

Магистерская диссертация

# Генерация рукописного текста на русском языке

Фомин Сергей Александрович

**Научный руководитель:**

Ступников Сергей Александрович

2.06.2023

# Генерация рукописного текста

- Преобразование фрагмента текста в изображение рукописного текста заданным почерком
- На входе ожидается целевой текст для генерации и рукописный образец почерка
  - Изображение с рукописным текстом целевого почерка
  - Отдельные символы рукописного текста целевого почерка
  - Числовой вектор целевого почерка
- На выходе получается сгенерированное заданным почерком изображение целевого текста

Распознавание рукописного текста – это классическая задача информатики, для решения которой в последнее время успешно применяются нейронные сети. Однако, чтобы достичь высокой точности работы, современной нейронной сети требуются не только большие вычислительные ресурсы, но и огромное количество данных для обучения. Один из самых известных способов увеличить объем данных для обучения – аугментация – это генерация дополнительных данных из имеющихся данных. Существуют разные способы



Evaporation of sodium from the pool C is minimised by a close fitting cylindrical block of electrolyte T attached to the +ve nickel conductor by nickel circlips. Fixing and



Evaporation of sodium from the pool C is minimised by a close fitting cylindrical block of electrolyte T attached to the +ve nickel conductor by nickel circlips. Fixing and

# Актуальность

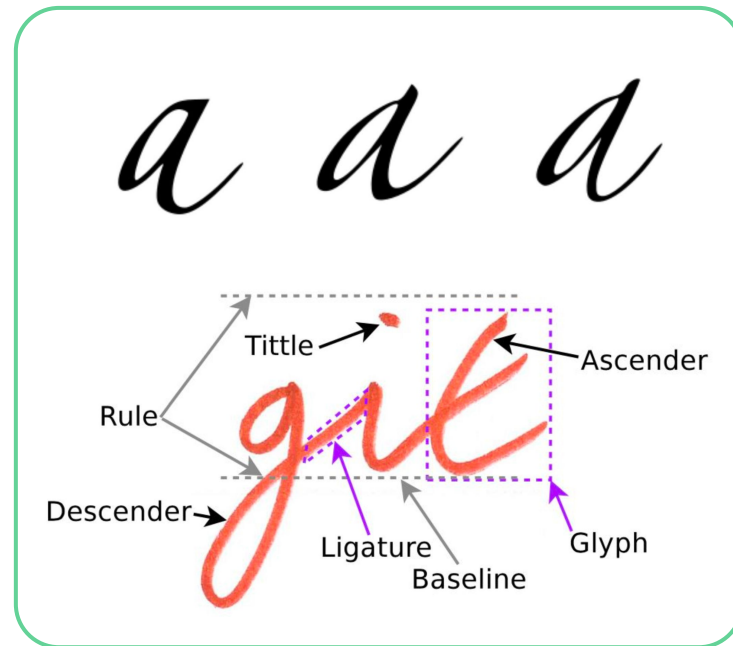
- Сложность нейросетевых алгоритмов увеличивается, интерфейс взаимодействия - облегчается
- Для качественного обучения и достижения высокой точности нейросетевых моделей необходимо собирать большие обучающие наборы данных
- Основную роль в обучении нейронной сети занимает правильная подготовка обучающих данных
- Данные для обучения должны удовлетворять критериям точности, полноты и непротиворечивости
- Аугментация - генерация дополнительных данных из имеющихся данных
- Генерация почерка - один из способов аугментации рукописных наборов текстов для улучшения качества их распознавания нейросетевыми подходами

# Цели и решаемые задачи

- **Разработать подход генерации рукописного текста на русском языке**
- **Показать применимость предложенного подхода в задаче аугментации набора данных для улучшения качества распознавания рукописного текста**
- Исследование существующих подходы генерации рукописного текста на английском языке
- Выделение основных этапов генерации рукописного текста
- Сбор набора данных для генерации рукописного текста на русском языке
- Разработка и программная реализация методов для генерации рукописного текста на русском языке
- Оценка качества предложенных методов генерации рукописного текста на русском языке в задаче распознавания почерка

