Министерство образования и науки Российской Федерации
Московский физико-технический институт (государственный университет)

Физтех-школа прикладной математики и информатики
Кафедра Интеллектуальной Обработки Документов

Выпускная квалификационная работа бакалавра

Исследование стратегий маскирования для semi-supervised предобучения моделей в задаче распознавания текста

Автор:
Студент 031 группы
Бухтуев Григорий Андреевич

Научный руководитель:
Рящиков Александр Павлович

Москва 2024

Аннотация

Исследование стратегий маскирования для semi-supervised предобучения моделей в задаче распознавания текста
Бухтуев Григорий Андреевич

Распознавание текста на изображениях (OCR) играет важную роль в различных приложениях, от цифровизации документов до систем помощи водителю. Semi-supervised обучение, использующее как размеченные, так и неразмеченные данные, является многообещающим подходом для повышения эффективности моделей OCR. В данной работе исследуется потенциал различных стратегий маскирования в semi-supervised обучении для повышения точности распознавания японского текста на изображениях.

В работе рассматриваются три основные стратегии маскирования: маскирование патчей входного изображения, маскирование входных представлений декодера и их комбинация. Проведен ряд экспериментов с различными архитектурами моделей. Для оценки качества разработанных моделей использовалась метрики Character Accuracy и Fragment Accuracy на тестовом наборе данных с японским текстом.

Результаты экспериментов показали, что применение стратегий маскирования позволяет повысить точность распознавания текста по сравнению с базовой моделью, обученной без маскирования. Комбинация маскирования патчей изображения и входных представлений декодера продемонстрировала наилучшие результаты.

Проведенное исследование вносит вклад в развитие области semi-supervised обучения для OCR, демонстрируя эффективность различных стратегий маскирования. Полученные результаты могут быть использованы для разработки более точных и эффективных систем распознавания японского текста, а предложенные подходы могут быть адаптированы и для других языков.

Содержание

1 Введение

1.1 Задача распознавания текста

Распознавание текста (Optical Character Recognition, OCR) – это область исследований в области искусственного интеллекта, целью которой является автоматизация процесса преобразования изображений, содержащих текст, в машиночитаемый текстовый формат. Данная задача имеет широкий спектр применений, включая автоматизацию документооборота, поиск по изображениям, системы помощи водителям, и многие другие.

- Типы задач распознавания текста

  Существует несколько основных категорий распознавания текста, характеризующихся типом входного изображения и сложностью обработки:

  - **OCR (Optical Character Recognition):** Классический OCR предназначен для распознавания печатного или машинописного текста в относительно простых условиях, например, в отсканированных документах с хорошим качеством изображения и четким отделением текста от фона.
  - **STR (Scene Text Recognition):** STR специализируется на распознавании текста в естественных сценах, где текст может быть изображен на сложных фонах, с различными шрифтами, цветами, перспективой и освещением.
  - **HWR (Handwritten Text Recognition):** HWR предназначен для распознавания рукописного текста, что является особенно сложной задачей из-за вариативности почерков, стилей письма и возможных дефектов изображения.

- Основные этапы распознавания текста

  Несмотря на разнообразие подходов, большинство систем распознавания текста реализуют следующие этапы:

  1. **Предобработка изображения:** Этот этап включает в себя ряд операций, направленных на улучшение качества изображения и выделение текстовых областей: бинаризацию, шумоподавление, сегментацию текста.
  2. **Извлечение признаков:** На этом этапе из изображений символов или слов извлекаются дискриминативные признаки, характеризующие их визуальные свойства. Для этого могут использоваться различные методы обработки изображений, в том числе и глубокое обучение.
  3. **Классификация:** Извлеченные признаки подаются на вход классификатора (например, нейронной сети), который определяет, какой символ или слово представлены на изображении.

4. **Постобработка:** На этом этапе могут применяться дополнительные методы для улучшения точности распознавания, такие как языковые модели, коррекция ошибок и др.

- Сложности и вызовы

  Разработка эффективных и универсальных систем распознавания текста сталкивается с рядом сложностей:

  – **Разнообразие условий съемки:** Освещение, ракурс, качество изображения и другие факторы могут значительно влиять на качество изображения и затруднять распознавание текста.
  – **Разнообразие шрифтов и языков:** Существует огромное количество шрифтов, стилей письма и языков, что затрудняет создание универсальных систем, способных распознавать любой текст, например распознавание текста на языках с иероглифической письменностью, таких как японский, представляет особую сложность.
  – **Сложные фоны:** Распознавание текста на сложных фонах (например, с множеством объектов, теней, градиентов) представляет собой сложную задачу.
  – **Ограниченное количество размеченных данных:** Обучение эффективных OCR-моделей требует больших объемов размеченных данных, получение которых дорого и трудоемко.

1.2 semi-supervised предобучение моделей

Полуавтоматическое предобучение моделей (semi-supervised pretraining) представляет собой перспективный подход в машинном обучении, который стремится использовать преимущества как размеченных, так и неразмеченных данных для повышения эффективности моделей.

- Суть метода

  – **Предобучение на неразмеченных данных:** Модель обучается на большом объеме неразмеченных данных, используя методы самообучения (self-supervised learning). Примеры методов: предсказание замаскированных слов (masked language modeling), восстановление поврежденных изображений (image inpainting) и т.д.
  – **Дообучение на размеченных данных:** Предобученная модель дообучается на меньшем объеме размеченных данных для решения конкретной задачи.

- Преимущества semi-supervised предобучения

  – **Эффективное использование неразмеченных данных:** Позволяет извлекать ценную информацию из неразмеченных данных, которых обычно гораздо больше, чем размеченных.

  - **Улучшение обобщающей способности:** Предобучение на больших объемах данных помогает модели изучать более общие и устойчивые представления, что повышает ее обобщающую способность.
  - **Сокращение потребности в размеченных данных:** Дообучение предобученной модели требует меньше размеченных данных для достижения высокой производительности.

- Применение в распознавании текста

  - **Маскирование патчей изображения:** Методы, основанные на маскировании части входного изображения (например, MAE[1]), позволяют модели изучать контекстуальные связи и восстанавливать скрытую информацию.
  - **Маскирование последовательностей:** Аналогично masked language modeling в NLP, можно маскировать токены в последовательности символов и обучать модель их предсказывать.

1.3 Стратегии маскирования в нейронных сетях

Маскирование (masking) – это важный метод, используемый в нейронных сетях для выборочной обработки или игнорирования информации. Он широко применяется как на этапе предобучения, так и во время обучения модели для решения конкретных задач.

Существует несколько уровней, на которых можно применять маскирование в нейронных сетях:

- Маскирование на уровне входных данных

  - **Маскирование патчей изображения (Image Patch Masking):** Используется преимущественно в моделях компьютерного зрения. Случайные патчи (участки) изображения заменяются заглушками (например, нулями, средним значением пикселей или случайным шумом).

    Пример: Masked Autoencoders [1], SimMIM [2].
  - **Маскирование токенов (Token Masking):** Применяется в моделях обработки естественного языка (NLP). Случайные токены (слова или подслова) в последовательности заменяются специальным токеном MASK.

    Пример: BERT [3].
  - **Маскирование последовательностей (Sequence Masking):** Используется для работы с последовательностями разной длины. Элементы последовательности, следующие за определенной точкой, маскируются.

    Пример: RNN для обработки текста с переменной длиной.

- Маскирование на уровне скрытых представлений (Latent Representations)

  - **Маскирование признаков (Feature Masking):** Внутренние признаки (features), получаемые на промежуточных слоях нейронной сети, выборочно маскируются.

