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 (STR) 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 STR. 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, размерами патчей и значениями mask ratio. Для оценки качества разработанных моделей использовалась метрика Character Accuracy (CharAcc) на тестовом наборе данных с японским текстом.

Результаты экспериментов показали, что все три рассмотренные стратегии маскирования превосходят baseline модель, обученную без использования маскирования. Наилучшие результаты были достигнуты при комбинировании маскирования патчей изображения и входных представлений декодера, что позволило достичь прироста CharAcc на X % по сравнению с baseline.

Данная работа вносит вклад в развитие области semi-supervised обучения для STR, 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.

1.1.1 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.

1.1.2 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.

1.1.3 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 Полуавтоматическое предобучение моделей (Semi-Supervised Pretraining)

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.

1.2.1 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.

1.2.2 Преимущества полуавтоматического 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.

1.2.3 Применение в распознавании текста (OCR/STR):

- **Маскирование патчей изображения:** Методы, основанные на маскировании части входного изображения (например, MAE - Masked Autoencoders), 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:

1.3.1 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 (MAE), SimMIM.
- **Маскирование токенов (Token Masking):** Применяется в моделях обработки естественного языка (NLP). Случайные токены (слова или подслова) в последовательности заменяются специальным токеном MASK. Пример:BERT, RoBERTa.
- **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.

1.3.2 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.

1.3.3 Маскирование на уровне выходов (Output Masking):

Используется для обучения моделей с несколькими выходами, где не все выходы важны для каждой обучающей выборки.

1.3.4 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 (Masked Image Modeling, MIM)

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

1.4.1 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.

1.4.2 Различные стратегии маскирования в MIM:

- Размеры и форма маски:
  - Блочная маска (Block Masking):Изображение разбивается на блоки, и некоторые блоки скрываются. **Пример:** BEiT (BERT-like Image Transformer).
  - Случайная маска (Random Masking): Пиксели маскируются случайным образом с определенной вероятностью. **Пример:** MAE (Masked Autoencoders).
  - Структурированная маска (Structured Masking): Маска имеет определенную структуру, например, линии, круги, сетки. **Пример:** CutOut.
- 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.

1.4.3 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. **Распознавание текста на изображениях (Optical Character Recognition, OCR):** Технология, позволяющая компьютерам "читать"текст на изображениях, преобразуя его в машиночитаемый формат для дальнейшей обработки.
2. **STR (Scene Text Recognition):** подвид OCR, который фокусируется на распознавании текста в естественных сценах, например, на вывесках, дорожных знаках, этикетках. Отличается от обычного OCR сложностью распознавания из-за разнообразных фонов, шрифтов, ракурсов и освещения.
3. **Полуавтоматическое обучение (Semi-Supervised Learning):** Подход к машинному обучению, использующий как размеченные (с известными ответами), так и неразмеченные данные для обучения модели, что позволяет эффективно использовать большие объемы неразмеченных данных.
4. **Предобучение с использованием маскирования (Masked Pre-training):** Метод самообучения (self-supervised learning), при котором модель обучается предсказывать замаскированные (скрытые) части входных данных, например, пиксели изображения или токены текста.
5. **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. **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.
7. **Внимания (Attention):** Механизм, используемый в нейронных сетях, позволяющий модели сосредоточиться на наиболее важных частях входных данных при выполнении задачи.
8. **Mask ratio:** Параметр в методах маскирования, определяющий долю входных данных, которая будет скрыта от модели во время обучения.
9. **Архитектура модели (Model Architecture):** Описание структуры и организации нейронной сети, включая типы используемых слоев, их количество, связи между ними и другие параметры.

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

11. **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.

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

13. **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.

14. **Сверточная нейронная сеть (CNN, Convolutional Neural Network):** 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.

15. **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.

16. **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.

17. **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.

18. **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.

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

20. **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.

21. **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.

22. **Vision Transformer (ViT):** 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.

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

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

25. **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.

26. **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.