# Терминологические понятия

- Графема - единица текста
- Глиф - единица графики
- Типы глифов:
  - Обычный (**л, г**)
  - Раздельный (**й, i**)
  - Восходящий (**б, l**)
  - Подстрочный (**р, g**)
- Лигатура - соединительный элемент между глифами



# Родственные работы

- **Haines T. S. F., Mac Aodha O., Brostow G. J. My text in your handwriting (2016)**
- Balreira D. G., Walter M. Handwriting synthesis from public fonts (2017)
- **Blanes A. R. Synthetic handwritten text generation (2018)**
- Aksan E., Pece F., Hilliges O. Deepwriting: Making digital ink editable via deep generative modeling (2018)
- Ji B., Chen T. Generative adversarial network for handwritten text (2019).
- Fogel S. et al. ScrabbleGAN: Semi-supervised varying length handwritten text generation (2020)
- **Kotani A., Tellex S., Tompkin J. Generating handwriting via decoupled style descriptors (2020)**

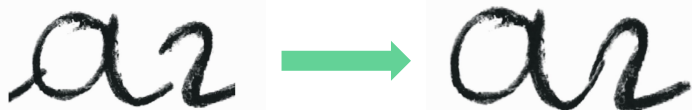
# Основные этапы генерации рукописного текста

- Генерация глифов данного образца почерка



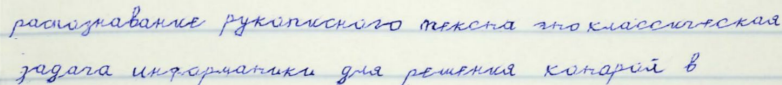
аааааа аааааа

- Генерация лигатур для сгенерированных глифов



а я → ая

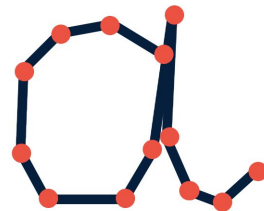
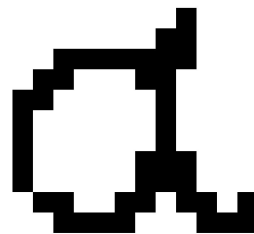
- Финальная обработка рукописного текста



распознавание рукописного текста это классическая  
задача информатики для решения которой в

# Предлагаемые подходы генерации рукописного текста

- Растровый подход
  - Глифы представляются в виде растровых изображений
  - Хорошо поддаются обработке сверточными нейронными сетями
- Векторный подход
  - Глифы представляются в виде векторных изображений
  - Более естественное представление глифов





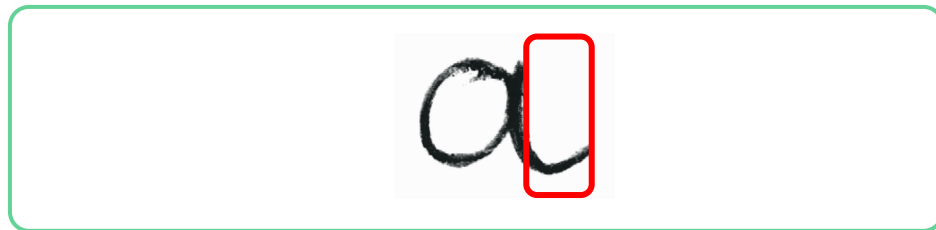
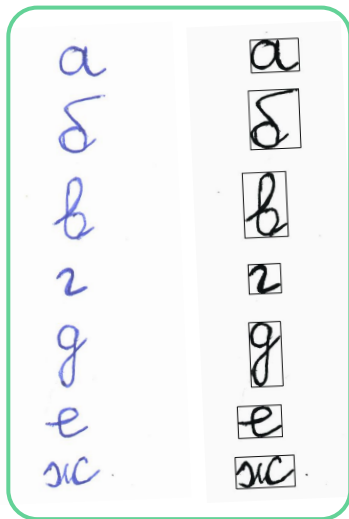
# Растровый подход генерации рукописного текста

