виды искажений.
2. Выбрана архитектура нейронной сети, соответствующая требованиям к точности и устойчивости распознавания.
3. Подготовлен специализированный набор данных, отражающий возможные варианты искажений в CAPTCHA.
4. Осуществлено обучение модели на подготовленном датасете.
5. Проведена тестовая проверка качества распознавания и оценка эффективности модели.

**Современная реализация текстовых CAPTCHA**

Современные текстовые CAPTCHA обычно состоят из букв и цифр. Зачастую используются символы латинского алфавита (как прописные, так и строчные) и цифры от 0 до 9. Но обычно реализации исключают символы, которые могут быть легко перепутаны, например, буквы «O» и цифру «0», буквы «I» и «l» и тому подобное. Рекомендуемый набор символов в генераторах на некоторых CRM платформах выглядит следующим образом: ABCDEFGHJKLMNPQRSTWXYZ23 456789.

Длина последовательности символов в CAPTCHA обычно составляет от 4 до 8 символов, что обеспечивает баланс между удобством для пользователя и безопасностью, однако конкретная длина может варьироваться в зависимости от требований системы безопасности.

Для усложнения автоматического распознавания текстовые CAPTCHA подвергаются различным искажениям:

1. Геометрические искажения: символы могут быть искажены, повернуты или наклонены, что затрудняет их распознавание автоматическими системами.
2. Перекрытие символов: символы могут быть расположены близко друг к другу или даже перекрываться, что усложняет их сегментацию и последующее распознавание.
3. Добавление шума: на изображение могут быть добавлены различные шумы, такие как линии, точки или круги, чтобы затруднить распознавание символов.
4. Сложные фоны: использование фонов с различными цветами или узорами, что делает выделение символов более сложным.
5. Нелинейные искажения: применение нелинейных трансформаций к символам, что делает их форму менее предсказуемой для автоматических систем распознавания.

Эти методы направлены на повышение сложности автоматического распознавания CAPTCHA, сохраняя при этом относительную легкость распознавания для человека.

**Подготовка датасета с изображениями и выбор модели нейронной сети**

Качество используемого датасета оказывает существенное влияние на итоговую точность работы модели. Для эффективного обучения необходимо, чтобы набор данных соответствовал следующим требованиям:

1. Достаточное количество изображений для каждого символа, что обеспечивает статистическую устойчивость модели.
2. Разнообразие данных, включающее:
   1. Различные углы наклона символов.
   2. Вариативность написания символов и их искажения.
   3. Наличие побочных визуальных элементов, создающих препятствия для распознавания.
   4. Использование различных шрифтов.
3. Переменная длина последовательностей символов, что позволяет модели адаптироваться к разным формам CAPTCHA.

Включение указанных факторов способствует обучению модели на более широком спектре признаков, что, в свою очередь, повышает её способность к обобщению на ранее невидимых данных.

Поскольку в открытом доступе отсутствует достаточное количество данных для формирования сбалансированного датасета, было принято решение о генерации синтетических изображений с использованием специализированных библиотек. В качестве основного инструмента выбрана библиотека captcha на языке Python, обладающая необходимым функционалом для создания изображений CAPTCHA с заданными параметрами. Данная библиотека поддерживает генерацию изображений с пользовательскими шрифтами и различными эффектами искажений, что исключает необходимость привлечения дополнительных инструментов.

После создания изображений все они прошли этапы предобработки, направленные на улучшение качества данных и повышение эффективности обучения модели. Предобработка включала следующие этапы:

1. Преобразование изображений в градации серого для уменьшения количества каналов и снижения вычислительной нагрузки.
2. Бинаризация изображений с целью получения контрастного представления символов (белый текст на черном фоне)
3. Удаление шумов и фона с использованием морфологических операций, в частности, дилатации.

Примеры сгенерированных и предобработанных CAPTCHA приведены на рисунке ниже:

YKQ9out

**Рис. 1. Примеры CAPTCHA для распознавания: а – сгенерированное изображение; б – изображение после предобработки.**

Для распознавания текста с переменной длиной последовательности в задачах CAPTCHA наиболее часто применяются следующие архитектуры нейронных сетей:

1. Оптическое распознавание символов (OCR).
2. Рекуррентные сверточные нейронные сети (CRNN).
3. Архитектуры последовательного обучения (Seq-to-Seq).

