Распознавание русскоязычного рукописного текста на изображениях с использованием сверточной рекуррентной нейронной сети

А.С. Жданов, Т.С.Евдокимова

Казанский национальный исследовательский технический университет – КАИ, им. Туполева, Казань

Аннотация: В статье представлены результаты разработки алгоритма и десктоп-приложения для распознавания русского рукописного текста на изображениях с использованием технологий компьютерного зрения и глубокого обучения. Изучены классические и современные методы распознавания, разработан и реализован алгоритм, обеспечивающий точность распознавания 71%. Приложение позволяет пользователю загружать изображения, получать оцифрованный текст и сохранять результаты в личном кабинете. Программная реализация включает блок обучения модели с оценкой метрик точности и полноты. Приложение соответствует всем поставленным требованиям, обеспечивая удобство использования и функциональность.
Ключевые слова: глубокое обучение, рукописный текст, изображение, данные, обучение модели, компьютерное зрение, извлечение признаков, сверточная нейронная сеть, рекуррентная нейронная сеть.

Введение

В современном мире цифровизации и автоматизации особое внимание уделяется разработке технологий, способных повысить эффективность обработки и анализа данных. Одной из ключевых задач в этой области является распознавание рукописного текста, которое находит применение в различных сферах, включая архивирование документов, образование и офисную работу [1]. Распознавание рукописного текста представляет собой сложную задачу из-за высокой вариативности почерка, наличия искажений на изображениях и нестандартных символов.

Традиционные методы, такие как оптическое распознавание символов, эффективны для работы с печатным текстом, но не справляются с рукописным из-за его неоднородности [2]. Современные подходы, основанные на глубоком обучении, включая сверточные и рекуррентные нейронные сети с использованием метода выравнивания временных последовательностей (Connectionist temporal classification — CTC),

демонстрируют высокую точность даже при обработке сложных и искажённых данных. Эти методы позволяют эффективно анализировать последовательности символов и адаптироваться к различным стилям письма [3].

В данной работе рассматривается разработка алгоритма и десктоп-приложения для распознавания русскоязычного рукописного текста на изображениях. Основной целью является создание системы, способной автоматически преобразовывать изображения с рукописным текстом в читаемый цифровой формат. Входные данные включают изображения, полученные из различных источников, таких как сканированные документы, фотографии или снимки с мобильных устройств. Выходные данные представляют собой распознанный текст, который может быть сохранён в форматах, пригодных для дальнейшего использования.

Актуальность работы обусловлена необходимостью автоматизации процессов обработки рукописных документов, что позволяет сократить временные затраты и повысить точность распознавания. Разработанное решение может быть применено в различных областях, включая образование, архивирование и офисную работу, где требуется быстрая и точная обработка рукописных данных [4].

Метод решения задачи

Сверточная рекуррентная нейронная сеть (Convolutional Recurrent Neural Networks — CRNN) — это гибридная нейронная сеть, сочетающая в себе достоинства сверточных нейронных сетей для извлечения локальных признаков из изображений и рекуррентных нейронных сетей, представленных двумя слоями двунаправленной долгой краткосрочной памяти (Long Short-Term Memory — LSTM), которые отвечают за обработку последовательностей [5]. CRNN активно используется в задачах распознавания рукописного и машинописного текста, текстов с искажениями, а также для работы с последовательностями символов и

других объектов, где важно учитывать пространственно-временные зависимости. Для обучения таких сетей в задачах, где отсутствует явное соответствие между входными и выходными данными (например, распознавание речи или текста), применяется метод CTC.

Сверточная часть сети устроена таким образом, что исходное изображение разделяется на k (гиперпараметр) вертикальных сегментов вдоль горизонтальной оси. Каждый из этих сегментов после прохождения сверточных слоёв преобразуется в вектор признаков размерности (1, n). Благодаря уменьшению одного из измерений до единицы, размерность карты признаков снижается без потери информации ((k, 1, n) -> (k, n)). В итоге на выходе формируется двумерная матрица признаков размером (k, n), которая передаётся на вход рекуррентной нейронной сети. Здесь n также является гиперпараметром, определяющим размерность вектора признаков для каждого окна, что фактически соответствует количеству карт признаков, полученных на последнем этапе свертки [6].