- Генерация глифов
  - Перенос стиля изображения (Style Transfer)
  - Генерация глифов на основе архитектуры UNet
  - Генерация глифов на основе вариационного автокодировщика (VAE)
  - Генерация глифов на основе генеративно-состязательной нейросети (GAN)
- Генерация лигатур
  - Эвристический алгоритм генерации соединительных элементов
- Финальная обработка
  - Наложение цвета и текстуры
  - Добавление фонового изображения
  - Деформация и наклон

```
def generate(text):  
    glyphs = []  
    for char in text:  
        glyph = generate_glyph(char)  
        glyphs.append(glyph)  
  
    words = []  
    word = None  
    for i in range(len(text)):  
        if text[i] == " ":  
            words.append(word)  
            word = None  
            word = generate_ligature(  
                word,  
                glyphs[i],  
                text[i-1],  
                text[i],  
            )  
        words.append(word)  
  
    img = generate_handwriting(words)  
    img = process_ink(img)  
    img = process_background(img)  
    img = process_tilt(img)  
    img = process_deformations(img)  
    return img
```

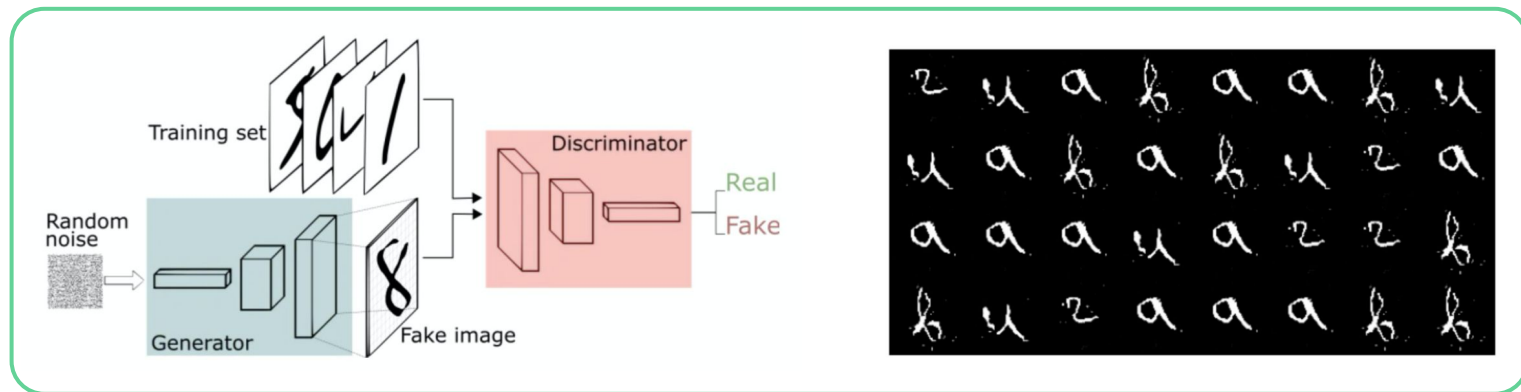
# Генерация глифов: набор данных

- Набор данных из отдельных глифов русского алфавита
- Количество: 89 элементов по 10 экземпляров
- Разметка: тип глифа, возможность правого и левого соединения



# Генерация глифов: генеративно-сопоставительная нейронная сеть (GAN)

- Высокое качество генерации по сравнению с другими генерирующими подходами
- Эксперименты с архитектурой сверточной нейронной сети, гиперпараметрами, функцией потерь
- Аугментация и повышение разнообразия генерации



