Оглавление

[**Список сокращений** 2](#_Toc183037638)

[**1. Введение** 3](#_Toc183037639)

[**2. Передача стиля без нейронных сетей** 5](#_Toc183037640)

[**3. Истоки нейронной передачи стиля** 6](#_Toc183037641)

[**4. Таксономия алгоритмов передачи стиля** 8](#_Toc183037642)

[**5. Генеративные состязательные сети** 13](#_Toc183037643)

[**Список литературы** 14](#_Toc183037644)

# **Список сокращений**

**NPR** - Nonphotorealistic Rendering

**CNN** - Convolutional Neural Network

**NST** - Neural Style Transfer

**IB – AR** - Image Based Artistic Rendering

**SBR** – Stroke Based Rendering

**MRF** - Markov Random Fields

**IOB – IR** - Image Reconstruction based on Online Image Optimisation

**MOB – IR** - Image Reconstruction based on Offline Model Optimisation

**IOB – NST** - Neural Method Based on Online Image Optimisation

**MOB – NST** - Neural Method Based on Offline Model Optimisation

**GAN** - Generative Adversarial Network

**BN** – Batch Normalization

**IN** - Instance Normalisation

**PSPM** - Per-Style-Per-Model

**MSPM** - Multiple-Style-Per-Model

**ASPM** - Arbitrary-Style-Per-Model

**CIN** - Conditional Instance Normalisation

# 

# **1. Введение**

В настоящее время автоматизация управления и мониторинга технологических процессов играет важную роль в промышленности, энергетике и других отраслях. Использование программируемых реле (ПР) становится всё более популярным благодаря их гибкости, экономичности и возможности интеграции в сложные системы. Для разработки алгоритмов работы таких устройств разработаны специализированные среды программирования, одной из которых является **OWEN Logic**.

**OWEN Logic** — это специализированная среда программирования, предназначенная для создания алгоритмов управления программируемыми реле и панелями, производимыми компанией «ОВЕН». Она позволяет разрабатывать и отлаживать проекты автоматизации в удобной графической форме, используя стандарты IEC 61131-3. С помощью OWEN Logic возможно создавать как простые, так и сложные алгоритмы управления, адаптированные под конкретные задачи автоматизации.

Актуальность данного дипломного проекта обусловлена необходимостью упрощения разработки и улучшения качества программ для управления технологическими процессами. OWEN Logic предоставляет все необходимые инструменты для проектирования таких решений, однако важно тщательно проработать архитектуру алгоритмов, организовать эффективное взаимодействие с периферийным оборудованием и учесть специфику управления в реальных условиях.

Целью данного дипломного проекта является разработка информационной системы для автоматизации управления технологическими процессами на базе OWEN Logic.

Для достижения этой цели необходимо решить следующие задачи:

1. Изучить возможности и функционал среды программирования OWEN Logic.
2. Исследовать особенности программируемых реле и панелей серии ПР, производимых компанией «ОВЕН».
3. Спроектировать и разработать алгоритм управления процессом с учетом требований предметной области.
4. Реализовать взаимодействие устройства с периферийными модулями.
5. Провести отладку и тестирование системы в режиме симуляции и на реальном оборудовании.
6. Оценить эффективность разработанного решения.

### 1.1 Описание предметной области

Современные промышленные процессы требуют высокой степени автоматизации, чтобы обеспечить повышение эффективности, снижение затрат и улучшение качества выпускаемой продукции. Автоматизация включает в себя управление различными технологическими процессами, мониторинг работы оборудования и интеграцию с внешними системами.

Ключевую роль в реализации таких систем играют программируемые реле (ПР), которые обеспечивают управление и контроль за процессами с минимальным участием человека. ПР активно используются в самых разных областях, таких как:

* Производство (управление линиями, упаковочным и фасовочным оборудованием);
* Энергетика (мониторинг работы генераторов, распределительных щитов и трансформаторов);
* ЖКХ (управление насосными станциями, системами отопления и вентиляции);
* Умные дома (автоматизация освещения, управления климатом, безопасностью).

В этих условиях программируемые реле обеспечивают компактность, надёжность и экономичность решений. Их применение требует создания алгоритмов, которые позволяют реле выполнять задачи управления и обмениваться данными с внешними системами.