    Пример: Dropout (можно рассматривать как форму маскирования признаков).
  - **Маскирование внимания (Attention Masking):** В архитектурах с механизмом внимания (attention mechanism), маска используется для предотвращения внимания к определенным частям входной последовательности.

    Пример: Transformer для машинного перевода, где маска предотвращает внимание к словам, расположенным правее текущего обрабатываемого слова.

- Преимущества маскирования
  - **Регуляризация:** Предотвращает переобучение, вынуждая модель обучаться более общим и устойчивым представлениям.
  - **Самообучение (Self-Supervised Learning):** Маскирование создает искусственную задачу восстановления скрытой информации, что позволяет модели обучаться на неразмеченных данных.
  - **Обработка вариативности данных:** Позволяет эффективно работать с данными разной длины, пропущенными значениями и другими видами вариативности.

1.4 Стратегии маскирования изображений

Маскирование изображений (MIM) – это подход к самообучению моделей компьютерного зрения, основанный на принципе маскирования части изображения и обучении модели на его восстановление.

- Общая схема MIM
  1. **Маскирование:** Часть изображения скрывается маской.
  2. **Кодирование:** Замаскированное изображение подаётся на вход энкодера, который формирует скрытое представление (latent representation).
  3. **Декодирование:** Декодер получает скрытое представление и пытается восстановить исходное изображение (или его замаскированные части).
  4. **Обучение:** Модель обучается минимизировать разницу между восстановленным и оригинальным изображением.

- Различные стратегии маскирования в MIM
  - Размеры и форма маски:

    * Блочная маска (Block Masking): Изображение разбивается на блоки, и некоторые блоки скрываются.
      Пример: BEiT [4].
    * Случайная маска (Random Masking): Пиксели маскируются случайным образом с определенной вероятностью.
      Пример:MAE [1].
    * Структурированная маска (Structured Masking): Маска имеет определенную структуру, например, линии, круги, сетки.
      Пример:CutOut [5].
  - Тип маски:
    * Бинарная маска (Binary Masking): Пиксели либо скрыты, либо нет (1 или 0).
    * Непрерывная маска (Continuous Masking): Интенсивность пикселей маскируется частично, например, умножается на значение от 0 до 1.
  - Стратегии предсказания:
    * Восстановление пикселей (Pixel Reconstruction): Модель предсказывает значения пикселей замаскированных областей.
    * Предсказание признаков (Feature Reconstruction): Модель предсказывает признаки (features) скрытых областей на некотором уровне энкодера.
    * Предсказание токенов (Token Prediction): Скрытое представление квантуется в дискретные токены, и модель предсказывает эти токены.

- Преимущества MIM
  - Эффективное самообучение: Позволяет обучать модели на огромных объемах неразмеченных изображений.
  - Улучшение качества представлений: Модели, обученные с использованием MIM, формируют более информативные и устойчивые представления изображений.
  - Широкая применимость: Предварительно обученные модели MIM могут использоваться для решения различных задач компьютерного зрения: классификации, обнаружения объектов, сегментации и др.

1.5 Определения понятий

1. **Патч изображения (Image Patch):** Небольшой фрагмент изображения, обычно квадратной формы, на которые разбивается изображение для обработки в моделях компьютерного зрения, таких как Vision Transformers [6].

2. **Признаки (Features):** Измеримые характеристики данных, которые модель использует для своего обучения и принятия решений. В контексте OCR, признаки могут включать в себя форму, размер, текстуру и расположение символов.

3. **Внимания (Attention):** Механизм, используемый в нейронных сетях, позволяющий модели сосредоточиться на наиболее важных частях входных данных при выполнении задачи.

4. **Mask ratio:** Параметр в методах маскирования, определяющий долю входных данных, которая будет скрыта от модели во время обучения.

5. **Архитектура модели (Model Architecture):** Описание структуры и организации нейронной сети, включая типы используемых слоев, их количество, связи между ними и другие параметры.

6. **Точность распознавания (Accuracy):** Метрика, используемая для оценки качества работы OCR-систем. Измеряется как процент правильно распознанных символов или слов в тексте.

7. **Японский язык:** Язык с иероглифической системой письма, где каждый символ может представлять целый слог или слово. Распознавание японского текста представляет собой сложную задачу для OCR из-за большого числа символов и их сложной структуры.

8. **Размер патча:** Параметр, определяющий размер патчей изображения, на которые оно разбивается для обработки в моделях компьютерного зрения.

9. **Нейронная сеть (Neural Network):** Вычислительная модель, вдохновленная структурой и функциями мозга, состоящая из взаимосвязанных узлов (нейронов), организованных в слои. Каждый нейрон выполняет простую математическую операцию над своим входом, а связи между нейронами имеют веса, которые корректируются в процессе обучения для выполнения целевой задачи.

10. **Сверточная нейронная сеть (CNN):** Специализированный тип нейронной сети, предназначенный для эффективной обработки изображений. CNN используют операцию свертки для извлечения локальных признаков на разных уровнях абстракции, что позволяет им эффективно распознавать образы, объекты и текстуры.

11. **Transformer:** Архитектура нейронной сети, основанная на механизме внимания (attention), что позволяет ей эффективно обрабатывать последовательности данных, таких как текст или временные ряды. Transformer не использует рекуррентные связи, как RNN, а обрабатывает все элементы последовательности параллельно, что значительно ускоряет обучение и позволяет модели улавливать зависимости на больших расстояниях.

12. **Рекуррентная нейронная сеть (RNN, Recurrent Neural Network):** Тип нейронной сети, специализированный для обработки последовательных данных, таких как текст или временные ряды. RNN обладают памятью, позволяющей им учитывать предыдущие элементы последовательности при обработке текущего элемента.

13. **Энкодер-декодер (Encoder-Decoder):** распространенная архитектура нейронных сетей, состоящая из двух частей: энкодер преобразует входные данные в скрытое представление, а декодер генерирует выходные данные на основе этого представления. Широко используется в задачах перевода, генерации текста и распознавания образов.

14. **Входные представления декодера (Decoder Input Embeddings):** В моделях энкодер-декодер, декодер принимает на вход "сжатое" представление входных данных, созданное энкодером. В контексте OCR, входные представления декодера могут содержать информацию о визуальных признаках изображения.

15. **Baseline:** базовая модель или результат, с которым сравниваются результаты других моделей или экспериментов.

16. **Предобучение (Pretraining):** этап обучения модели на большом наборе данных (обычно неразмеченных), прежде чем она будет настроена под конкретную задачу. Позволяет модели выучить общие закономерности данных, что повышает эффективность обучения на меньших, специализированных наборах данных.

17. **Дообучение (Finetuning):** этап настройки предобученной модели под конкретную задачу с использованием меньшего, специализированного набора данных (обычно размеченных). На этом этапе корректируются веса модели, чтобы она лучше справлялась с заданной задачей.

18. **Vision Transformer (ViT) [6]:** тип нейронной сети, изначально разработанный для обработки изображений. В отличие от CNN, ViT разбивает изображение на патчи и обрабатывает их как последовательности, используя механизм внимания.

19. **CTCLoss (Connectionist Temporal Classification Loss): [7]** функция потерь, используемая для обучения моделей распознавания последовательностей (например, текста или речи), когда нет чётких границ между элементами последовательности во входных данных.

20. **MSE loss (Mean Squared Error Loss):** функция потерь, которая вычисляет среднеквадратичную ошибку между предсказанными и целевыми значениями. Часто используется для задач регрессии.

21. **Функция потерь перекрестной энтропии (CrossEntropyLoss):** Функция потерь, часто используемая в задачах классификации, которая измеряет разницу между двумя распределениями вероятностей:

предсказанным распределением вероятностей по классам и истинным распределением. Она штрафует модель за неверные предсказания и побуждает ее выдавать вероятности, более близкие к истинным меткам.