# Генерация лигатур

- Для каждой графемы осуществляется разметка на возможность правого и левого соединений
- Глифы генерируются с наличием этих соединений
- Изображение глифа со стороны соединения попиксельно сдвигается вверх или вниз для образования необходимой лигатуры



```
def generate(l_img, r_img, l_char, r_char):  
    is_l_ligature = has_right_ligature(l_char)  
    is_r_ligature = has_left_ligature(r_char)  
  
    left_start = find_start_right_side(l_img)  
    right_start = find_start_left_side(r_img)  
  
    shift_distance = right_start - left_start  
  
    if is_l_ligature and is_r_ligature:  
        shift_right_side(l_img, shift_distance / 2)  
        shift_left_side(r_img, shift_distance / 2)  
    else if is_l_ligature:  
        shift_right_side(l_img, shift_distance)  
    else:  
        shift_left_side(r_img, shift_distance)  
  
    return concatenate(l_img, r_img)  
  
def shift_right_side(img, distance):  
    alpha = calculate_alpha(distance)  
    if distance >= 0:  
        quadratic_shift_up(img, alpha)  
    else:  
        quadratic_shift_down(img, alpha)
```

# Финальная обработка

- Генерация рукописного текста на основе растровых подходов генерации глифов и лигатур

распознавание рукописного текста это классическая задача информатики для решения которой в последнее время успешно применяются нейронные сети

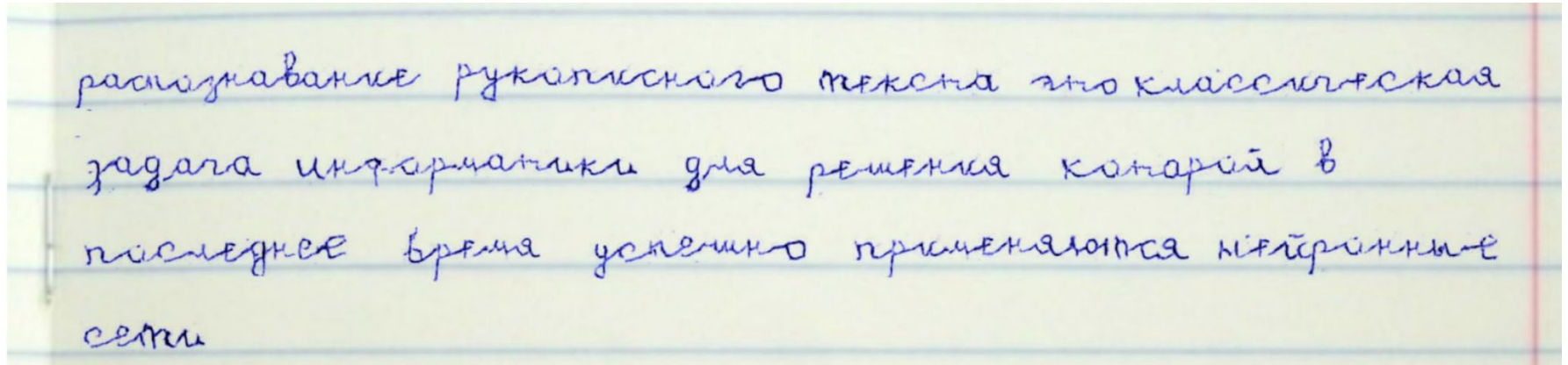
# Финальная обработка

- Наложение цвета и неравномерность затемнения (имитация чернил)

распознавание рукописного текста это классическая задача информатики для решения которой в последнее время успешно применяются нейронные сети