Для разработки таких алгоритмов используется **OWEN Logic** — программная среда, ориентированная на разработку программ для устройств компании «ОВЕН». Эта среда программирования предоставляет следующие возможности:

* Создание алгоритмов управления на графическом языке FBD (Function Block Diagram), соответствующем стандарту МЭК 61131-3;
* Визуальная настройка параметров работы устройств;
* Моделирование процессов в симуляторе для проверки работоспособности алгоритма до его загрузки в устройство;
* Интеграция с внешними системами через протоколы Modbus RTU, Modbus TCP и другие.

### 1.2 Сравнительный анализ программ-аналогов

На рынке существует множество средств программирования и автоматизации, однако OWEN Logic выгодно выделяется благодаря своей узкой специализации под оборудование компании «ОВЕН» и ориентированности на решения для отечественного рынка.

Для сравнения рассмотрим две другие среды программирования:

1. **Codesys**
   * Используется для программирования ПЛК различных производителей.
   * Обеспечивает работу с множеством языков программирования стандарта IEC 61131-3, включая FBD, LD, ST.
   * Поддерживает сложные функции: ПИД-регуляторы, программирование визуализации и многое другое.  
     **Недостатки**: сложность освоения для начинающих, отсутствие интеграции с конкретным оборудованием компании «ОВЕН».
2. **Tia Portal (Siemens)**
   * Предназначена для работы с оборудованием Siemens, включая ПЛК серии S7.
   * Поддерживает интеграцию с системой SCADA.
   * Обеспечивает высокую гибкость при проектировании и отладке.  
     **Недостатки**: высокая стоимость, отсутствие интеграции с оборудованием других производителей.

**OWEN Logic** же является более доступным и специализированным решением для оборудования компании «ОВЕН». Она обеспечивает полный цикл разработки, от проектирования алгоритма до записи программы в устройство, а также предоставляет возможность симуляции и тестирования.

**Преимущества OWEN Logic**:

* Простота использования даже для разработчиков с базовыми навыками;
* Узкая специализация под оборудование компании «ОВЕН»;
* Возможность быстрого создания прототипов и внедрения изменений в реальное производство.

**Недостатки**:

* Ограниченная совместимость с оборудованием других производителей;
* Требуется детальное знание возможностей программируемых реле для создания сложных проектов.

# **2. Передача стиля без нейронных сетей**

Художественная стилизация — давняя тема исследований. Благодаря широкому спектру приложений она была важной областью исследований на протяжении более двух десятилетий. До появления NST соответствующие исследования расширились в область, называемую нефотореалистичным рендерингом (NPR). В этом разделе кратко рассмотрим некоторые из этих алгоритмов художественного рендеринга без CNN. В частности, сосредоточимся на художественной стилизации 2D-изображений, которая в [14] называется художественным рендерингом на основе изображений (IB-AR).

**2.1.1 Рендеринг на основе штрихов**

Относится к процессу размещения виртуальных штрихов (например, мазков кисти, плиток, точек) на цифровом холсте для рендеринга фотографии в определенном стиле [16]. Процесс SBR обычно начинается с исходной фотографии, постепенно компонуя штрихи для соответствия фотографии и, наконец, создавая нефотореалистичное изображение, которое выглядит как фотография, но с художественным стилем. В ходе этого процесса разрабатывается целевая функция для управления жадным или итеративным размещением штрихов. Цель алгоритмов SBR — точно отобразить предписанный стиль. Поэтому они, как правило, эффективны при имитации определенных типов стилей (например, масляной живописи, акварели, эскизов). Однако каждый алгоритм SBR тщательно разработан только для одного конкретного стиля и не способен имитировать произвольный стиль.

**2.1.2 Методы на основе областей**

Региональный рендеринг должен включать сегментацию регионов, чтобы обеспечить адаптацию рендеринга на основе содержимого в них. Ранние алгоритмы IB-AR на основе регионов используют их форму для управления размещением штрихов [17], [18]. Таким образом, различные шаблоны штрихов могут быть созданы в различных семантических регионах изображения. Сонг и др. [19] далее предлагают региональный алгоритм IB-AR для управления геометрией для художественных стилей. Их алгоритм создает упрощенные эффекты рендеринга форм, заменяя регионы несколькими каноническими формами. Учет регионов при рендеринге позволяет локально контролировать уровень детализации. Однако проблема в SBR сохраняется: один региональный алгоритм рендеринга не способен имитировать произвольный стиль.