В блоке рекуррентной нейронной сети используются два двунаправленных слоя LSTM [7], которые преобразуют входную последовательность. Классификация элементов последовательности по нескольким классам осуществляется с помощью полносвязного слоя, где число классов равно длине алфавита плюс один дополнительный класс для служебного символа.

В качестве примера рассмотрим алфавит, состоящий из символов «0123456789» и служебного символа. Пример классификации может быть представлен следующим образом:

[10, 5, 5, 10, 8, 10, 3, 3, 10, 0, 10, 1, 1, 10, 4, 10, 9, 9, 10, 2, 10, 7, 7, 10, 6].

Применяя функцию argmax для декодирования матрицы softmax-векторов, получаем последовательность индексов, соответствующих элементам алфавита. Для удобства индексы заменяются на символы:

«-55-8-33-0-11-4-99-2-77-6».

Из-за особенностей разделения исходного изображения на окна, некоторые символы могут дублироваться, так как попадают в несколько окон. С помощью служебного символа (10) можно устранить лишние дубли, когда символ изображения попадает в несколько окон, а также сохранить истинные дубли (например, пятерки, тройки, единицы, девятки и семерки в данном примере).

Метод CTC применяется для выравнивания последовательности. В результате его работы раздвоившиеся символы объединяются (с сохранением истинных дублей), а служебные символы удаляются. Выходная последовательность принимает вид:

[5, 8, 3, 0, 1, 4, 9, 2, 7, 6, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1].

После выравнивания последовательности вектор дополняется символами -1 до размера, соответствующего выходу сети. Далее все элементы со значением -1 удаляются, а оставшиеся индексы заменяются на символы алфавита. Итоговый результат: «5830149276». Таким образом, исходная последовательность успешно обработана, а дублирующиеся символы корректно объединены с сохранением «настоящих» дублей. Переход между методами глубокого обучения иллюстрируется на рис. 1.

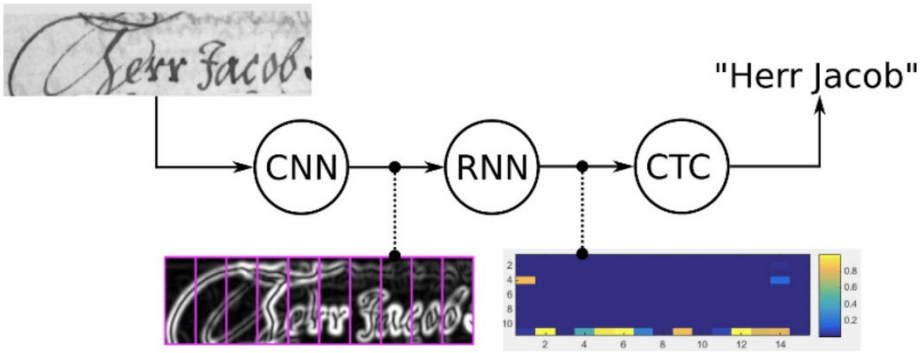

Рис. 1. – Переход из одного метода глубокого обучения в другой

Рассмотрим CTC более подробно. Алгоритм позволяет обойти проблему, когда мы не знаем, как входные данные соотносятся с выходными. Предположим, у нас есть входная последовательность $X = [x_1, x_2, \dots, x_T]$ и выходная $Y = [y_1, y_2, \dots, y_U]$. Необходимо найти четкое отображение X на Y, но также существуют проблемы: X и Y могут различаться по длине, нет точного соответствия элементов. С помощью алгоритма CTC можно решить данные проблемы. Для X мы выводим распределение выходных данных по всем возможным Y по вероятности. Те необходимо определить Y* по формуле (1):

$$Y^* = \underset{Y}{argmax}\, p(Y \mid X). \tag{1}$$

где p(Y|X) – условная вероятность того, что из X получим Y, Y*-максимизация по Y.