22. **Латентные признаки (Latent Features):** Скрытые, не наблюдаемые напрямую характеристики данных, которые модель обучается извлекать. В контексте STR, латентные признаки могут отражать форму, стиль, семантику символов или их сочетаний.

23. **Промежуточные латентные признаки:** Латентные признаки, извлекаемые моделью на промежуточных слоях энкодера или декодера, а не только на конечном слое.

24. **Edit Distance(расстояние Левенштейна):** метрика, используемая для измерения различия между двумя строками (последовательностями символов). Она определяется как минимальное количество операций редактирования, необходимых для преобразования одной строки в другую.

25. **Character Error Rate (CER):** метрика, которая представляет собой нормированное расстояние Левенштейна между распознанным и эталонным текстом.

26. **Character Accuracy (CharAcc):** 100 - CER.

27. **Word Accuracy (WordAcc):** метрика, которая представляет долю слов, для которых расстояние Левенштейна между распознанным и эталонным словом равно нулю (т.е. слова совпадают полностью).

28. **Fragment Accuracy (FragAcc):** точность распознавания на уровне фрагментов текста.

29. **Hidden size:** количество признаков (измерений) в векторах скрытого состояния, которые обрабатываются на каждом слое энкодера и декодера.

30. **Слой (layer):** блок обработки информации, состоящий из механизмов внимания и полносвязных сетей, которые применяются последовательно для анализа связей между элементами последовательности и формирования их контекстуального представления.

31. **Размерность feedforward:** размер скрытого слоя в полносвязной сети, которая применяется к каждому токену внутри слоя энкодера или декодера для нелинейного преобразования признаков. Обычно размерность feedforward в несколько раз больше, чем hidden size модели.

32. **Эпоха обучения:** один проход по всему набору данных во время обучения модели, включающий в себя подачу всех примеров для обучения (как правило, разделенных на батчи) и обновление весов модели на основе полученных результатов.

33. **Батч данных (batch):** небольшая порция данных из всего набора, которая подается на вход модели машинного обучения за один раз для вычисления градиентов и обновления весов модели во время обучения.

34. **Обучение модели:** итеративный процесс настройки параметров (весов) модели машинного обучения на основе обучающих данных с целью минимизации ошибки на данных, которые модель не видела ранее.

35. **Контекстуальные связи:** зависимости и взаимосвязи между элементами данных, при которых значение и интерпретация одного элемента определяются окружающими его элементами.

36. **Masked Language Modeling, MLM:** метод обучения языковых моделей, при котором случайные слова во входном тексте маскируются (заменяются специальным токеном, например, [MASK]), а модель должна предсказать эти замаскированные слова, основываясь на контексте оставшихся слов.

37. **Переобучение (overfitting):** явление в машинном обучении, когда модель слишком хорошо изучает обучающие данные, включая шум и случайные отклонения. В результате модель показывает отличные результаты на обучающих данных, но плохо работает на новых, невиданных данных.

38. **Латентное представление (latent representation):** сжатое, закодированное представление данных, которое захватывает наиболее важные характеристики исходной информации, часто в форме, недоступной для прямого человеческого понимания.

39. **Dropout:** метод регуляризации, используемый в нейронных сетях для предотвращения переобучения. Он заключается в случайном "выключении"(игнорировании) определенной доли нейронов во время каждого шага обучения.

40. **Компьютерное зрение (Computer Vision, CV):** междисциплинарная область, занимающаяся разработкой теории и методов, позволяющих компьютерам "видеть"и интерпретировать визуальную информацию из окружающего мира, получаемую с помощью камер и других сенсоров.

41. **Контролируемое обучение (supervised learning):** парадигма машинного обучения, в которой алгоритм строит модель, отображающую входные данные в выходные, оптимизируя свои параметры на основе набора данных с известными парами "вход - желаемый выход".

42. **Транскрипция исторических документов:** задача автоматического преобразования рукописных или машинописных исторических документов в машиночитаемый текстовый формат с использованием методов компьютерного зрения и обработки естественного языка.

2 Постановка задачи

2.1 Актуальность

Распознавание текста на изображениях (OCR) является важной задачей компьютерного зрения с широким спектром практических применений. Semi-supervised learning представляет собой перспективный подход к повышению эффективности OCR-моделей, позволяя использовать как ограниченные размеченные, так и обширные неразмеченные данные. В частности, предобучение с использованием маскирования демонстрирует высокую эффективность в задачах самообучения моделей компьютерного зрения.

2.2 Проблема

Несмотря на многообещающие результаты, оптимальная стратегия маскирования для semi-supervised предобучения OCR-моделей остается открытым вопросом. Существуют различные подходы к маскированию: на уровне входных данных (патчи изображения), на уровне скрытых представлений (признаки) и их комбинации. Выбор наиболее эффективной стратегии зависит от множества факторов, включая архитектуру модели, характер данных и требования к точности распознавания.

2.3 Цель работы

Провести комплексное сравнение различных стратегий маскирования для semi-supervised предобучения моделей OCR и определить оптимальный подход для повышения точности распознавания текста на изображениях на японском языке.

2.4 Задачи исследования

1. Проанализировать существующие стратегии маскирования для предобучения моделей OCR.
2. Разработать и реализовать модификации архитектуры OCR-модели, поддерживающие различные стратегии маскирования.
3. Провести экспериментальное исследование эффективности различных стратегий маскирования на задаче распознавания японского текста.
4. Выбрать и обосновать оптимальную стратегию маскирования для semi-supervised предобучения OCR-моделей на японском языке.

2.5 Практическая значимость

Результаты исследования позволят разработать более эффективные методы обучения OCR-моделей, что актуально для широкого спектра приложений, связанных с обработкой изображений и информации, содержащей текст.

3 Обзор существующих решений

В области semi-supervised предобучения OCR и CV моделей с использованием маскирования существует ряд исследований, предлагающих различные стратегии и архитектуры.

3.1 Маскирование патчей изображения

Метод заключается в маскировании случайных патчей входного изображения. Модель должна восстановить скрытые патчи на основе видимой информации. Этот подход демонстрирует хорошие результаты в задачах компьютерного зрения, однако его эффективность для OCR может быть ниже из-за того, что текст имеет более структурированный характер, чем общие изображения.

- **MAE: [1]** Модель обучается восстанавливать случайно замаскированные патчи изображения, что позволяет ей изучать глобальные зависимости между различными частями изображения.
- **SimMIM: [2]** Предлагает упрощенный подход к маскированию изображений, используя простую линейную декодирующую голову для восстановления скрытых патчей.
- **SupMAE: [8]** В отличие от MAE, использующего только неразмеченные данные, SupMAE предлагает использовать контролируемое обучение на размеченных данных. В процессе обучения SupMAE предсказывает не пиксели изображения, а целевые метки для каждого замаскированного патча. Эксперименты показывают, что SupMAE эффективнее стандартного MAE, особенно при ограниченном количестве обучающих данных.
- **Revisiting Scene Text Recognition: A Data Perspective: [9]** В этой работе исследуется влияние объема и разнообразия данных на эффективность моделей распознавания текста. Авторы демонстрируют, что большие и разнообразные наборы данных критически важны для достижения высокой точности распознавания.

3.2 Маскирование входных представлений декодера

Этот подход заключается в маскировании части входных представлений, которые декодер получает от энкодера. Декодер должен научиться восстанавливать замаскированные части на основе контекста.