27. **Промежуточные латентные признаки (Intermediate Latent Features):** Латентные признаки, извлекаемые моделью на промежуточных слоях энкодера или декодера, а не только на конечном слое.

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

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

30. **Char Accuracy (CharAcc):** 100 - CER.

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

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

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

34. **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.

35. **Размерность feedforward network:** 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 (размер скрытого состояния) модели.

36. **PyTorch:** open-source библиотека машинного обучения для Python, разработанная Facebook, которая предоставляет широкие возможности для создания и обучения нейронных сетей, включая гибкое динамическое вычисление графов, GPU ускорение и богатый набор инструментов.

37. **PyTorch Lightning:** высокоуровневая библиотека, построенная на основе PyTorch, которая упрощает и ускоряет разработку и обучение нейронных сетей за счет автоматизации шаблонного кода, структурирования проекта и предоставления удобных инструментов для масштабирования, логирования и отладки.

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

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

40. **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.

2 Постановка задачи

2.1 Актуальность.

Распознавание текста на изображениях (Optical Character Recognition, OCR) является важной задачей компьютерного зрения с широким спектром практических применений. Полуавтоматическое обучение (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 (masked pre-training) 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, элементы последовательности), на уровне скрытых представлений (признаки, внимание) 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 Объект исследования:

Процесс semi-supervised предобучения моделей OCR с использованием маскирования.

2.6 Предмет исследования:

Влияние различных стратегий маскирования на точность распознавания текста на изображениях на японском языке.

2.7 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-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 STR может быть ниже из-за того, что текст имеет более структурированный характер, чем общие изображения.

- **Masked Autoencoders (MAE) He et al., 2021:** Модель обучается восстанавливать случайно замаскированные патчи изображения, что позволяет ей изучать глобальные зависимости между различными частями изображения.
- **SimMIM (A Simple Framework for Masked Image Modeling) Xie et al., 2021:** Предлагает упрощенный подход к маскированию изображений, используя простую линейную декодирующую голову для восстановления скрытых патчей.
- **SupMAE (Supervised Masked Autoencoders Are Efficient Vision Learners) 2022:** 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 Baek et al., 2023:** 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 Stratos et al., 2021:** 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 STR, 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 Li et al., 2022:** 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: Text Recognition with Masked Encoder-Decoder Pretraining Lyu et al., 2022:** 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 STR – проведем декомпозицию на более мелкие подзадачи.

4.1 Исследование предметной области и существующих решений

1. **Обзор литературы:**
   - Архитектура MAE показывает довольно хорошие результаты в задачах CV, за счёт self-supervised подхода. Реализуем эту идею в архитектуре 1.
   - Архитектура SimMIM наводит на мысль, что на вход энкодеру можно подавать не только видимые патчи изображения, но и маскированные, это значительно упрощает архитектуру по сравнению с MAE, но требует гораздо больше памяти и времени для стадии pretrain. Реализуем эту идею в архитектуре 2.
   - Как было показано в статье SupMAE, использование размеченных данных для обучения MAE может значительно улучшить его эффективность, Реализуем эту идею в архитектуре 3.
   - Статья Lacuna Reconstruction наводит нас на мысль, что так же можно маскировать входные представления декодера. Скрестив эту идею с предыдущими двумя получим архитектуры 4 и 5.
   - Статьи MaskOCR и MVLT навели нас на мысль, что можно скрестить маскирование патчей изображения и входных представлений декодера, что дало нам архитектуру 6.
2. **Датасет:** У нас имеется внутренний размеченный датасет с печатанными надписями на японском языке. Тренировочная выборка содержит приблизительно 2 миллиона картинок. Тестовая выборка содержит приблизительно 150 тысяч картинок.

Рис. 1: Пример надписи из тестового набора данных

3. **Выбор метрик оценки:** Для оценки качества наших OCR моделей будем использовать метрику CharAcc, основанную на Edit Distance. Так же будем использовать метрику Fragment Accuracy.