Для одной пары (X,Y) CTC может определить условную вероятность, при суммировании с набором допустимых выравниваний и вычислений вероятности одного выравнивания шаг за шагом по формуле (2).

$$p(Y \mid X) = \sum_{A \in A_{X,Y}} \prod_{t=1}^{T} p_t(a_t \mid X) \tag{2}$$

Глубокое обучение, особенно в контексте CRNN и CTC имеет свои достоинства и недостатки:

Достоинством является то, что CRNN способны автоматически извлекать важные признаки из данных, что снижает необходимость в ручной настройке и выборе признаков, высокая точность и производительность в задачах распознавания. CTC позволяет обучаться на последовательностях переменной длины, упрощает обучение моделей, работающих с временными рядами, высокая эффективность в распознавании речи и текста. Но также недостатками CRNN и CTC будут: необходимость большого объема

размеченных данных, высокие требования к вычислительным ресурсам, сложность настройки гиперпараметров и чувствительность к качеству данных [8].

Алгоритм решения задачи

Подготовка данных

Сбор изображений с русским рукописным текстом, которые будут использоваться для обучения, валидации и тестирования модели.

Данные должны охватывать разнообразные почерки, стили письма, размеры букв и условия съёмки (освещение, угол наклона). Под такие требование отлично подходит датасет «Cyrillic Handwriting Dataset» на Kaggle. Данный датасет состоит из 73830 сегментов рукописных текстов (кропов) на русском языке и разбит на тренировочные и тестовые наборы с разделением 95%, 5% соответственно.

Предварительная обработка изображений — это важный этап в любом проекте по обработке изображений с использованием машинного обучения и глубоких нейронных сетей. Основная цель предварительной обработки заключается в подготовке исходных данных для того, чтобы модель могла обучаться более эффективно. Этот этап включает в себя различные методы и техники, которые помогают улучшить качество изображений, ускорить обучение и улучшить обобщающую способность модели.

Изменение размера изображений. Изображения, полученные из разных источников, могут иметь различные размеры. Для того чтобы они подходили для обучения нейронной сети, необходимо привести их к единому размеру. Обычно размер изображения изменяется так, чтобы оно соответствовало размеру, необходимому для входа в модель (например, 224x224 пикселей для сети, обученной на ImageNet)

Удаление шума и фильтрация. В некоторых случаях на изображениях может присутствовать шум (например, из-за плохого качества камеры или

помех), который мешает модели обучаться. Для удаления шума применяются различные фильтры, такие как медианный фильтр, гауссовый фильтр и другие.

Обработка изображений для использования в нейронных сетях. Преобразованные изображения подготавливаются для подачи на вход нейронной сети: преобразование в тензоры, которые являются основной структурой данных для нейронных сетей.

Нормализация изображений — это процесс приведения значений пикселей изображения в стандартный диапазон, который легче воспринимается нейронными сетями. Без нормализации пиксели изображений могут иметь значения в диапазоне от 0 до 255 (как это обычно бывает в изображениях в формате 8 бит на канал), что может привести к проблемам с обучением, таким как долгие вычисления и нестабильность градиентов.

Существует два основных подхода к нормализации изображений: нормализация в диапазоне [0, 1] и нормализация по среднему значению и стандартному отклонению (Z-оценка). Преобразование значений пикселей из диапазона [0, 255] в диапазон [0, 1] помогает улучшить сходимость модели и ускоряет обучение. Обычно это выполняется простым делением каждого пикселя на 255. Для улучшения производительности модели в случае, если данные имеют разнообразие в пиксельных значениях, полезно выполнить нормализацию по среднему значению и стандартному отклонению. Этот метод обеспечивает, чтобы каждое изображение имело среднее значение 0 и стандартное отклонение 1 для каждого канала (например, RGB). Это может улучшить сходимость и ускорить обучение модели.

Извлечение признаков

На этом этапе используется сверточная нейронная сеть для извлечения признаков из предварительно обработанных изображений. Она помогает

выявить важные визуальные паттерны в изображениях, такие как линии, кривые и текстуры. Слои свертки применяются для создания карт признаков, а слои подвыборки (пулинг) уменьшают размерность и помогают извлечь наиболее важные характеристики.