- **Lacuna Reconstruction: Self-supervised Pre-training for Low-Resource Historical Document Transcription: [10]** Представлен метод самообучения, специально разработанный для транскрипции исторических документов, где объём размеченных данных ограничен. Модель обучается восстанавливать пропущенные фрагменты текста

(лакуны), что позволяет ей адаптироваться к особенностям старинных шрифтов и стилей письма. Метод может быть эффективен для OCR, так как он позволяет модели лучше учитывать контекст при распознавании символов.

3.3 Маскирование патчей изображения и элементов последовательности текста

Некоторые работы исследуют комбинацию маскирования на разных уровнях, чтобы использовать как визуальную, так и контекстуальную информацию для более точного распознавания текста.

- **Masked Vision-Language Transformers for Scene Text Recognition: [11]** В этой работе представлена модель, которая совместно обучается на замаскированных изображениях и текстовых описаниях, что позволяет ей эффективно извлекать как визуальные, так и семантические признаки для распознавания текста.
- **MaskOCR: [12]** Предложен метод предобучения OCR-моделей с использованием маскирования как на уровне изображения, так и на уровне текста. Модель с архитектурой энкодер-декодер учится восстанавливать замаскированные патчи изображения и генерировать соответствующую текстовую последовательность. Эксперименты показали, что такой подход позволяет эффективно использовать неразмеченные данные и улучшает обобщающую способность модели.

3.4 Ограничения существующих подходов

- **Большинство работ сосредоточено на английском языке:** Исследования по маскированию для OCR на других языках, особенно с иероглифической системой письма, остаются ограниченными.
- **Не иследовано комбинирование подходов:** Комбинирование маскирования патчей изображения и входных представлений декодера может потенциально привести к наилучшим результатам, так как позволяет модели учитывать как локальную информацию (из патчей изображения), так и глобальный контекст (из входных представлений декодера).

4 Исследование и построение решения задачи

Для решения поставленной задачи – сравнительный анализ стратегий маскирования для semi-supervised предобучения моделей OCR – проведем декомпозицию на более мелкие подзадачи и представим результаты исследования предметной области.

4.1 Исследование предметной области и существующих решений

В данном разделе рассматриваются ключевые идеи из литературы, используемый набор данных и выбранные метрики оценки.

1. **Обзор литературы и выбор архитектур:** Анализ существующих работ по применению методов маскирования в задачах компьютерного зрения и распознавания текста позволил выявить следующие перспективные идеи:

   - Архитектура MAE [1]: Применение самообучения (self-supervised learning) с реконструкцией замаскированных патчей изображения демонстрирует высокую эффективность в задачах CV. Данный подход будет реализован в методе 1.
   - Архитектура SimMIM [2]: Предложенное упрощение MAE, заключающееся в подаче на вход энкодера как видимых, так и маскированных патчей, позволяет сократить сложность модели, но требует больше вычислительных ресурсов. Эта идея будет использована в методе 2.
   - Архитектура SupMAE [8]: Использование размеченных данных для обучения MAE, как показано в данной работе, может значительно повысить эффективность предобучения. Данный подход будет реализован в методе 3.
   - Маскирование входных представлений декодера: Вдохновленные работой "Lacuna Reconstruction"[10], предлагающей маскировать части представлений на выходе из энкодера, мы исследуем маскирование входных представлений декодера. Комбинируя эту идею с MAE и SimMIM, получим методы 4 и 5.
   - Совместное маскирование патчей изображения и входных представлений декодера: Работы MaskOCR [11] и MVLT [11], демонстрирующие эффективность совместного обучения на замаскированных изображениях и текстовых последовательностях, побудили нас исследовать комбинацию маскирования на разных уровнях модели. Эта идея будет реализована в методе 6.

2. **Датасет:** Для обучения и оценки моделей будет использован внутренний датасет с изображениями, содержащими печатные надписи на японском языке.

   Тренировочная выборка: 2 миллиона изображений.

Тестовая выборка: 150 тысяч изображений.

Рис. 1: Пример изображения с печатным текстом из тестового набора данных.

3. **Выбор метрик оценки:** Для оценки качества OCR-моделей будут использоваться следующие метрики:
   - Character Accuracy на основе Edit Distance: Данная метрика позволит оценить точность распознавания на уровне символов с учетом расстояния Левенштейна.
   - Fragment Accuracy: Эта метрика будет использоваться для оценки точности распознавания на уровне фрагментов текста, что позволит более детально проанализировать работу моделей.

4.2 Разработка и реализация baseline-модели

1. **Выбор базовой архитектуры:** В условиях ограниченных вычислительных ресурсов и с целью оптимизации экспериментальной работы в качестве базовой архитектуры была выбрана модель Encoder-Decoder, представленная на Рисунке 2. В качестве энкодера и декодера была использована архитектура ViT [6] с четырьмя слоями, размерностью скрытого состояния 256 и размерностью feedforward 1024.

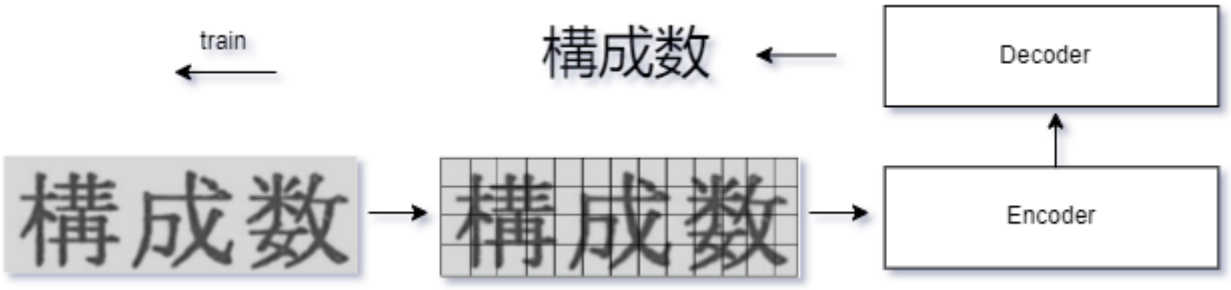

Рис. 2: Архитектура baseline-модели. Изображение разбивается на патчи размером 16x16 пикселей. Каждый патч подается на вход энкодеру. Выход декодера передается на вход линейному слою, который отображает каждое внутреннее представление декодера в вектор размерности, равной мощности алфавита.

2. **Реализация модели:** Программный код baseline-модели реализован на основе внутреннего фреймворка, предоставленного кафедрой. Фреймворк базируется на модулях PyTorch и PyTorch Lightning. В качестве функции потерь выбрана функция Connectionist Temporal Classification

Loss (CTC Loss) в связи с тем, что каждый патч изображения может содержать несколько символов или не содержать их вовсе.

3. **Обучение и оценка baseline-модели:** Процесс обучения модели проводился в течение 80 эпох с размером батча, равным 64. Количество итераций обучения, обрабатываемых моделью за одну эпоху, составило 20000.

   Результаты оценки модели:

   - CharAcc (точность распознавания символов): 95.8%
   - FragAcc (точность распознавания фрагментов текста): 79%

4.3 Разработка и реализация моделей с маскированием

4.3.1 Маскирование патчей изображения

В рамках исследования были изучены три метода маскирования патчей изображения для обучения моделей OCR с использованием подходов, основанных на автоэнкодерах.

- Метод 1: Маскирование с реконструкцией (MAE)

  Первый метод (Рисунок 3) основан на архитектуре (MAE) [1]. Процесс обучения состоит из двух этапов: предварительного обучения (pretraining) и тонкой настройки (finetuning).