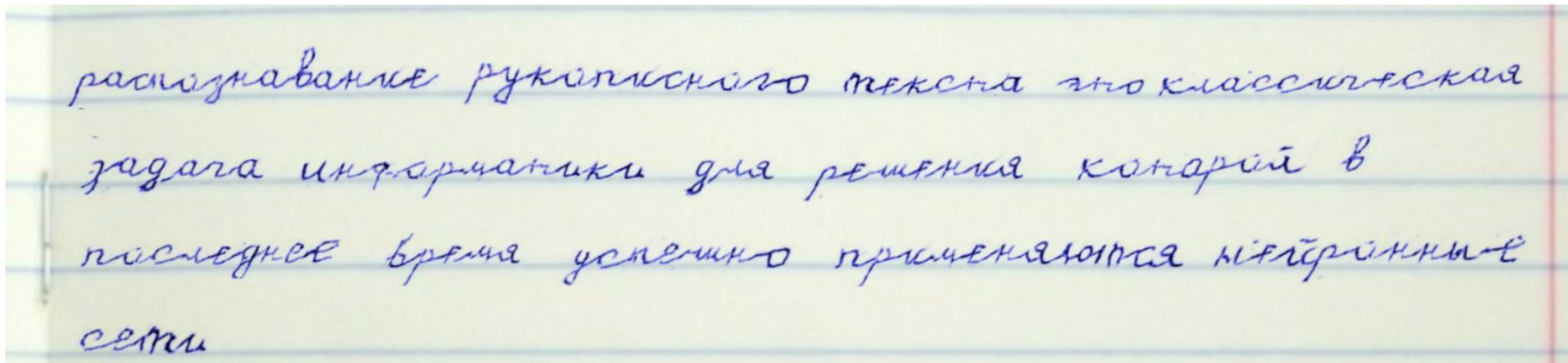
# Финальная обработка

- Добавление фонового изображения



# Финальная обработка

- Деформация и наклон текста



распознавание рукописного текста это классическая  
задача информатики для решения которой в  
последнее время успешно применяются нейронные  
сети



# Векторный подход генерации рукописного текста

- Генерация глифов
  - Нейросетевая модель представления графемы
  - Нейросетевая модель векторного кодировщика-декодировщика
- Генерация лигатур
  - Эвристический алгоритм генерации соединительных элементов
- Финальная обработка изображения
  - Наложение цвета и текстуры
  - Добавление фонового изображения
  - Деформация и наклон

```
def generate(text):
    words = []
    for word in text.split(" "):
        glyphs = generate_glyphs(word)
        words.append(glyphs)

    words_with_l = []
    for word in words:
        word_with_l = None
        for glyph in word:
            if use_ligatures(word_with_l, glyph):
                word_with_l = gen_ligature(
                    word_with_l, glyph
                )
            else:
                word_with_l = concatenate(
                    word_with_l, glyph
                )
        words_with_l.append(word_with_l)

    img = generate_handwriting(words_with_l)
    img = process_ink(img)
    img = process_background(img)
    img = process_tilt(img)
    img = process_deformations(img)
    return img
```

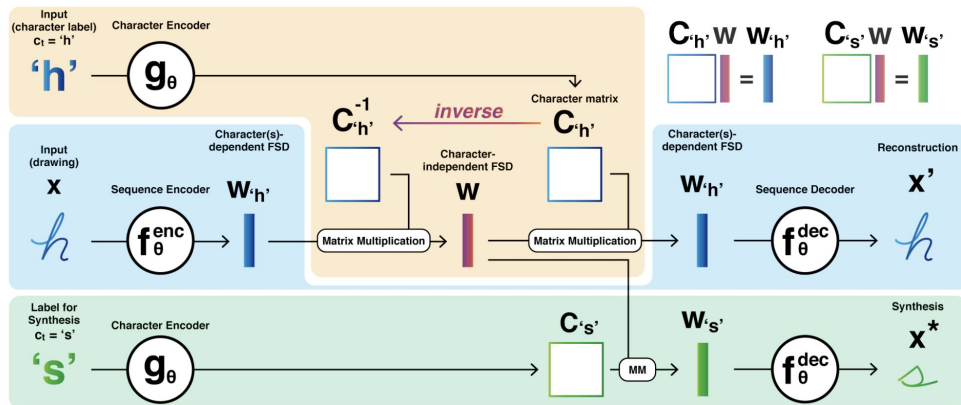
# Генерация глифов: набор данных

- Программа автоматического сбора и обработки векторного представления рукописного текста
- Набор данных из векторных представлений слов и словосочетаний целевого почерка
- Размер набора данных: 100 слов и словосочетаний без повторений