4.2 Разработка и реализация baseline-модели

1. **Выбор базовой архитектуры:** В силу ограниченности вычислительных ресурсов и для удобного проведения экспериментов в качестве базовой модели была

выбрана архитектура Encoder-Decoder проилюстрированная на Рисунке 2. В качестве энкодера и декодера была выбрана архитектура ViT в 4 слоями, размером скрытого состояния равным 256 и размерностью feedforward равным 1024.

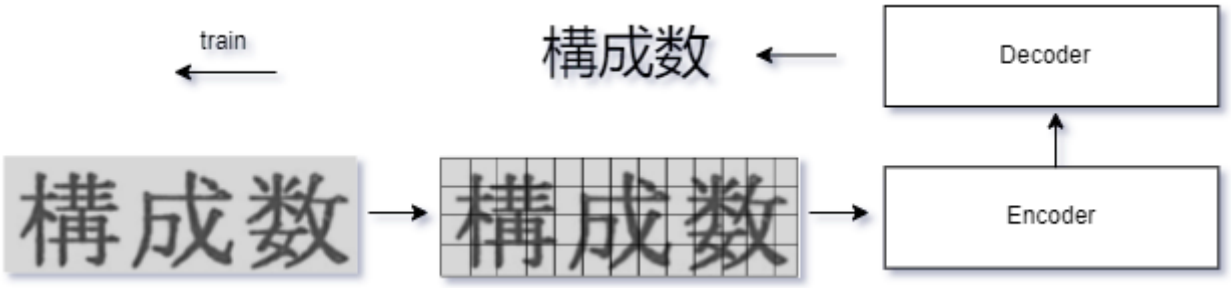

Рис. 2: Архитектура для baseline модели. Изображение делится на патчи размером 16 на 16 пикселей. Затем каждый патч поступает на вход энкодеру. Выход из декодера подается на вход линейному слою, который отображает каждое внутреннее представление декодера в вектор с размерностью алфавита.

2. **Реализация модели:** Код для baseline-модели написан на базе внутреннего фреймворка, предоставленного кафедрой. Фреймворк основывается на таких модулях, как PyTorch и PyTorch Lighting. В качестве функции потерь выбран Connectionist Temporal Classification Loss, так как каждый патч может содержать несколько символов или не содержать их вовсе.

3. **Обучение и оценка baseline-модели:** Обучение модели производилось в течении 80 эпох с размером бача равным 64. Количество батчей данных, которые модель обрабатывает за одну эпоху обучения 20000.

   Значения метрик: CharAcc: 95.8%, FragAcc: 79%.

4.3 Разработка и реализация моделей с маскированием

1. **Маскирование патчей изображения:**

   - Первый метод основан на статье MAE, Рисунок 3. Обучение происходит в 2 этапа. Сначала определенный процент(mask ratio) патчей изображения маскируется(зануляется), их позиции запоминаются, затем видимые патчи подаются на вход энкодеру, на выходе из энкодера добавляются маскированные патчи, затем всё подаётся на вход декодеру. На стадии pretrain модель обучается на задачу реконструкции изображения с помощью MSE Loss, затем на стадии finetune патчи перестают маскироваться и модель обучается с помощью CTC Loss. Для каждой задачи обучается свой декодер.