**2.1.3 Рендеринг на основе примеров.**

Цель рендеринга на основе примеров - изучить сопоставление между парой образцов. Эта категория методов IB-AR была впервые предложена Герцманом и др., которые предложили структуру, называемую аналогиями изображений [9]. Аналогии изображений направлены на изучение сопоставления между парой исходных изображений и целевыми стилизованными изображениями под наблюдением. Обучающий набор аналогии изображений состоит из пар нестилизованных исходных изображений и соответствующих стилизованных изображений с определенным стилем. Затем алгоритм аналогии изображений изучает аналогичное преобразование из примеров обучающих пар и создает аналогичные стилизованные результаты при задании тестовой входной фотографии. Аналогию изображений также можно расширить различными способами, например, для изучения размещения штрихов для рендеринга портретной живописи [20]. В целом, аналогии изображений эффективны для различных художественных стилей. Однако пары обучающих данных обычно недоступны на практике. Другим ограничением является то, что аналогии изображений используют только низкоуровневые характеристики изображения. Поэтому они обычно не могут эффективно улавливать содержание и стиль, что ограничивает производительность.

**2.2 Обработка и фильтрация изображений.**

Создание художественного изображения — это процесс, направленный на упрощение и абстракцию изображения. Поэтому, естественно, рассмотреть возможность принятия и комбинирования некоторых связанных фильтров обработки изображений для рендеринга данной фотографии. Например, в [21] Виннемоллер и др. впервые используют двусторонние [22] и разностные фильтры Гаусса [23] для автоматического создания эффектов, похожих на мультфильмы. По сравнению с другими категориями методов IB-AR, алгоритмы рендеринга, основанные на фильтрации изображений, обычно просты в реализации и эффективны на практике. За счет их стилевого разнообразия они очень ограничены.

# **3. Истоки нейронной передачи стиля**

Для автоматического переноса художественного стиля первым и самым важным вопросом является то, как моделировать и извлекать стиль из изображения. Поскольку стиль тесно связан с текстурой, простой способ – это связать визуальное моделирование стиля с ранее хорошо изученными методами визуального моделирования текстуры. После получения представления стиля, следующим вопросом является то, как реконструировать изображение с желаемой информацией о стиле, сохраняя его содержание, что решается методами реконструкции изображения.

**3.1 Визуальное моделирование текстур**

Визуальное моделирование текстур [24] ранее изучалось как сердце синтеза текстур [25], [26]. На протяжении всей истории существуют два различных подхода к моделированию визуальных текстур, которые являются параметрическим моделированием текстур с суммарной статистикой и непараметрическим моделированием текстур с марковскими случайными полями (MRF).

**3.1.1 Параметрическое моделирование текстур с суммарной статистикой**.

Один из путей к моделированию текстур — это сбор статистики изображения из образца текстуры и использование суммарного статистического свойства для моделирования текстуры. Идея впервые предложена Юлешем [27], который моделирует текстуры как пиксельную статистику N-го порядка. Позже работа в [28] использует ответы фильтра для анализа текстур вместо прямых пиксельных измерений. После этого Портилла и Симончелли [29] вводят текстурную модель, основанную на многомасштабных ориентированных откликах фильтра и используют градиентный спуск для улучшения синтезированных результатов. Более поздний параметрический подход к моделированию текстуры, предложенный Гатисом и др. [30], является первым для измерения сводной статистики в области CNN. Они разрабатывают представление на основе Грама для моделирования текстур, которое представляет собой корреляции между откликами фильтра в различных слоях предварительно обученной сети классификации (сети VGG) [31]. Более конкретно, представление на основе Грама кодирует статистику второго порядка набора откликов фильтра CNN. Далее я подробно объясню это представление для использования в следующих разделах.