С целью выбора наиболее эффективной модели были реализованы и протестированы все указанные подходы, после чего была выбрана архитектура, обеспечивающая наилучшую точность предсказаний.

Для обучения моделей был сформирован датасет из 100 000 изображений CAPTCHA, содержащих случайные последовательности символов длиной от 4 до 7. Такой объем данных позволяет добиться высокой обобщающей способности модели и снизить вероятность переобучения.

Оптическое распознавание символов (OCR Tesseract)

Изначально предполагалась реализация модели с использованием OCR, поскольку такие системы изначально разрабатывались для задач оптического распознавания текста. В качестве конкретной модели был выбран Tesseract.

Tesseract является одной из наиболее популярных систем OCR с открытым исходным кодом. Tesseract поддерживает более 100 языков, включая сложные письменности. В версии 4.0 в модель была интегрирована нейронная сеть на основе долговременной краткосрочной памяти (LSTM), что позволило существенно повысить точность распознавания, особенно при обработке сложных шрифтов и рукописного текста.

Для решения поставленной задачи предполагалось использовать предобученную модель Tesseract и дообучить её на специализированном датасете, содержащем изображения CAPTCHA с характерными искажениями. Однако в ходе экспериментов было установлено, что точность распознавания последовательностей символов целиком составляла 0%, а точность для отдельных символов оказалась крайне низкой. Это связано с тем, что архитектура Tesseract недостаточно устойчива к искажениям, характерным для CAPTCHA, таким как деформация символов, наложение шумов и изменение углов наклона.

Таким образом, было принято решение отказаться от использования Tesseract в пользу более адаптированных к данной задаче моделей, таких как сверточные рекуррентные нейронные сети (CRNN) или модели последовательного обучения (Seq-to-Seq), обладающие высокой устойчивостью к вариативности и искажениям, характерным для CAPTCHA.

Рекурсивные сверточные нейронные сети (CRNN)

Сверточно-рекуррентные нейронные сети (CRNN) представляют собой гибридную архитектуру, сочетающую в себе возможности сверточных нейронных сетей (CNN) и рекуррентных нейронных сетей (RNN). Данный подход используется в задачах, связанных с обработкой последовательных данных, таких как распознавание текста, речь и видео.

Основное преимущество CRNN заключается в способности CNN-части извлекать пространственные признаки из изображений, тогда как RNN-часть позволяет учитывать временные зависимости между последовательными фрагментами данных.

Разработанная модель CRNN для распознавания CAPTCHA включает в себя три ключевых блока:

1. Сверточный блок (CNN): предназначен для выделения признаков из изображений CAPTCHA. Включает в себя три последовательных сверточных слоя, а также методы нормализации и уменьшения размерности признакового пространства.
2. Рекуррентный блок (RNN): использует двунаправленные слои GRU, позволяющие модели учитывать зависимость между последовательными символами в CAPTCHA.
3. Выходной слой: полносвязный слой, который выполняет классификацию каждого символа в последовательности.

В данной архитектуре применяются слои Dropout для регуляризации, также используется l2-регуляризация, BatchNormalization для ускорения обучения и повышения устойчивости модели, а также функция softmax для предсказания классов символов.

После обучения данной модели результаты оказались превосходящими показатели OCR, однако все же не достигли удовлетворительного уровня. В частности, точность распознавания всей последовательности символов не превышала 10%, тогда как точность классификации отдельных символов составляла около 70%.

Архитектура последовательного обучения (Seq-to-Seq)

Модели последовательного преобразования (Seq-to-Seq) широко применяются для задач, связанных с обработкой последовательностей переменной длины. Они используются в таких областях, как машинный перевод, распознавание речи и анализ текстов. Данные модели основаны на архитектуре энкодера-декодера, где первый модуль кодирует входную последовательность в скрытое представление, а второй декодирует его в выходную последовательность.