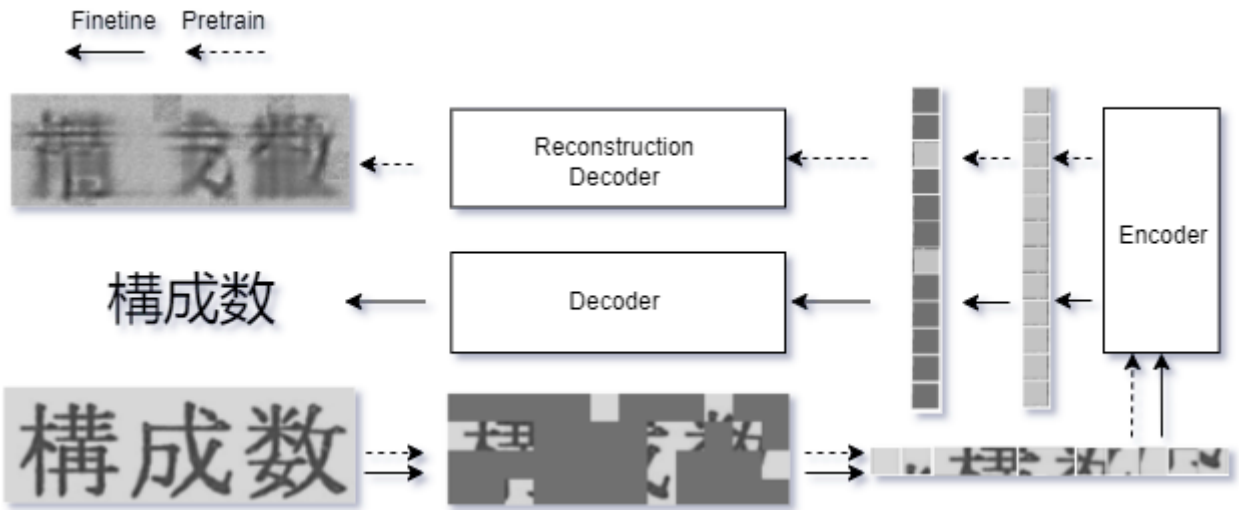

Рис. 3: Архитектура 1

- Второй метод основан на статье SimMIM, Рисунок 4. Отличие от Архитектуры 1 в том, что на вход энкодеру подаются все патчи, независимо от того маскированные они или нет.

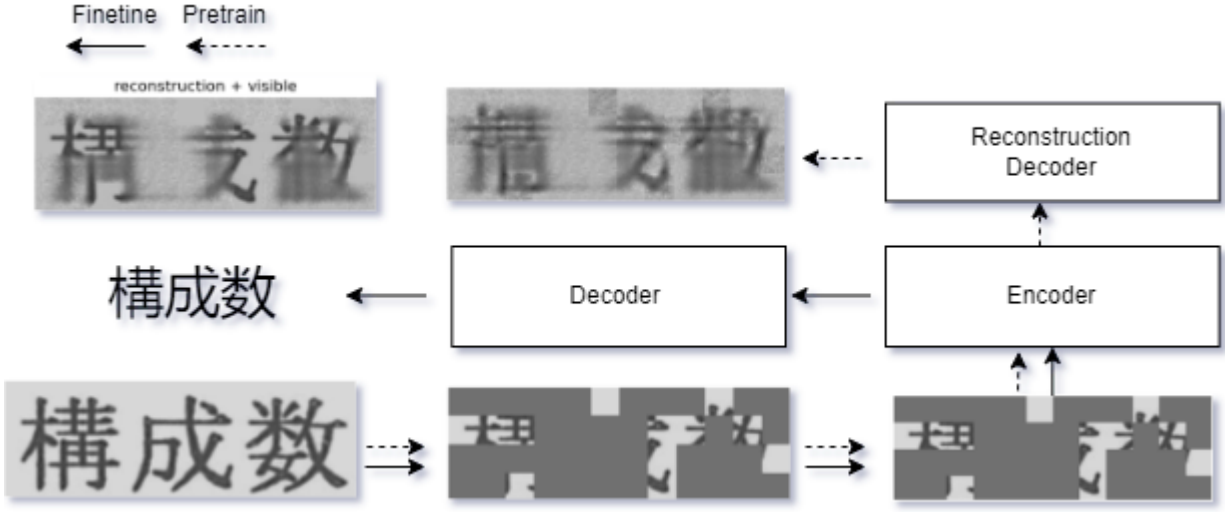

Рис. 4: Архитектура 2

- Третий метод основан на статье SupMAE, Рисунок 5. Отличие от Архитектуры 1 в том, что на стадии pretrain модель одновременно обучается сразу на две функции потерь(CTC Loss и MSE loss).