# Генерация глифов: идея векторного подхода

- Сверточная нейронная сеть для преобразования графемы в квадратную матрицу внутреннего представления этой графемы
- Рекуррентная нейронная сеть для преобразования векторного изображения глифа в числовой вектор фиксированной длины и наоборот
- Вектор почерка - внутреннее представление почерка в виде числового вектора



# Генерация глифов: результат

- Дообучение нейросетевой пары на собранном наборе данных
- Разнообразие сгенерированных глифов одной и той же графемы
- Генерация глифов с наличием естественных лигатур

аааааа а аааа

приветствуем участников и гостей соревнований

# Генерация лигатур

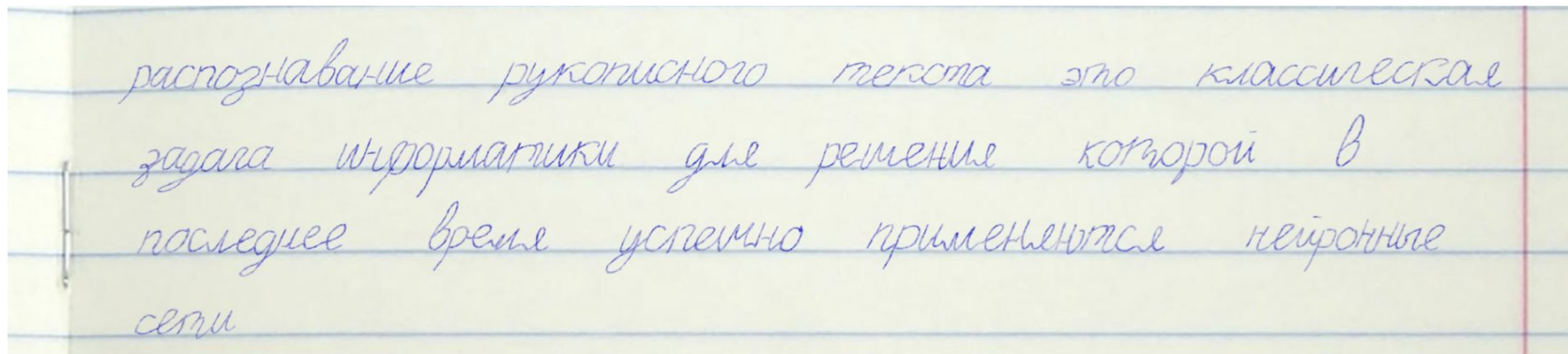
- Преобразование векторного изображения в растровое
- Модификация растрового алгоритма генерации лигатур - пропуск неестественных соединений за счет дополнительной разметки соединений



машинистратура

# Финальная обработка

- Генерация рукописного текста на основе векторных подходов генерации глифов и лигатур
- Наложение цвета и неравномерность затемнения (имитация чернил)
- Добавление фонового изображения
- Деформация и наклон текста