Одним из ключевых элементов Seq-to-Seq является механизм внимания, который позволяет декодеру динамически фокусироваться на различных частях входной последовательности при генерации выходных символов. Этот подход особенно полезен для распознавания CAPTCHA, так как символы в изображениях могут иметь разную ориентацию и степень искажения.

Кодировщик, в данной модели принимает входное изображение CAPTCHA и преобразует его в компактное представление. Архитектура кодировщика включает:

1. Четыре сверточных блока, слои пакетной нормализации и слои подвыборки для понижения размерности входных данных
2. Глобальный усредненный слой для получения векторного представления изображения.
3. Полносвязный слой для финального представления скрытого состояния
4. Рекуррентный слой для кодирования последовательности, возвращающий последнее скрытое состояние кодировщика.

Декодировщик выполняет пошаговую генерацию выходной последовательности, используя скрытое состояние кодировщика. В архитектуру декодировщика входят:

1. Входной слой для последовательности токенов.
2. Слой вложения, который преобразует входные токены в векторные представления.
3. Рекуррентный слой, обрабатывающий последовательность с учетом скрытого состояния кодировщика.
4. Механизм внимания, который позволяет декодеру учитывать релевантные части входного изображения.
5. Полносвязный слой с функцией активации softmax для предсказания вероятностей символов.

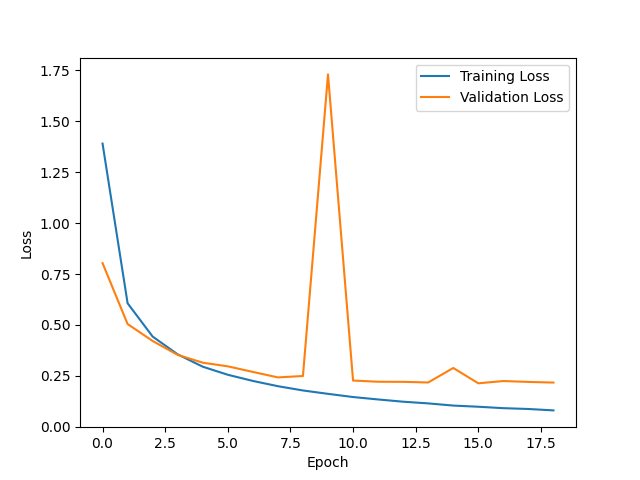
На начальных этапах экспериментов предложенная Seq-to-Seq-модель показала наилучшие результаты среди рассмотренных вариантов. В отличие от OCR- и CRNN-моделей, данная архитектура смогла достичь более высокой точности распознавания последовательностей символов, что обусловлено применением механизма внимания. Дальнейшая работа с моделью была сосредоточена на её оптимизации и улучшении параметров обучения.

Тестирование моедли нейронной сети

Как было установлено в предыдущих разделах, модель последовательного преобразования (Seq-to-Seq) продемонстрировала наилучшие результаты среди рассмотренных архитектур. Следующим этапом работы являлась оптимизация параметров модели, включая веса и коэффициенты регуляризации, с целью ускорения сходимости, минимизации риска переобучения и повышения точности распознавания целевых последовательностей.

Для проведения экспериментов исходный набор данных, содержащий 100 000 изображений, был случайным образом перемешан и разделён на три подмножества: обучающее, тестовое и валидационное в соотношении 80:10:10. Обучающая выборка использовалась непосредственно для обучения модели, валидационная — для контроля качества процесса обучения на каждой эпохе, а тестовая — для окончательной оценки модели на данных, с которыми она ранее не сталкивалась. В качестве основных метрик качества модели использовались функция потерь (loss) и точность (accuracy), рассчитываемая для каждого символа последовательности.

В процессе многократного обучения были экспериментально определены оптимальное количество эпох и значения гиперпараметров, обеспечивающие эффективное снижение функции потерь до приемлемых значений. График сходимости функции потерь представлен ниже.



**Рис. 2. График сходимости функции потерь.**

Для предотвращения переобучения использовался механизм ранней остановки, согласно которому обучение прекращалось при отсутствии уменьшения значения функции потерь на валидационной выборке в течение трёх последовательных эпох. В данном эксперименте обучение завершилось на 18-й эпохе. На графике видно, что функция потерь стабилизировалась после 10 эпохе, поэтому 10 эпоха является балансом между точностью распознавания последовательностей и скоростью обучения модели.