Рекуррентная нейронная сеть

Полученные из сверточной нейронной сети признаки передаются в рекуррентную для последовательной обработки. Рекуррентная нейронная сеть, особенно LSTM или долгократкосрочная память с управляемыми воротами, позволяет учитывать временные зависимости в данных. Рекуррентная нейронная сеть принимает последовательности признаков и обучается моделировать зависимости между ними, что особенно важно для распознавания текста, так как буквы могут быть расположены в разных позициях и иметь разные размеры.

Декодирование и использование CTC

Используется для обучения модели с разной длиной входных и выходных последовательностей. CTC позволяет модели предсказывать последовательности переменной длины, что особенно полезно в задачах распознавания текста. Модель рекуррентной нейронной сети, обученная с использованием CTC, выдает вероятности для каждого символа на каждой временной метке. CTC позволяет декодировать эти вероятности в конечную строку текста, устраняя проблемы с выравниванием входов и выходов.

Постобработка

- Фильтрация: удаление ненужных символов (например, пробелов или пустых символов) из предсказанной строки.
- Коррекция ошибок: использование словарей или языковых моделей для исправления возможных ошибок в распознанном тексте.

Оценка производительности

- Метрики: оценка качества распознавания с использованием

различных метрик, таких как точность, полнота, F1-мера и показатель символьной ошибки).

- Тестирование на новых данных: проверка производительности модели на тестовом наборе данных, который не использовался в процессе обучения.

Оптимизация и дообучение

Оптимизация и дообучение включают тонкую настройку гиперпараметров модели, таких как скорость обучения, количество слоев и нейронов, а также дополнительное обучение, которое может потребоваться для улучшения производительности модели на новых данных.

Разработка и интеграция приложения

Создание пользовательского интерфейса включает разработку удобного интерфейса для пользователей, который позволяет загружать изображения и получать результаты распознавания, а интеграция модели предполагает внедрение обученной модели в конечное приложение, будь то десктопное или мобильное [9].

Результаты

Алгоритм решения задачи был реализован программно на языке Python в среде разработки PyCharm 2023. Для работы с компьютерным зрением использовались библиотеки PyTorch и TensorFlow [10]. Разработка предназначена для выполнения на персональном компьютере с операционной системой Microsoft Windows 11.

Было проведено 100 эпох при обучении модели. Точность достигла 0.7121, что превышает заявленные в требовании 67% (0.67). График на рис. 2 показывает постепенное приближение к этому значению с увеличением числа эпох. Значение потерь составило около 0.1004, что свидетельствует о хорошем обучении модели. Потери снижаются на протяжении всех эпох. Точность классификации положительных классов достиг 0.795. Полнота

выросла до 0.78, модель охватывает все положительные примеры. Итоговое значение F1-мера составило 0.7874, что говорит о сбалансированности между точностью и полнотой. Уровень уверенности составил 0.85. Фрагмент прогноза модели для слов представлен на рис. 3, результат предсказания правильных символов и правильных слов на тестовом наборе на рис. 4, точность и потери на валидационном наборе на рис. 5.