**3.1.2 Непараметрическое моделирование текстур с помощью MRF**.

Другой заметной методологией моделирования текстур является использование непараметрической повторной выборки. Различные непараметрические методы основаны на модели MRF, которая предполагает, что в текстурном изображении каждый пиксель полностью характеризуется своим пространственным соседством. При этом предположении Эфрос и Люнг [25] предлагают синтезировать каждый пиксель по одному путем поиска похожих соседств в исходном текстурном изображении и назначения соответствующего пикселя. Их работа является одним из самых ранних непараметрических алгоритмов с MRF. После их работы Вэй и Левой [26] еще больше ускорили процесс сопоставления соседств, всегда используя фиксированное соседство.

**3.2 Реконструкция изображения**

В целом, важным шагом для многих задач зрения является извлечение абстрактного представления из входного изображения. Реконструкция изображения — это обратный процесс, который заключается в реконструкции всего входного изображения из извлеченного представления. Ранее изучалось, как проанализировать конкретное представление изображения и выяснить, какая информация содержится в абстрактном представлении. Здесь наше основное внимание уделяется алгоритмам реконструкции изображений на основе представления CNN, которые можно разделить на онлайн-реконструкцию изображений на основе оптимизации изображений (IOB-IR) и офлайн-реконструкцию изображений на основе оптимизации моделей (MOB-IR).

**3.2.1 Онлайн-реконструкция изображений на основе оптимизации изображений**.

Первый алгоритм для обратного представления представлений CNN предложен Махендраном и Ведалди [33, 34]. Их алгоритм итеративно оптимизирует изображение (обычно начиная со случайного шума), пока оно не получит аналогичное желаемое представление CNN. Процесс итеративной оптимизации основан на градиентном спуске в пространстве изображений. Поэтому процесс отнимает много времени, особенно когда желаемое восстановленное изображение большое.

**3.2.2 Реконструкция изображений в автономном режиме на основе оптимизации модели.**

Чтобы решить проблему эффективности [33, 34], Досовицкий и Брокс [35] предлагают обучать сеть прямого распространения заранее и возлагать вычислительную нагрузку на стадию обучения. На стадии тестирования обратный процесс можно просто выполнить с помощью прямого прохода сети. Их алгоритм значительно ускоряет процесс реконструкции изображения. В своей более поздней работе [36] они дополнительно объединяют Генеративную состязательную сеть (GAN) [37] для улучшения результатов.

# **4. Таксономия алгоритмов передачи стиля**

NST является подмножеством вышеупомянутых методов IB-AR, основанных на примерах. В этом разделе дадим классификацию алгоритмов NST, а затем подробно рассмотрим основные нефотореалистичные алгоритмы NST на основе 2D-изображений. Конкретнее, для каждого алгоритма начнем с представления основной идеи, а затем обсудим его слабые и сильные стороны. Поскольку сложно определить понятие стиля [3, 38] и, следовательно, очень субъективно определить, какие критерии важны для создания успешного алгоритма передачи стиля [39], попытаемся оценить эти алгоритмы более структурно, сосредоточившись только на деталях, семантике, глубине и вариациях мазков кисти. Текущие методы NST попадают в одну из двух категорий: онлайн-нейронные методы на основе оптимизации изображений (IOB-NST) и офлайн-нейронные методы на основе оптимизации моделей (MOB-NST). Первая категория передает стиль путем итеративной оптимизации изображения, т. е. алгоритмы, принадлежащие к этой категории, построены на методах IOB-IR. Вторая категория оптимизирует генеративную модель в автономном режиме и создает стилизованное изображение за один прямой проход, который использует идею методов MOB-IR.

**4.1 Онлайн нейронные методы на основе оптимизации изображений**

DeepDream [40] — первая попытка создания художественных изображений путем обращения представлений CNN с помощью методов IOB-IR. Дальнейшее объединение методов визуального моделирования текстур для моделирования стиля, впоследствии предлагаются алгоритмы IOB-NST, которые закладывают ранние основы для области NST. Их основная идея заключается в том, чтобы сначала моделировать и извлекать информацию о стиле и содержании из соответствующих изображений стиля и содержания, повторно объединять их в качестве целевого представления, а затем итеративно реконструировать стилизованный результат, который соответствует целевому представлению. В целом, различные алгоритмы IOB-NST используют одну и ту же технику IOB-IR, но различаются способом моделирования визуального стиля, который построен на вышеупомянутых двух категориях методов визуального моделирования текстур. Общим ограничением алгоритмов IOB-NST является то, что они вычислительно затратны из-за итеративной процедуры оптимизации изображения.