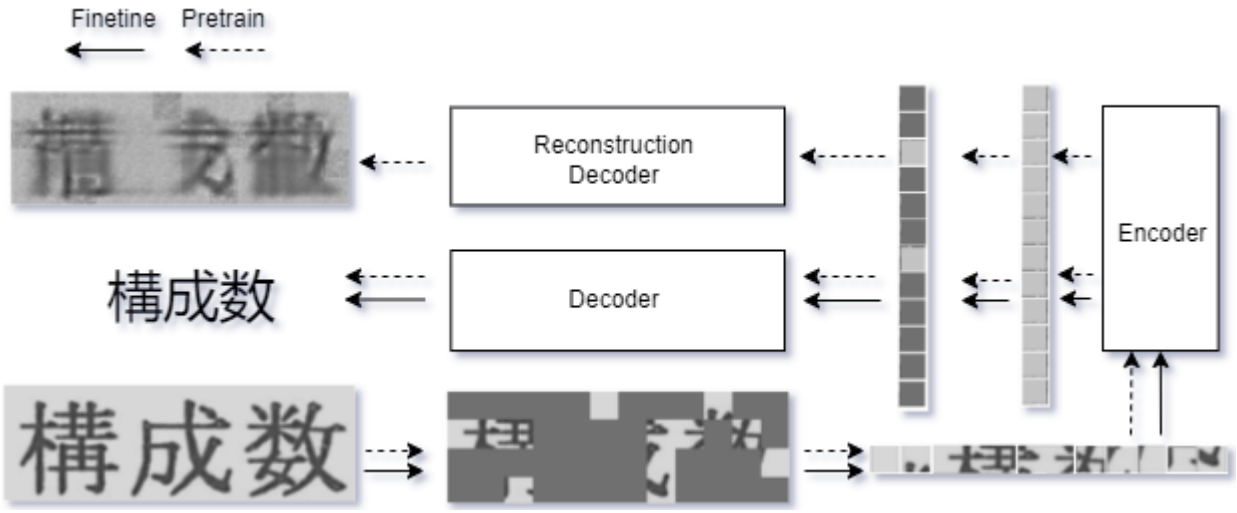

Рис. 5: Архитектура 3

2. **Маскирование входных представлений декодера:**

   - Четвёртый метод основан на идее маскирования входных представлений декодера, Рисунок 6. Обучение происходит в 2 этапа. Сначала Изображение делится на патчи размером 16 на 16 пикселей. Патчи подаются на вход энкодеру, на выходе из энкодера промежуточные представления случайно маскируются(зануляются) с определенным процентом(mask ratio), Всё подаётся на вход декодеру. На стадии pretrain модель обучается с помощью CTC Loss, затем на стадии finetune патчи перестают маскироваться и модель продолжает обучается на задачу распознавания текста. Для каждой задачи обучается свой декодер.