  – Предварительное обучение:

    1. Маскирование:
       Случайным образом выбирается определенный процент патчей изображения и заменяется нулями (маскируется). Позиции замаскированных патчей сохраняются.
    2. Кодирование: Незамаскированные патчи подаются на вход энкодера.
    3. Декодирование: К выходным данным энкодера добавляются замаскированные патчи, после чего информация передается на вход декодера.
    4. Реконструкция: Модель обучается на задачу реконструкции исходного изображения, используя среднеквадратичную ошибку (MSE Loss) между реконструированным и исходным изображением.

  – Тонкая настройка:

    1. Маскирование патчей изображения не производится.
    2. Модель обучается на задачу распознавания текста с использованием CTC Loss.

  Для каждого этапа используется отдельный декодер.

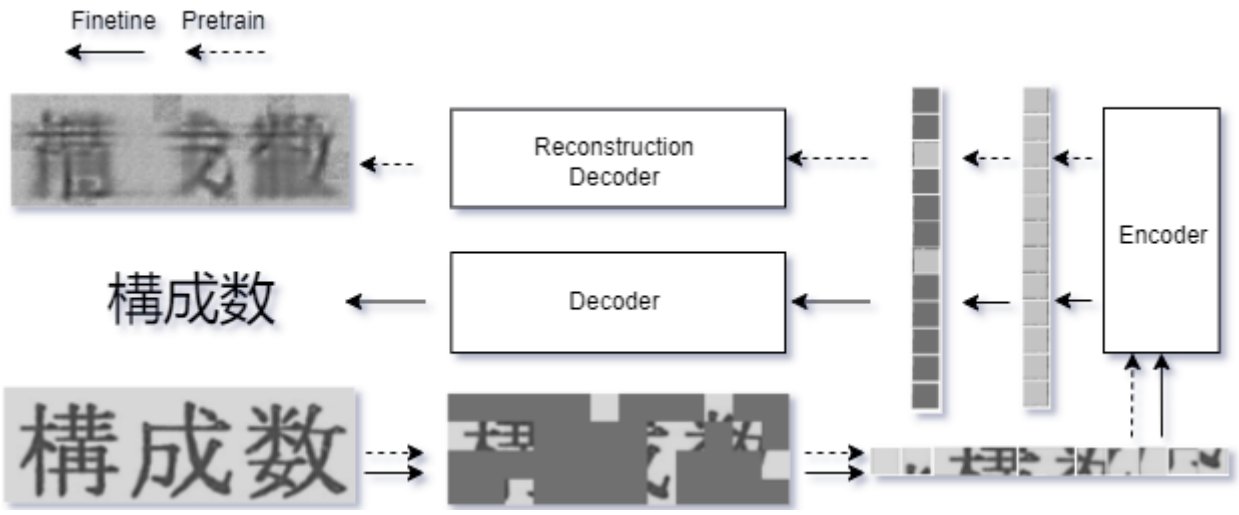

Рис. 3: Архитектура модели с маскированием патчей изображения (MAE).

- Метод 2: Маскирование с полным входом (SimMIM)

  Второй метод (Рисунок 4) основан на архитектуре SimMIM [2] и отличается от первого метода тем, что на вход энкодера подаются все патчи изображения, включая замаскированные. Остальные этапы предварительного обучения и тонкой настройки аналогичны методу 1.

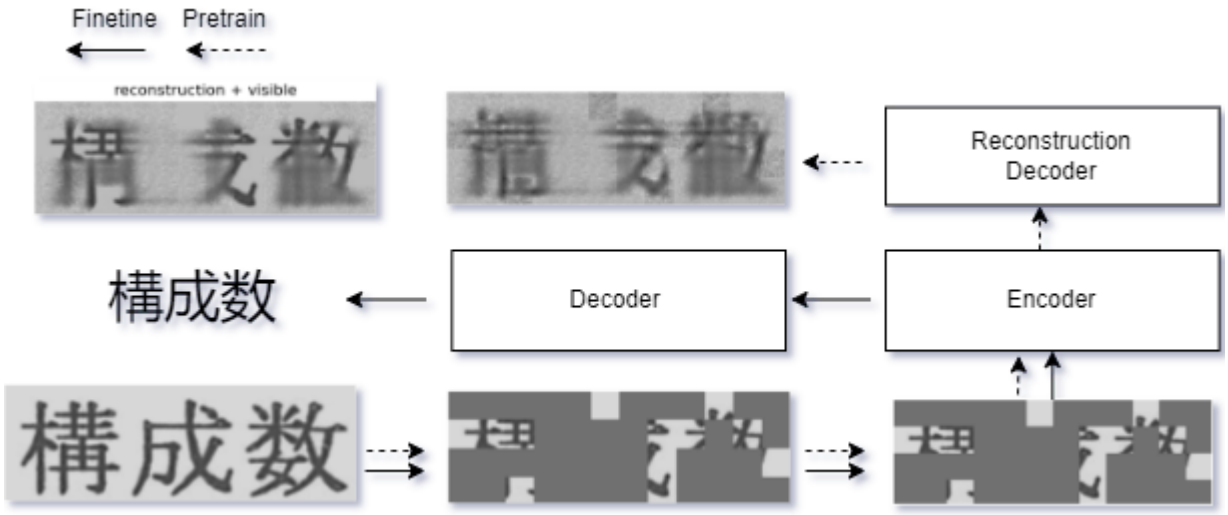

Рис. 4: Архитектура модели с маскированием патчей изображения (SimMIM).

- Метод 3: Маскирование с совместным обучением (SupMAE)

  Третий метод, представленный на Рисунке 5, основан на архитектуре SupMAE [8] и отличается от первого метода тем, что на этапе предварительного обучения модель обучается одновременно на двух функциях потерь: MSE Loss для реконструкции изображения и CTC Loss для распознавания текста. Этап тонкой настройки аналогичен методам 1 и 2.

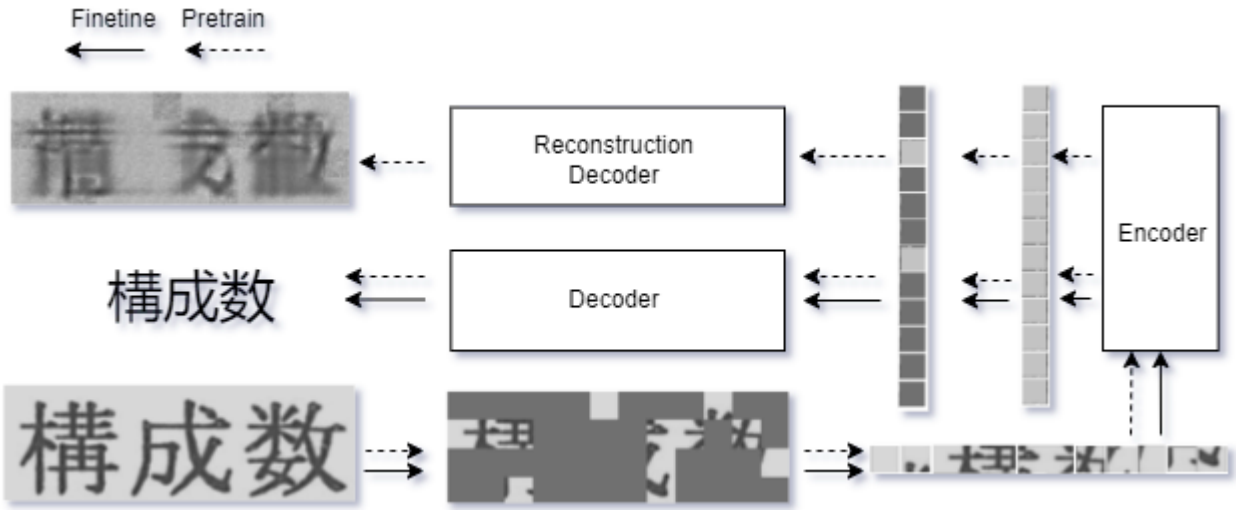

Рис. 5: Архитектура модели с маскированием патчей изображения (SupMAE).