**4.1.1 Параметрические нейронные методы с суммарной статистикой**

Первое подмножество методов IOB-NST основано на параметрическом моделировании текстуры с суммарной статистикой. Стиль характеризуется как набор пространственных суммарных статистик. Начнем с представления первого алгоритма NST, предложенного Гатисом и др. [4, 10]. Реконструируя представления из промежуточных слоев сети VGG-19, Гатис и др. наблюдают, что глубокая сверточная нейронная сеть способна извлекать содержимое изображения из произвольной фотографии и некоторую информацию о внешнем виде из известного произведения искусства. Согласно этому наблюдению, они создают компонент содержимого нового стилизованного изображения, штрафуя разницу высокоуровневых представлений, полученных из содержимого и стилизованных изображений, и далее строят компонент стиля, сопоставляя суммарные статистики стиля и стилизованных изображений на основе Грама, которые выводятся из предложенной ими техники моделирования текстуры [30] (раздел 3.1)

**4.1.2 Непараметрические нейронные методы с MRF**

Непараметрический IOB-NST построен на основе непараметрического моделирования текстур с MRF. Эта категория рассматривает NST на локальном уровне, т. е. работает с патчами, чтобысоответствовать стилю.Ли и Ванд [41] первыми предложили алгоритм NST на основе MRF. Они обнаружили, что параметрический метод NSTсо сводной статистикой фиксирует только попиксельные корреляции признаков и не ограничивает пространственнуюкомпоновку, что приводит к менее визуально правдоподобному результату дляфотореалистичных стилей. Их решение состоит в том, чтобы смоделировать стильнепараметрическим способом и ввести новую функцию потери стиля,которая включает в себя априорную MRF.

**4.2 Офлайновые нейронные методы на основе оптимизации модели**

Хотя IOB-NST способен выдавать впечатляющие стилизованные изображения, все еще существуют некоторые ограничения. Наиболее беспокоящим ограничением является проблема эффективности. Вторая категория MOB-NST решает проблему скорости и вычислительных затрат.

**4.2.1 Методы c одним стилем на модель**

**Параметрический PSPM с суммарной статистикой.**

Первые два алгоритма MOB-NST предложены Джонсоном и др. [42] и Ульяновым и др. [43] соответственно. Эти два метода разделяют схожую идею, которая заключается в предварительном обучении сети прямого распространения специфического стиля и получении стилизованного результата с одним прямым проходом на этапе тестирования. Они отличаются только архитектурой сети, для которой дизайн Джонсона и др. примерно следует сети, предложенной Рэдфордом и др. [44], но с остаточными блоками, а также дробно-шаговыми свертками, а Ульянов и др. используют многомасштабную архитектуру в качестве сети-генератора. Целевая функция похожа на алгоритм Гатиса и др. [4], что указывает на то, что они также являются параметрическими методами с суммарной статистикой. Алгоритмы Джонсона и др. и Ульянова и др. достигли передачи стиля в реальном времени. Однако их алгоритм в основном следует алгоритму Гатиса и др. [4], что заставляет их встретиться с теми же вышеупомянутыми проблемами, что и алгоритм Гатиса и др. [4] (например, отсутствие учета согласованности деталей и информации о глубине). Вскоре после [42, 43] Ульянов и др. [45] далее обнаружили, что простое применение нормализации к каждому отдельному изображению, а не к пакету изображений (точнее пакетная нормализация (BN)) приводит к значительному улучшению качества стилизации. Эта нормализация отдельного изображения называется нормализацией экземпляра (IN), которая эквивалентна пакетной нормализации, когда размер пакета установлен равным 1. Показано, что сеть передачи стиля с IN сходится быстрее, чем BN, а также дает визуально лучшие результаты. Одна из интерпретаций заключается в том, что IN является формой нормализации стиля и может напрямую нормализовать стиль каждого изображения до желаемого [46].