Анализ графика сходимости функции потерь показывает наличие резкого увеличения её значения на 9-й эпохе, что может быть обусловлено следующими факторами:

1. Перемешивание данных перед каждой эпохой могло привести к образованию несбалансированной выборки, содержащей значительное число сложных примеров.

Динамическое изменение скорости обучения, осуществляемое с помощью механизма регулирования скорости обучения (learning rate scheduler), могло повлиять на изменение функции потерь.

Окончательная точность распознавания отдельных символов составила 0.9263.

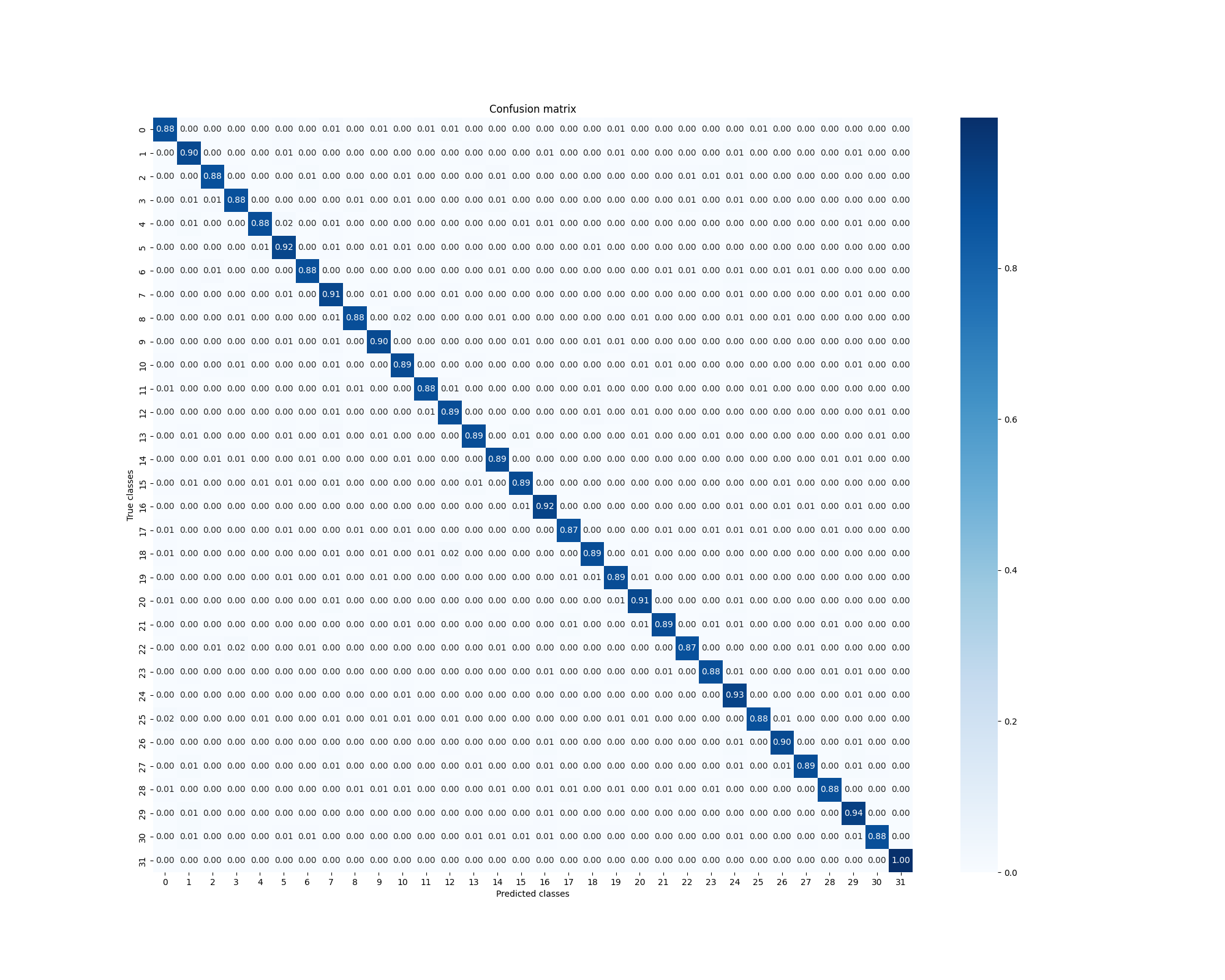
После подбора оптимальных значений гиперпараметров модель была сохранена и протестирована на валидационной выборке. Точность распознавания последовательностей различной длины представлена в таблице.