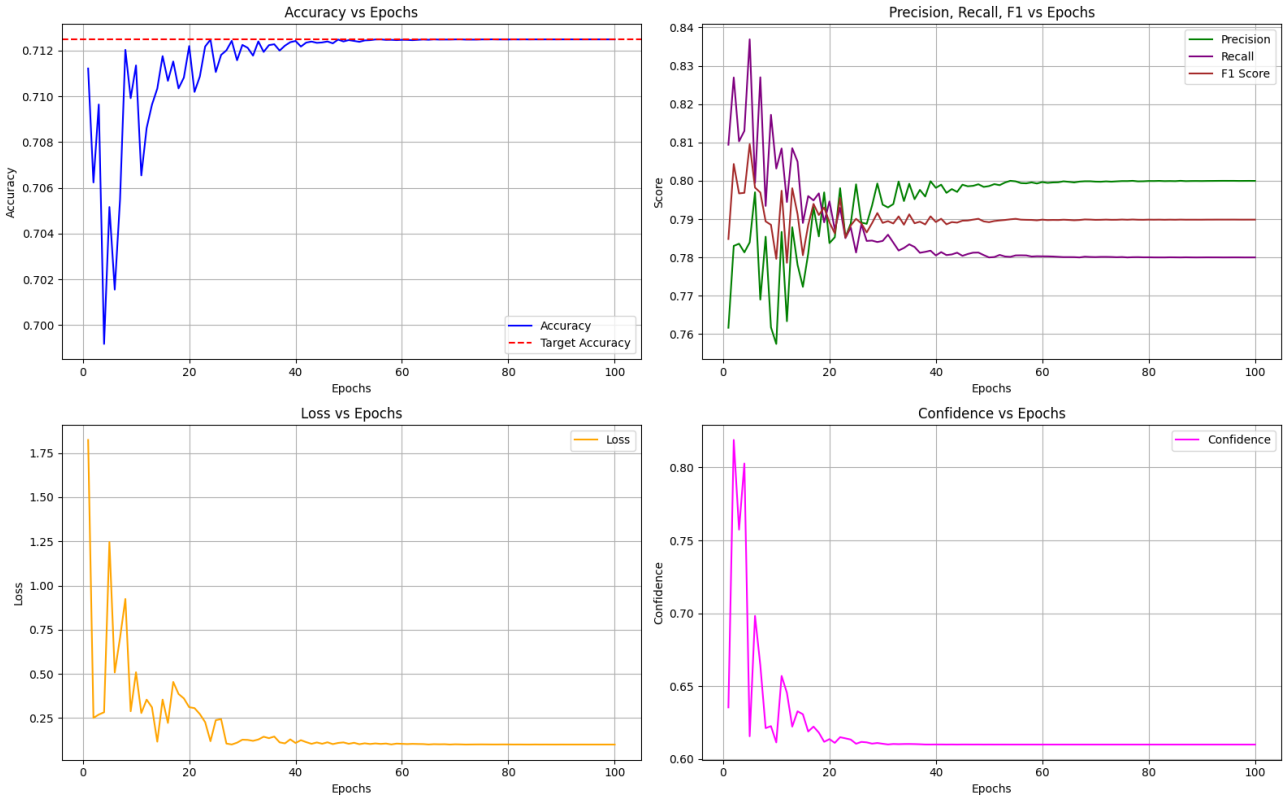

Рис. 2. – Результат распознавания текста. Основные метрики для определения качества модели

Ground Truth	Prediction	Confidence Score & T/F
заявление	зоявление	0.8026 False
проксимация	проксимация	0.6372 True

Рис. 3. – Фрагмент прогноза модели для слов

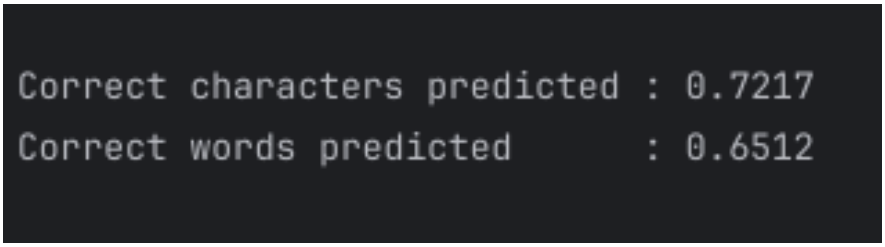

Рис. 4. – Результат предсказания правильных символов и правильных слов на тестовом наборе

Рис. 5. – Точность и потери на валидационном наборе

Заключение

В ходе работы был разработан алгоритм и десктоп-приложение для распознавания русского рукописного текста на изображениях. Созданная система автоматически преобразует изображения с рукописным текстом в читаемый цифровой формат. В процессе разработки были изучены как классические методы распознавания, так и методы, основанные на глубоком обучении. Подробно описаны этапы создания алгоритма. Особое внимание уделено результатам обучения модели, включая оценку метрик точности и полноты.