**Непараметрический PSPM с MRF.**

Другая работа Ли и Ванда [47] вдохновлена ​​алгоритмом NST [41] на основе MRF в разделе 4.1.2. Они решают проблему эффективности, обучая марковскую сеть прямого распространения с использованием состязательного обучения. Подобно [41], их алгоритм представляет собой непараметрический метод на основе патчей с MRF. Показано, что их метод превосходит алгоритмы Джонсона и др. и Ульянова и др. в сохранении когерентных текстур в сложных изображениях благодаря их дизайну на основе патчей. Однако их алгоритм имеет менее удовлетворительную производительность с нетекстурными стилями (например, изображениями лиц), поскольку их алгоритм не учитывает семантику. Другие недостатки их алгоритма включают отсутствие учета глубинной информации и вариаций мазков кисти, которые являются важными визуальными факторами.

**4.2.2 Нейронные методы с несколькими стилями на модель**

Хотя вышеописанные подходы PSPM могут создавать стилизованные изображения на два порядка быстрее, чем предыдущие методы IOB-NST, отдельные генеративные сети должны быть обучены для каждого конкретного стиля, что довольно трудоемко и негибко. Но многие картины (например, картины импрессионистов) имеют схожие мазки, краски и отличаются только цветовыми палитрами. Интуитивно понятно, что обучать отдельную сеть для каждой из них излишне. Поэтому предлагается MSPM (Multiple-Style-Per-Model), который улучшает гибкость PSPM (Per-Style-Per-Model) за счет дальнейшего включения нескольких стилей в одну единую модель.

**Привязка только небольшого числа параметров к каждому стилю.**

Ранняя работа Дюмулена и др. [48] построена на основе предложенного слоя IN в алгоритме PSPM [45] (Раздел 4.2.1). Они неожиданно обнаружили, что использование тех же сверточных параметров, но только их масштабирование и сдвиг в слоях IN достаточно для моделирования разных стилей. Поэтому они предлагают алгоритм для обучения условной мультистильной сети передачи на основе условной нормализации экземпляра (CIN).

**4.2.3 Нейронные методы произвольного стиля на модель**

Третья категория, ASPM-MOB-NST, нацелена на одну модель для всех, т. е. одну обучаемую модель для передачи произвольных художественных стилей. Существует также два типа ASPM, один из которых построен на непараметрическом моделировании текстур с MRF, а другой — на параметрическом моделировании текстур с суммарной статистикой.

**Непараметрический ASPM с MRF.**

Первый алгоритм ASPM предложен Ченом и Шмидтом [49]. Сначала они извлекают набор патчей активации из активаций функций контента и стиля, вычисленных в предварительно обученной сети VGG. Затем они сопоставляют каждый патч контента с наиболее похожим патчем стиля и меняют их местами (называется «Обмен стилями» в [49]). Стилизованный результат может быть получен путем реконструкции результирующей карты активации после «Обмена стиля» с помощью техник IOB-IR или MOB-IR. Алгоритм Чена и Шмидта более гибок, чем предыдущие подходы, благодаря своей характеристике «одна модель для всех стилей». Но стилизованные результаты [49] менее привлекательны, поскольку патчи контента обычно меняются местами с патчами стиля, которые не являются репрезентативными для желаемого стиля. В результате контент отлично сохраняется, в то время как стиль, как правило, не хорошо отражается.

# **5. Генеративные состязательные сети**

В последние годы генеративные состязательные сети (Generative Adversarial Networks, GAN) стали одной из самых обсуждаемых и активно исследуемых технологий в области искусственного интеллекта и машинного обучения. Представленные Иэном Гудфеллоу и его коллегами [51] в 2014 году, GAN быстро зарекомендовали себя в качестве мощного инструмента для генерации данных, превосходящих по качеству решения других подходов. Архитектура GAN состоит из двух основных компонентов: генератора и дискриминатора. Генератор направлен на создание новых данных, которые, по возможности, должны быть неотличимы от реальных, в то время как дискриминатор предназначен для определения подлинности данных, различая реальные образцы от синтезированных генератором. Во время обучения обе модели соревнуются: генератор стремится обмануть дискриминатор, в то время как дискриминатор пытается лучше отличать фальшивки от реальности. Этот процесс представляет собой нулевую игру, где каждая модель со временем совершенствуется, приводя к улучшению общей производительности.