4.3.2 Маскирование входных представлений декодера

В дополнение к маскированию патчей изображения, были исследованы два метода, основанные на маскировании входных представлений декодера.

- Метод 4: Маскирование представлений декодера с CTC-обучением

  Четвертый метод (Рисунок 6) реализует идею маскирования входных представлений декодера и также состоит из двух этапов обучения:

  – Предварительное обучение:

  1. Формирование патчей: Изображение разбивается на патчи размером 16x16 пикселей.
  2. Кодирование: Патчи подаются на вход энкодера.
  3. Маскирование представлений: На выходе из энкодера, полученные промежуточные представления случайно маскируются (зануляются) с определенной вероятностью (mask ratio).
  4. Декодирование: Замаскированные представления передаются на вход декодера.
  5. Распознавание текста: Модель обучается с использованием CTC Loss.

  – Тонкая настройка:

  1. Маскирование представлений декодера не производится.
  2. Модель продолжает обучаться на задачу распознавания текста с использованием CTC Loss.

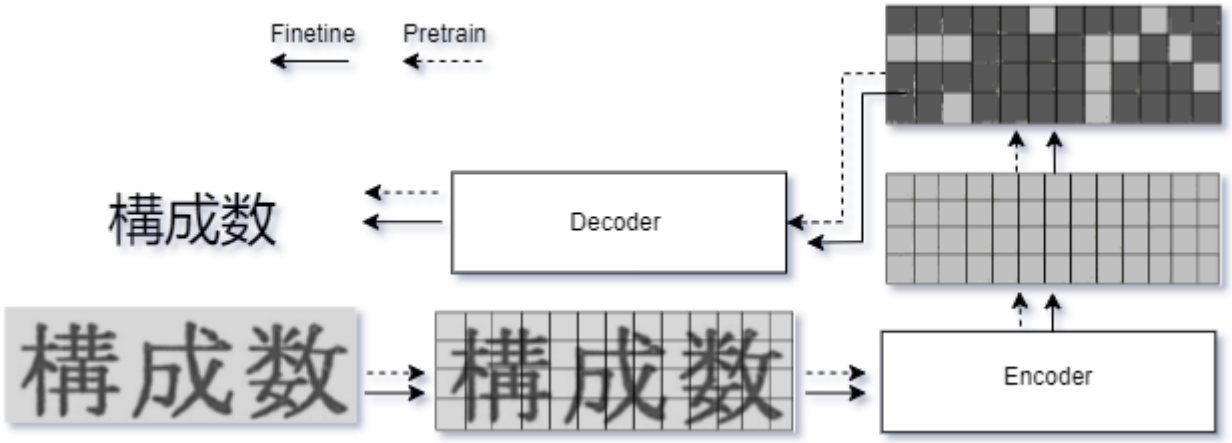

Рис. 6: Архитектура модели с маскированием входных представлений декодера (CTC-обучение).

- Метод 5: Маскирование представлений декодера с совместным обучением

  Пятый метод (Рисунок 7) отличается от четвертого тем, что на этапе предварительного обучения модель обучается одновременно на двух функциях потерь: CTC Loss для распознавания текста и MSE Loss для реконструкции входных представлений декодера. Этапы тонкой настройки аналогичны методу 4.

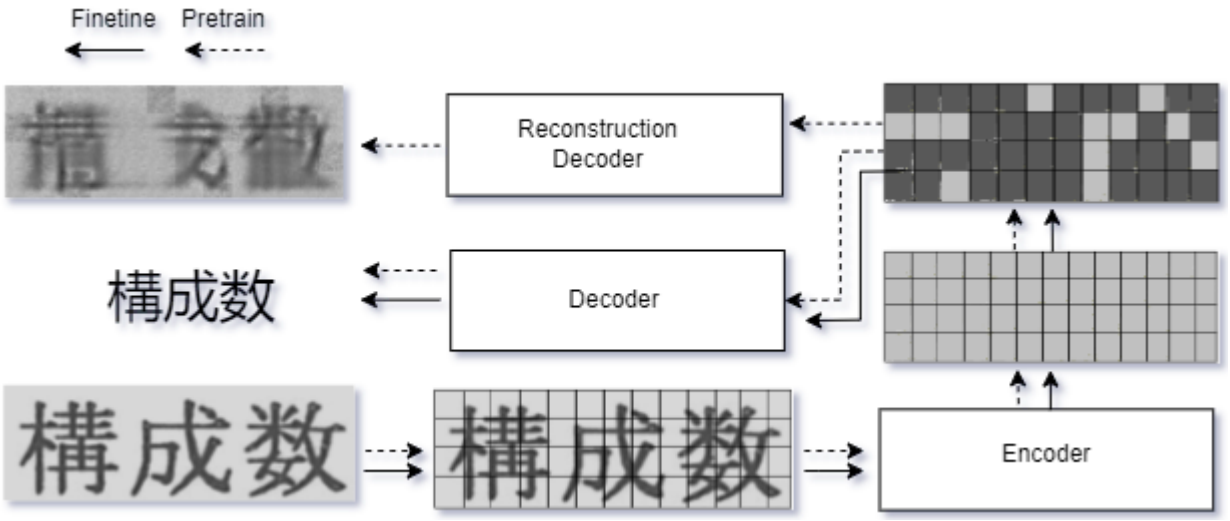

Рис. 7: Архитектура модели с маскированием входных представлений декодера (совместное обучение).

4.3.3 Комбинированное маскирование

В рамках исследования был изучен метод, комбинирующий маскирование патчей изображения и входных представлений декодера.

- Метод 6:

  Шестой метод (Рисунок 8) реализует идею совместного маскирования патчей изображения и входных представлений декодера. Процесс

обучения также разделен на два этапа: предварительное обучение и тонкую настройку.

– Предварительное обучение:
  1. Формирование и маскирование патчей: Изображение разделяется на патчи размером 16x16 пикселей. Случайным образом выбирается и маскируется определенный процент патчей (mask ratio).
  2. Кодирование: Патчи, как замаскированные, так и не замаскированные, подаются на вход энкодера.
  3. Маскирование представлений: На выходе из энкодера полученные промежуточные представления также случайно маскируются с определенной вероятностью (mask ratio).
  4. Декодирование: Замаскированные представления передаются на вход декодера.
  5. Распознавание текста: Модель обучается с использованием CTC Loss.

– Тонкая настройка:
  1. Маскирование патчей изображения и представлений декодера не производится (mask ratio равен нулю).
  2. Модель продолжает обучаться на задачу распознавания текста с использованием CTC Loss.