Обученная модель продемонстрировала точность распознавания 71%, что является достойным результатом для задачи обработки рукописного текста. Приложение полностью соответствует поставленным требованиям: пользователь может загружать изображения, получать оцифрованный текст и сохранять результаты в истории личного кабинета. Данное решение обладает потенциалом для дальнейшего улучшения и может быть применено в различных сферах, таких, как обработка документов, образование и персональное использование.

Литература

1. George Stockman, Linda G. Shapiro. Computer vision. Pearson. 2001. 608 p.

2. Илларионов А. А., Чернов А. А. Обзор методов распознавания текста //Аллея науки. 2019. Т. 2. №. 5. С. 1094-1098.

3. Зарипова Р.С., Кривоногова А.Е. Распознавание текстов с использованием нейронных сетей // NovaUm. Ru. 2018. №. 11. С. 38-40.

4. Игнатьева О.В. и др. Исследование работы методов машинного зрения в условиях изменения освещенности для встраиваемых систем // Инженерный вестник Дона. 2024. №. 5. URL: ivdon.ru/ru/magazine/archive/n5y2024/9194

5. Маркин Е.И., Зупарова В.В., Сальников И.И. Распознавание рукописного текста с использованием нейронных сетей // Научное обозрение. Педагогические науки. 2019. №. 3-2. С. 44-47.

6. Кучуганов А.В., Г.В. Лапинская. Подходы к обработке изображений с использованием сверточных сетей. // Материалы конференции УдГУ. Ижевск: УдГУ, 2020. URL: mns.udsu.ru/conf/report/Kuchuganov2.pdf

7. Frinken, V., Fischer, A., Baumgartner, M., & Bunke, H. (2014). Keyword spotting for self-training of BLSTM NN based handwriting recognition systems. Pattern Recognition. 2014. vol. 47. i. 3. pp.1073-1082.

8. Li Y. et al. HTR-VT: Handwritten text recognition with vision transformer //Pattern Recognition. 2025. vol. 158. i. 11. pp. 1-9.

9. XinSheng Z., Yu W. Industrial character recognition based on improved CRNN in complex environments //Computers in Industry. 2022. vol. 142. pp. 1-12.

10. Вильданов А.Р. Разработка интеллектуального программного модуля распознавания изображений для звуковых очков // Инженерный вестник Дона. 2022. №. 12. URL: ivdon.ru/ru/magazine/archive/n12y2022/8066.

References

1. George Stockman, Linda G. Shapiro. Computer vision. Pearson. 2001. 608 p.
2. Illarionov A.A., Chernov A.A. Alleya nauki. 2019. Vol. 2. №. 5. pp. 1094-1098.
3. Zaripova R.S., Krivonogova A.E. NovaUm.Ru. 2018. №. 11. pp. 38-40.
4. Ignat'eva O.V. et al. Inzhenernyj vestnik Dona. 2024. №. 5. URL: ivdon.ru/ru/magazine/archive/n5y2024/9194
5. Markin E.I., Zuparova V.V., Salnikov I.I. Nauchnoe obozrenie. Pedagogicheskie nauki. 2019. №. 3-2. pp. 44-47.
6. Kuchuganov A.V., Lapinskaya G.V. Materialy konferentsii UdGU. Izhevsk: UdGU, 2020. URL: mns.udsu.ru/conf/report/Kuchuganov2.pdf
7. Frinken, V., Fischer, A., Baumgartner, M., & Bunke, H. (2014). Pattern Recognition. 2014. vol. 47. i. 3. pp.1073-1082.
8. Li Y. et al. Pattern Recognition. 2025. vol. 158. i. 11. pp. 1-9.
9. XinSheng Z., Yu W. Computers in Industry. 2022. vol. 142. pp. 1-12.
10. Vildanov A.R. Inzhenernyj vestnik Dona. 2022. №. 12. URL: ivdon.ru/ru/magazine/archive/n12y2022/8066.

Дата поступления: 5.02.2025

Дата публикации: 26.03.2025