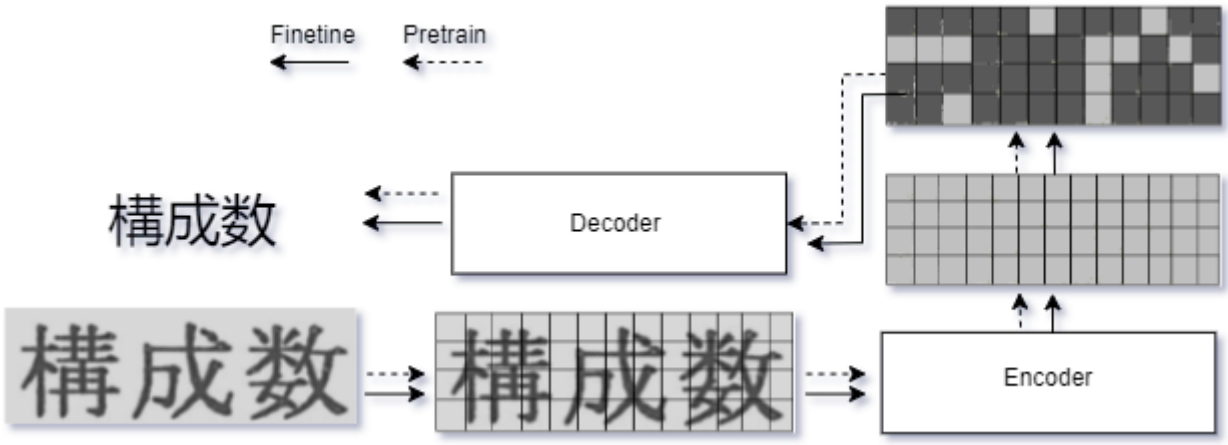

Рис. 6: Архитектура 4

   - В пятом методе, Рисунок 7, в отличии от Архитектуры 4, на стадии pretrain модель одновременно обучается сразу на две функции потерь(CTC Loss и MSE loss).

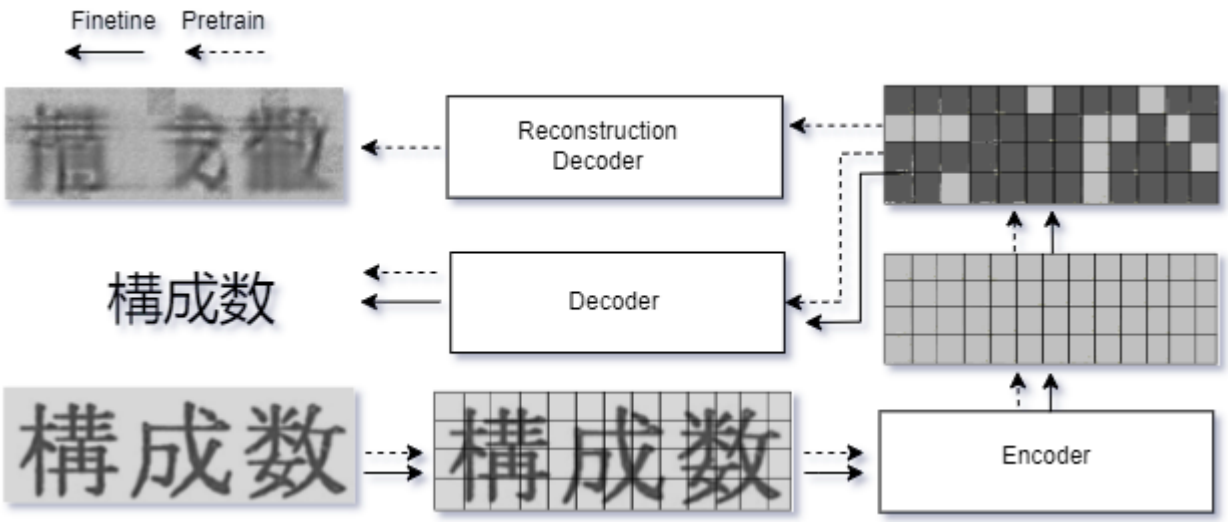

Рис. 7: Архитектура 5

3. **Комбинированное маскирование:**

   - Шестой метод основан на идее совмещения маскирования патчей изображения и входных представлений декодера, Рисунок 8. Обучение по прежнему происходит в 2 этапа. Сначала Изображение делится на патчи размером 16 на 16 пикселей. Определенный процент(mask ratio) патчей изображения маскируется. Патчи подаются на вход энкодеру, на выходе из энкодера промежуточные представления также случайно маскируются с определенным процентом(mask ratio), Всё подаётся на вход декодеру. На обоих стадиях модель обучается с помощью CTC Loss. Стадия finetune отличается от pretrain тем, что mask ratio равен нулю.