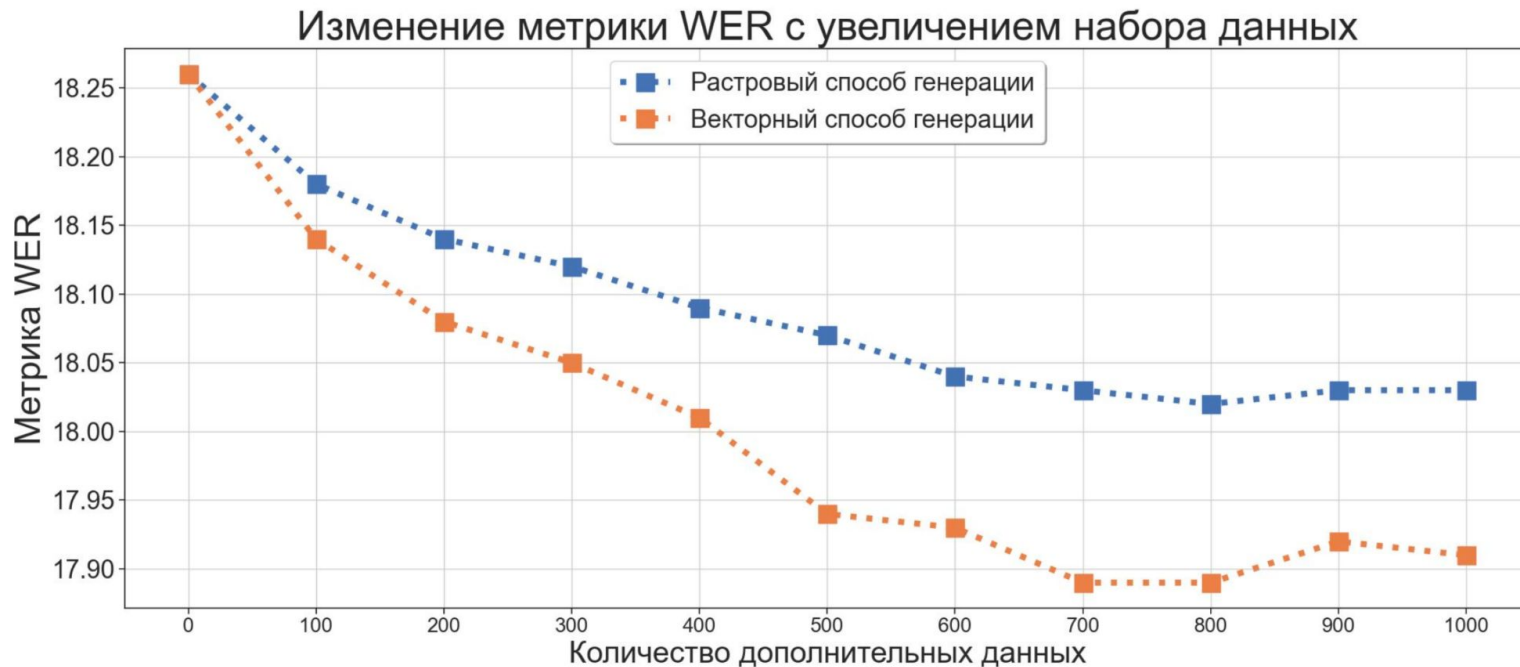


# Оценка качества: метрика WER

- Нейросетевая модели распознавания почерка AttentionHTR
- Открытый набор данных русских рукописных текстов HKR
- Генерация трех различных почерков разной степени сложности

	Почерк 1	Почерк 2	Почерк 3
Способ расширения набора данных			
Базовая аугментация	16.54	18.23	27.98
Растровый подход генерации	16.47	18.03	<b>26.96</b>
Векторный подход генерации	<b>16.44</b>	<b>17.91</b>	27.60

# Оценка качества: размер обучающего набора





# Результаты работы

- Исследованы существующие подходы генерации рукописного текста на английском языке
- Выделены основные этапы генерации почерка: генерация глифов, генерация лигатур, финальная обработка
- Собрано и обработано несколько наборов данных разных форматов
- Разработано и программно реализовано два полноценных подхода генерации рукописного текста на русском языке: растровый и векторный. В том числе разработан эвристический алгоритм генерации лигатур и финальной обработки
- Произведена оценка качества предложенных подходов в задаче распознавания почерка и доказана применимость каждого из них