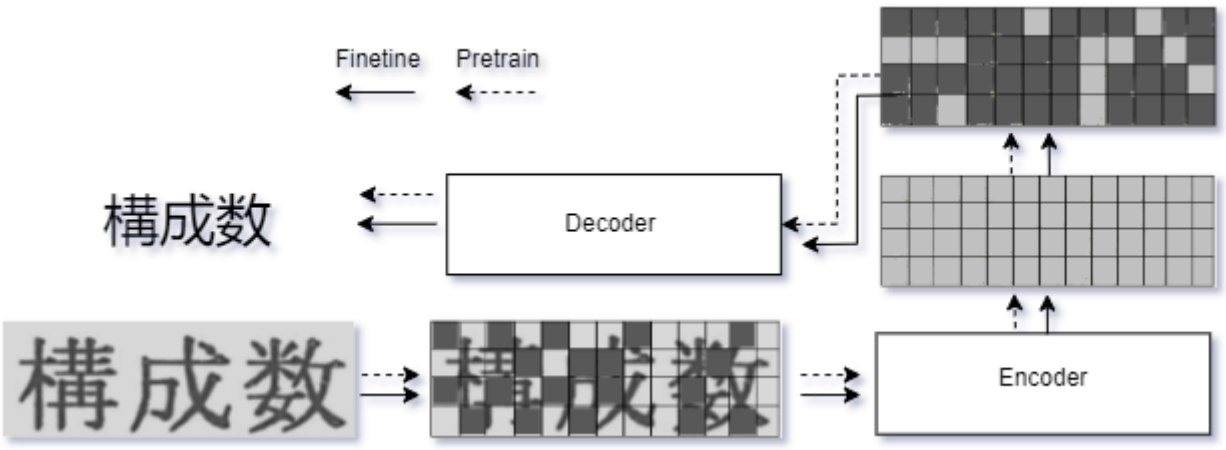

Рис. 8: Архитектура модели с совместным маскированием патчей изображения и входных представлений декодера.

4.4 Проведение экспериментов и анализ результатов

Данный подраздел посвящен описанию проведенных экспериментов по сравнению различных стратегий маскирования для semi-supervised обучения моделей OCR на японском языке. Представлены результаты экспериментов и сформулированы выводы.

1. Обучение моделей с маскированием:

   Обучение моделей с различными стратегиями маскирования проводилось с использованием следующих параметров:

   Количество эпох:

   - Предварительное обучение (pretraining): 60
   - Тонкая настройка (finetuning): 60

   Размер маскированного патча: 16x16 px

2. Оценка качества моделей:

   Для оценки качества OCR-моделей были использованы метрики CharAcc (Character Accuracy) и FragAcc (Fragment Accuracy). Результаты представлены в Таблице 1.

№ метода	Маскирование	Pretrain	Mask ratio (%)	CharAcc (%)	FragAcc (%)
0(baseline)	нет	нет	0	95.8	79
1	Патчи изображения	MSE	25	96.2	80.2
1	Патчи изображения	MSE	75	96.35	80.9
2	Патчи изображения	MSE	75	96.25	80.5
3	Патчи изображения	MSE+CTC	25	96.4	81.15
4	Представления декодера	CTC	75	96.45	81.34
5	Представления декодера	MSE+CTC	25	96.4	81.2
6	Комбинирование	CTC	33 и 33	96.5	81.36

Таблица 1: Сравнительный анализ эффективности различных стратегий маскирования на основе экспериментальных данных.

3. Выводы:

   - Применение стратегий маскирования оказывает положительное влияние на качество распознавания текста. Все рассмотренные методы, использующие маскирование, превзошли базовую модель по метрикам CharAcc и FragAcc, что свидетельствует об эффективности данного подхода при semi-supervised обучении.
   - Маскирование на уровне патчей изображения с функцией потерь MSE на этапе pretrain демонстрирует высокую эффективность. Наблюдается тенденция к улучшению показателей качества распознавания с увеличением доли маскируемых патчей.
   - Различные архитектуры моделей с маскированием патчей изображения демонстрируют сравнимые результаты.
   - Введение функции потерь CTC в процесс обучения с маскированием, как на уровне патчей изображения, так и на уровне представлений декодера, позволяет получить дополнительный прирост точности.

- Маскирование на уровне представлений декодера, особенно с использованием функции потерь CTC, показало себя более эффективным, чем маскирование патчей изображения. Данный результат указывает на перспективность использования маскирования на более высоких семантических уровнях при обучении OCR-моделей.
- Наилучшие результаты были достигнуты при комбинировании маскирования на уровне патчей изображения и на уровне представлений декодера, что подтверждает гипотезу о целесообразности совместного использования различных подходов к маскированию.

Полученные результаты свидетельствуют об успешном решении поставленной задачи. Разработаны и исследованы различные стратегии маскирования для semi-supervised обучения моделей OCR на японском языке.

5 Описание практической части

В данном разделе представлено частичное описание программной реализации исследования, включая выбор инструментов разработки и архитектуру кода.

- **Инструменты разработки:**

  Для реализации и исследования были выбраны следующие инструменты:

  Язык программирования: Python

  Библиотеки:

  - PyTorch: Фреймворк глубокого обучения, используемый для создания, обучения и оценки нейронных сетей. Обеспечивает автоматическое дифференцирование, работу с тензорами и GPU-ускорение.
  - OpenCV (cv2): Библиотека компьютерного зрения, применяемая для загрузки, предобработки и манипуляции с изображениями.
  - NumPy: Библиотека для научных вычислений, предоставляющая функции для работы с многомерными массивами, линейной алгеброй и математическими операциями.

  Выбор Python обусловлен его простотой, читаемостью и широкой распространенностью в сообществе машинного обучения. PyTorch предоставляет гибкий и эффективный фреймворк для работы с нейронными сетями, а OpenCV и NumPy — необходимые инструменты для обработки изображений и численных данных.

- **Оптимизатор и планировщик скорости обучения**

  В качестве оптимизатора использовался алгоритм AdamW.

  Параметры оптимизатора:

  - Скорость обучения (lr): 0.001
  - Коэффициенты экспоненциального усреднения для моментов градиента ($\beta 1$, $\beta 2$): 0.9 и 0.999 соответственно
  - Коэффициент затухания весов (weight_decay): 0.01

  Для планирования скорости обучения применялся метод MultiStepLR, который позволяет уменьшать скорость обучения в определенные моменты времени.

  В качестве параметров планировщика использовались:

  - Множитель скорости обучения (γ): 0.1
  - Эпохи, на которых происходит уменьшение скорости обучения: 40 и 55

- **Вычислительные ресурсы:**

  Для проведения экспериментов и обучения моделей использовалось следующее аппаратное обеспечение:

  Графический процессор: NVIDIA GeForce RTX 4060 Ti. Объём видеопамяти: 16 ГБ.

  Процесс обучения моделей сходился приблизительно за 4 дня (120 эпох) на указанном оборудовании. Во время обучения каждая модель занимала около 14 Гб видеопамяти.

- **Архитектура кода:**

  – **Преобразование изображения в набор патчей**

  Для преобразования входного изображения в набор патчей и их встраивания в пространство признаков использовался слой нейронной сети PatchEmbed.

  Принцип работы:

  1. **Инициализация слоя:**
     * Принимает параметры, определяющие размер изображения, размер патча, количество входных каналов, размерность пространства признаков.
     * Вычисляет размер сетки патчей и общее количество патчей.
     * Создает сверточный слой, который отвечает за извлечение патчей и формирование их представлений.
  2. **Прямой проход (forward):**
     * Принимает на вход батч изображений.
     * Дополняет изображение по краям (padding), чтобы его размер точно делился на размер патча.
     * Применяет сверточный слой к входному изображению. Сверточный слой выполняет одновременно две функции:
       · Делит изображение на патчи с помощью заданного размера ядра.
       · Проецирует каждый патч в пространство признаков заданной размерности.
     * Возвращает тензор, содержащий представления всех патчей входного изображения.

  В итоге, слой PatchEmbed преобразует входное изображение в последовательность векторов-признаков, где каждый вектор представляет собой один патч изображения.

  – **Случайное маскирование патчей (Random Masking)**

  Для реализации стратегии маскирования патчей используется функция random_masking.