Таблица

Точность предсказаний для последовательностей различной длины

|  |  |
| --- | --- |
| Длина последовательности | Точность распознавания |
| 4 символов | 0.9305 |
| 5 символов | 0.7450 |
| 6 символов | 0.4575 |
| 7 символов | 0.1915 |

Также была построена матрица ошибок, позволяющая проанализировать частоту и характер ошибок модели при классификации различных классов.



**Рис. 3. График матрицы ошибок распознавания моделью отдельных символов.**

Анализ полученных результатов показывает, что точность распознавания последовательностей значительной длины остаётся относительно низкой. Это можно объяснить высокой зависимостью модели Seq-to-Seq от объёма обучающих данных: для эффективного обобщения признаков, извлекаемых из изображений, требуется значительное количество примеров. Следовательно, увеличение размера обучающего набора данных потенциально может способствовать повышению точности модели, однако это также накладывает дополнительные требования к вычислительным ресурсам, необходимым для её обучения.

Заключение

Для решения задачи по автоматизированному решению текстовых CAPTCHA была разработана и реализована нейросетевая модель, основанная на архитектуре Seq-to-Seq. Проведён анализ структуры текстовых CAPTCHA, включая допустимые символы и характерные виды искажений, на основе чего был сформирован специализированный датасет объёмом 100 000 изображений. Модель обучалась на данных с переменной длиной последовательностей символов, что позволило повысить универсальность решения.

Результаты тестирования продемонстрировали способность модели эффективно распознавать текстовые CAPTCHA различной сложности, подтверждая её применимость в задачах автоматизации тестирования web-приложений. Полученные данные могут быть использованы как для улучшения средств тестирования, так и для анализа устойчивости CAPTCHA-систем, применяемых на практике.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. CAPTCHA in 1C-Bitrix [Electronic resource]. – URL: https://dev.1c-bitrix.ru/user\_help/settings/settings/captcha.php
2. What is a CAPTCHA and how does it work [Electronic resource]. – URL: [https://ru-brightdata.com/blog/web-data-ru/what-is-a-captcha](https://ru-brightdata.com/blog/web-data-ru/what-is-a-captcha" \t "_new)
3. «Break me completely!» How some algorithms generate CAPTCHAs and others break them [Electronic resource]. – URL: https://proglib.io/p/lomay-menya-polnostyu-kak-odni-algoritmy-generiruyut-kapchu-a-drugie-ee-vzlamyvayut-2020-03-05
4. Rogov A.V. CAPTCHA in the era of digital threats: evolution, vulnerabilities, and solutions // Scientific Statements. 2024. No. 21(68). P. 18–22.
5. How to configure CAPTCHA [Electronic resource]. – URL: https://support.simai.ru/learn/courses/course/3/lesson/309/
6. Tesseract OCR: How it works and when to use it [Electronic resource]. – URL: [https://www.klippa.com/en/blog/information/tesseract-ocr](https://www.klippa.com/en/blog/information/tesseract-ocr" \t "_new)
7. Tesseract project GitHub repository [Electronic resource]. – URL: [https://github.com/tesseract-ocr/tesseract](https://github.com/tesseract-ocr/tesseract" \t "_new)
8. Training Tesseract OCR with custom data [Electronic resource]. – URL: https://saiashish90.medium.com/training-tesseract-ocr-with-custom-data-d3f4881575c0
9. Once again about handwritten text recognition, this time using CRNN [Electronic resource]. – URL: [https://habr.com/ru/articles/720614/](https://habr.com/ru/articles/720614/" \t "_new)
10. Bezvikonny N.V., Guskov A.A. Software emulation of an optomagnetic neural network for handwritten text analysis // World of Science. 2024. P. 1–63.
11. Sequence-to-sequence models [Electronic resource]. – URL: https://www.ultralytics.com/ru/glossary/sequence-to-sequence-models
12. Morkovnikov N.M., Kipyatkova I.S. Study of encoder-decoder model construction methods for Russian speech recognition // Information and Control Systems. 2019. No. 4. P. 1–9.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| Лаптев Александр Владимирович  Студент 2-го курса магистратуры, Институт цифровых технологий, электроники и физики, ФГБОУ ВО Алтайский государственный университет, Россия, 656049, г. Барнаул, пр-т. Ленина, 61,  Эл. почта:LaptevAlexander2001@gmail.com |