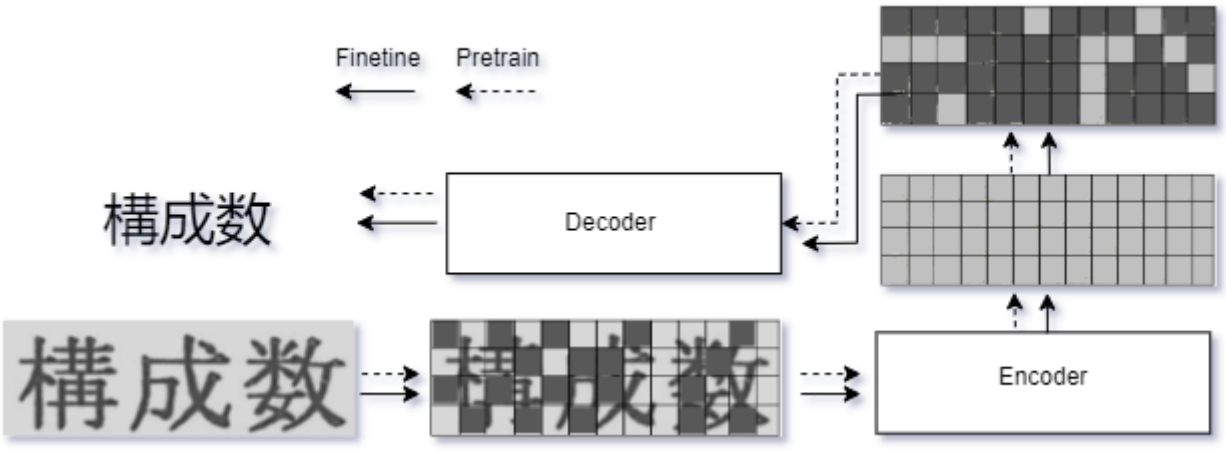

Рис. 8: Архитектура 6

4.4 Проведение экспериментов и анализ результатов

1. Обучение моделей с маскированием:

   - Колличество эпох для стадии pretrain для каждой архитектуры: 60
   - Колличество эпох для стадии finetune для каждой архитектуры: 60
   - Размер маскированного патча: 16x16 px

2. Оценка качества моделей:

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

3. Формулировка выводов:

   - Маскирование патчей изображения позволяет улучшить точность распознавания текста по сравнению с обучением без маскирования. Разные архитектуры моделей с маскированием патчей могут демонстрировать сравнимые результаты.
   - Маскирование входных представлений декодера также эффективно для повышения точности распознавания текста. Данный подход показал себя более эффективным, чем маскирование патчей изображения.
   - Комбинация разных стратегий маскирования позволяет добиться наибольшей точности распознавания текста.

   Полученные результаты свидетельствуют о том, что поставленная задача решена. Удалось разработать и исследовать различные стратегии маскирования для semi - supervised обучения моделей STR на японском языке. Все исследованные стратегии маскирования превзошли baseline модель по точности распознавания текста. Наилуч ший результат был достигнут при комбинировании маскирования патчей изображения и входных представлений декодера.

5 Описание практической части

Описание кода

1. Выбор языка и библиотек

Язык программирования:Python Библиотеки: PyTorch: для создания и обучения нейронных сетей, работы с тензорами, автоматического дифференцирования. OpenCV (cv2):для предобработки изображений (загрузка, изменение размера, нормализация). NumPy:для работы с массивами и математическими операциями. tqdm:для визуализации прогресса обучения.

Мотивы выбора:

Python: популярный язык с большим сообществом, особенно в области машинного обучения. Обладает простым синтаксисом и множеством библиотек. PyTorch: гибкий и мощный фреймворк, позволяющий легко создавать и обучать сложные нейронные сети.

2. Архитектура кода

Код организован в виде модульной структуры, которая включает в себя следующие компоненты:

data loader.py: Загрузка и предобработка датасета, реализация маскирования патчей изображения. models.py: Определение архитектуры модели STR, включая энкодер, декодер и механизмы маскирования. train.py: Функции для обучения и валидации модели, сохранения весов и ведения логов. evaluate.py: Оценка обученной модели на тестовом наборе данных и расчет метрик качества. utils.py: Вспомогательные функции, например, для работы с конфигурационными файлами, визуализации результатов.