  Принцип работы:

1. **Инициализация:**
   * Принимает на вход тензор патчей изображения x размерности (N, L, D), где:
     · N - размер батча,
     · L - количество патчей в последовательности,
     · D - размерность вектора-признака патча.
   * Принимает на вход коэффициент маскирования, который определяет долю патчей, подлежащих маскированию.
   * Вычисляет количество патчей, которые останутся не замаскированными (len_keep).
   * Генерирует тензор случайных чисел той же размерности, что и x, значения которого распределены равномерно в интервале $0, 1$.
2. **Маскирование:**
   * Сортирует случайные числа в порядке возрастания отдельно для каждого изображения в батче с помощью $torch.argsort$.
   * Оставляет в качестве не замаскированных патчей len_keep патчей, соответствующих наименьшим случайным числам.
   * Формирует тензор маски, где 0 соответствует не замаскированному патчу, а 1 - замаскированному.
3. **Результат:**
   * Возвращает тензор с замаскированными патчами и тензор маски.

Таким образом, функция random_masking случайным образом выбирает и маскирует определенную долю патчей входного изображения.

6 Заключение

Проведенное исследование продемонстрировало эффективность применения стратегий маскирования для semi-supervised обучения моделей оптического распознавания символов (OCR) на японском языке. Анализ шести различных архитектур, реализующих маскирование на уровне патчей изображения и/или входных представлений декодера, показал, что все они превосходят по точности распознавания baseline модель, не использующую маскирование.

6.1 Ключевые выводы

1. Маскирование как эффективный метод аугментации данных: Применение стратегий маскирования способствует изучению модели более обобщенных и устойчивых представлений, что подтверждается превосходством всех архитектур с маскированием над baseline.

2. Синергетический эффект комбинированного подхода: Совместное маскирование патчей изображения и представлений декодера демонстрирует наилучшую точность распознавания, подчеркивая потенциал комбинирования разных подходов к маскированию.

3. Перспективность маскирования представлений декодера: Архитектуры, маскирующие входные представления декодера, продемонстрировали высокую эффективность, превосходя даже методы с маскированием изображений.

6.2 Направления для дальнейших исследований

- Оптимизация параметров маскирования: Дальнейшее изучение влияния размера маскируемых патчей, значения mask ratio и других параметров на эффективность обучения.

- Применение к другим языкам и наборам данных: Оценка эффективности исследованных подходов на более широком спектре языков и типов данных.

- Разработка новых стратегий маскирования: Исследование более сложных и адаптивных методов маскирования, учитывающих специфику задачи OCR и структуру языка.

Полученные результаты открывают новые возможности для развития более точных и эффективных систем OCR, что имеет высокую практическую значимость для многих областей, связанных с автоматизацией обработки текстовой информации.

Список литературы

[1] Masked autoencoders are scalable vision learners / Kaiming He, Xinlei Chen, Saining Xie et al. // *arXiv preprint arXiv:2111.06377*. — 2021.

[2] SimMIM: A simple framework for masked image modeling / Zhenda Xie, Zheng Zhang, Yue Cao et al. — 2022. — Pp. 15233–15242.

[3] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova // *arXiv preprint arXiv:1810.04805*. — 2018.

[4] Beit: Bert pre-training of image transformers / Hangbo Bao, Li Dong, Furu Wei et al. // *arXiv preprint arXiv:2106.08254*. — 2021.

[5] *DeVries, Terrance*. Improved regularization of convolutional neural networks with cutout / Terrance DeVries, Graham W Taylor, Deng Guo // *arXiv preprint arXiv:1708.04552*. — 2017.

[6] An image is worth 16x16 words: Transformers for image recognition at scale / Alexey Dosovitskiy, Lucas Beyer, Alexander Koltun et al. // *arXiv preprint arXiv:2010.11929*. — 2020.

[7] Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks / Alex Graves, Santiago Fernández, Faustino Gomez, Jürgen Schmidhuber // *Proceedings of the 23rd international conference on Machine learning*. — 2006. — Pp. 369–376.

[8] Supmae: Supervised masked autoencoders are efficient vision learners / Hang Gao, Xing Wang, Weijian Xie et al. // *arXiv preprint arXiv:2212.13367*. — 2022.

[9] Revisiting scene text recognition: A data perspective / Jongmin Baek, Geewook Kim, Junyeop Lee et al. // *arXiv preprint arXiv:1904.01906*. — 2019.

[10] Lacuna Reconstruction: Self-supervised Pre-training for Low-Resource Historical Document Transcription / Benedikt Stammer, Simon Wickham, Caspar Garz, Tobias Fingscheidt // *arXiv preprint arXiv:2307.11307*. — 2023.

[11] Masked vision-language transformers for scene text recognition / Kevin Lyons, Gautam Nawhal, Alexei Baevski et al. // *arXiv preprint arXiv:2202.13120*. — 2022.

[12] Maskocr: Text recognition with masked visual-linguistic modeling / Yuanhao Li, Jianhua Sun, Yue Meng et al. // *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. — 2022. — Pp. 17426–17435.

Приложение

6.3 Дополнительные исследования

- **Уменьшение размера патчей**

  В ходе предварительных экспериментов было замечено, что использование патчей размером 16x16 пикселей может приводить к тому, что некоторые символы на изображении полностью перекрываются одним маскированным патчем. Это может негативно сказаться на качестве обучения модели, поскольку информация о таких символах будет потеряна.

  Для решения данной проблемы был исследован подход с использованием патчей меньшего размера - 8x8 пикселей. Предварительные эксперименты показали, что уменьшение размера патчей позволяет снизить вероятность полного перекрытия символов и улучшить качество реконструкции изображения на этапе предварительного обучения.

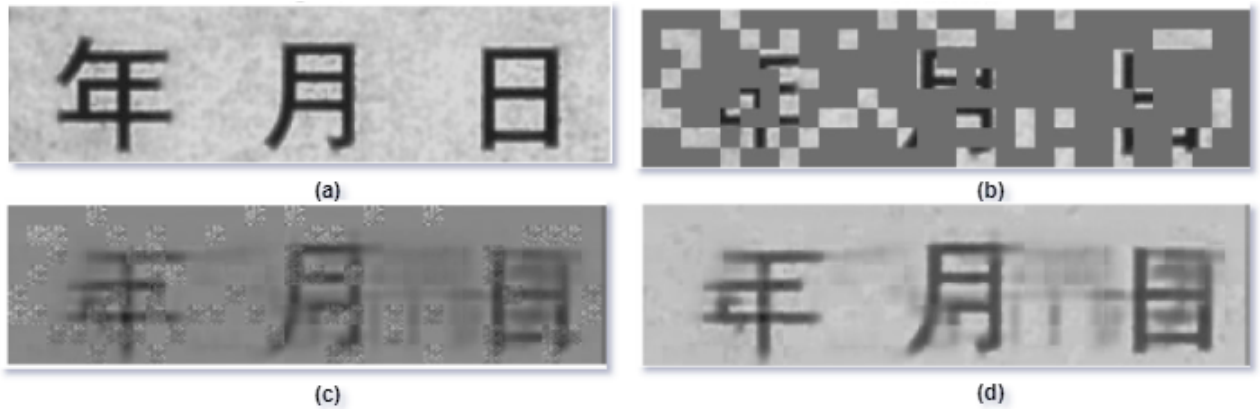

Рис. 9: Пример использования патчей меньшего размера при обучении архитектуры 1. (a) Оригинальное изображение. (b) Изображение с замаскированными патчами. (c) Реконструированное изображение. (d) Совмещенное изображение, показывающее оригинальные и реконструированные патчи. Параметры: размер патча – 8x8 пикселей, mask ratio – 75%, Эпох предварительного обучения – 60.

Однако, из-за ограничений по времени и вычислительным ресурсам, провести полное исследование эффективности данного подхода, включая этап тонкой настройки (finetune) модели для задачи распознавания текста, не удалось.