*A.V. LAPTEV*

**AUTOMATION OF TEXT-BASED CAPTCHA SOLVING**

**Automation of web application testing is often complicated by the need to interact with protective mechanisms such as text-based CAPTCHAs. To overcome this challenge, this paper proposes a method for automatic CAPTCHA recognition using neural network models. The developed architecture is designed for robust character recognition in distorted and noisy images typical of CAPTCHAs, which enhances the reliability and completeness of automated tests. The paper presents the results of model training on a specialized dataset and evaluates its accuracy under various conditions. The obtained data demonstrate the potential of the proposed approach in ensuring the uninterrupted operation of automated testing tools.**

***Keywords:*** *CAPTCHA, automated recognition, neural network, TensorFlow, OCR, Tesseract, CRNN, Seq-to-Seq, Python.*

REFERENCES

1. CAPTCHA in 1C-Bitrix [Electronic resource]. – URL: https://dev.1c-bitrix.ru/user\_help/settings/settings/captcha.php
2. What is a CAPTCHA and how does it work [Electronic resource]. – URL: [https://ru-brightdata.com/blog/web-data-ru/what-is-a-captcha](https://ru-brightdata.com/blog/web-data-ru/what-is-a-captcha" \t "_new)
3. «Break me completely!» How some algorithms generate CAPTCHAs and others break them [Electronic resource]. – URL: https://proglib.io/p/lomay-menya-polnostyu-kak-odni-algoritmy-generiruyut-kapchu-a-drugie-ee-vzlamyvayut-2020-03-05
4. Rogov A.V. CAPTCHA in the era of digital threats: evolution, vulnerabilities, and solutions // Scientific Statements. 2024. No. 21(68). P. 18–22.
5. How to configure CAPTCHA [Electronic resource]. – URL: https://support.simai.ru/learn/courses/course/3/lesson/309/
6. Tesseract OCR: How it works and when to use it [Electronic resource]. – URL: [https://www.klippa.com/en/blog/information/tesseract-ocr](https://www.klippa.com/en/blog/information/tesseract-ocr" \t "_new)
7. Tesseract project GitHub repository [Electronic resource]. – URL: [https://github.com/tesseract-ocr/tesseract](https://github.com/tesseract-ocr/tesseract" \t "_new)
8. Training Tesseract OCR with custom data [Electronic resource]. – URL: https://saiashish90.medium.com/training-tesseract-ocr-with-custom-data-d3f4881575c0
9. Once again about handwritten text recognition, this time using CRNN [Electronic resource]. – URL: [https://habr.com/ru/articles/720614/](https://habr.com/ru/articles/720614/" \t "_new)
10. Bezvikonny N.V., Guskov A.A. Software emulation of an optomagnetic neural network for handwritten text analysis // World of Science. 2024. P. 1–63.
11. Sequence-to-sequence models [Electronic resource]. – URL: https://www.ultralytics.com/ru/glossary/sequence-to-sequence-models
12. Morkovnikov N.M., Kipyatkova I.S. Study of encoder-decoder model construction methods for Russian speech recognition // Information and Control Systems. 2019. No. 4. P. 1–9.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| **Alexander V. Laptev** 2nd-year Master's student, Institute of Digital Technologies, Electronics and Physics, Altai State University, Barnaul, Russia, 656049, 61 Lenin Avenue Email: LaptevAlexander2001@gmail.com |