3. Схема функционирования

1. Загрузка и предобработка данных: Датасет загружается, изображения предобрабатываются (изменение размера, нормализация), текстовые метки преобразуются в подходящий формат. 2. Создание и обучение модели: Создается экземпляр модели STR с выбранной архитектурой и механизмами маскирования. Модель обучается на обучающем наборе данных с использованием заданных параметров (оптимизатор, функция потерь, количество эпох). 3. Оценка модели: Обученная модель оценивается на тестовом наборе данных для расчета метрик точности распознавания текста (CharAcc).

4. Теоретическая сложность алгоритма

Теоретическая сложность алгоритма обучения нейронной сети зависит от многих факторов, таких как архитектура сети, размер датасета, выбранный оптимизатор и другие параметры. В общем случае, обучение нейронной сети — задача NP-трудная.

5. Характеристики функционирования

Скорость: Скорость работы кода зависит от вычислительной мощности оборудования (CPU, GPU), размера модели и датасета, а также от эффективности реализации. Память: Объем используемой памяти зависит от размера модели, размера батча и разрешения изображений.

6 Заключение

В данной главе представлены результаты экспериментов по исследованию эффективности различных стратегий маскирования в semi-supervised обучении для повышения точности распознавания японского текста на изображениях.

4.1 Результаты маскирования патчей изображения

Архитектура 1.1 (Mask ratio: 25процента, patch size: 16px):CharAcc на тестовом наборе данных составила 96.14процента, что превышает baseline результат (95.8 процента) на 0.34 процента. Архитектура 1.2 (Mask ratio: 75процента, patch size: 16px):Увеличение mask ratio до 75 процента привело к дополнительному улучшению показателя CharAcc до 96.32 процента, что составляет прирост в 0.52 процента относительно baseline. Архитектура 2 (SimMIM, Mask ratio: 75процента, patch size: 16px): Использование архитектуры SimMIM с mask ratio 75 процента показало схожий результат с Архитектурой 1.2 — 96.29 процента CharAcc, что также выше baseline.

Выводы:

Маскирование патчей изображения позволяет улучшить точность распознавания текста по сравнению с обучением без маскирования. Более высокий mask ratio приводит к более значительному повышению точности. Разные архитектуры моделей с маскированием патчей могут демонстрировать сравнимые результаты.

4.2 Результаты маскирования входных представлений декодера

Архитектура 4 (Mask ratio: 75процента):Применение маскирования к входным представлениям декодера с mask ratio 75 процента привело к значительному росту показателя CharAcc до 96.4 процента, что составляет 0.6 процента улучшения относительно baseline.

Выводы:

Маскирование входных представлений декодера также эффективно для повышения точности распознавания текста. Данный подход показал себя более эффективным, чем маскирование патчей изображения.

4.3 Результаты комбинированного маскирования

Архитектура 3 (Masked Vision-Language Transformer, Mask ratio: 25процента): Использование архитектуры, совмещающей в себе механизмы маскирования патчей изображения и входных представлений декодера, привело к CharAcc на уровне 96.34 процента. Архитектура 5 (Mask ratio: 33 процента + 33процента): Комбинация маскирования патчей изображения с mask ratio 33 процента и маскирования входных представлений декодера с тем же mask ratio показала наилучший результат— 96.5 процента CharAcc. Это составляет прирост в 0.7 процента относительно baseline.

Выводы:

Комбинация разных стратегий маскирования позволяет добиться наибольшей точности распознавания текста. Подбор оптимальных параметров маскирования для каждого из подходов играет важную роль.

4.4 Степень решения задачи

Полученные результаты свидетельствуют о том, что поставленная задача решена. Удалось разработать и исследовать различные стратегии маскирования для semi-supervised обучения моделей STR на японском языке. Все исследованные стратегии маскирования превзошли baseline модель по точности распознавания текста. Наилучший результат был достигнут при комбинировании маскирования патчей изображения и входных представлений декодера с определенным соотношением mask ratio.

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

[4] Masked Sequence to Sequence Pre-training for Neural Machine Translation / Kaitao Song, Xu Tan, Tao Qin et al. // *arXiv preprint arXiv:1905.07450*. — 2019.