С момента создания концепции GAN на свет появилось множество модификаций, таких как DCGAN для работы с изображениями, CycleGAN для трансформации стиля изображений без парных примеров и StyleGAN для генерации фотореалистичных изображений. Эти улучшения существенно способствовали успешному применению GAN в различных прикладных областях, включая анимацию, медицинскую визуализацию и искусство.

Введение метода генеративных состязательных сетей (GAN) в область переноса стиля представляет собой значительное продвижение в области генерации изображений и обработки визуальной информации. Таким образом, генеративные состязательные сети предоставляют мощный и гибкий инструмент для реализации задачи переноса стиля, демонстрируя превосходные результаты благодаря своему уникальному подходу к обучению и генерации данных. Эти преимущества позволяют использовать GAN для создания высококачественных визуальных произведений, эффективно достигая гармоничного соединения структурных и стилистических элементов, что в результате приводит к инновационным решениям в цифровом искусстве и дизайне.

# **Список литературы**

[1] B. Gooch and A. Gooch, Non-photorealistic rendering. Natick, MA, USA: A. K. Peters, Ltd., 2001.

[2] T. Strothotte and S. Schlechtweg, Non-photorealistic computer graphics: modeling, rendering, and animation. Morgan Kaufmann, 2002.

[3] P. Rosin and J. Collomosse, Image and video-based artistic stylisation. Springer Science & Business Media, 2012, vol. 42.

[4] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2414–2423.

[5] A. A. Efros and W. T. Freeman, “Image quilting for texture synthesis and transfer,” in Proceedings of the 28th annual conference on Computer graphics and interactive techniques. ACM, 2001, pp. 341–346.

[6] I. Drori, D. Cohen-Or, and H. Yeshurun, “Example-based style synthesis,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2. IEEE, 2003, pp. II–143.

[7] O. Frigo, N. Sabater, J. Delon, and P. Hellier, “Split and match: Example-based adaptive patch sampling for unsupervised style transfer,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 553–561.

[8] M. Elad and P. Milanfar, “Style transfer via texture synthesis,” IEEE Transactions on Image Processing, vol. 26, no. 5, pp. 2338– 2351, 2017

[9] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, “Image analogies,” in Proceedings of the 28th annual conference on Computer graphics and interactive techniques. ACM, 2001, pp. 327–340.

[10] L. A. Gatys, A. S. Ecker, and M. Bethge, “A neural algorithm of artistic style,” ArXiv e-prints, Aug. 2015.

[11] I. Prisma Labs, “Prisma: Turn memories into art using artificial intelligence,” 2016. [Online]. Available: <http://prisma-ai.com>

[12] “Ostagram,” 2016. [Online]. Available: <http://ostagram.ru>

[13] A. J. Champandard, “Deep forger: Paint photos in the style of famous artists,” 2015. [Online]. Available: <http://deepforger.com>

[14] J. E. Kyprianidis, J. Collomosse, T. Wang, and T. Isenberg, “State of the ‘art’: A taxonomy of artistic stylization techniques for images and video,” IEEE transactions on visualization and computer graphics, vol. 19, no. 5, pp. 866–885, 2013.

[15] A. Semmo, T. Isenberg, and J. Dollner, “Neural style transfer: A ¨ paradigm shift for image-based artistic rendering?” in Proceedings of the Symposium on Non-Photorealistic Animation and Rendering. ACM, 2017, pp. 5:1–5:13.

[16] A. Hertzmann, “Painterly rendering with curved brush strokes of multiple sizes,” in Proceedings of the 25th annual conference on Computer graphics and interactive techniques. ACM, 1998, pp. 453– 460.

[17] A. Kolliopoulos, “Image segmentation for stylized nonphotorealistic rendering and animation,” Ph.D. dissertation, University of Toronto, 2005.

[18] B. Gooch, G. Coombe, and P. Shirley, “Artistic vision: painterly rendering using computer vision techniques,” in Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering. ACM, 2002, pp. 83–ff.

[19] Y.-Z. Song, P. L. Rosin, P. M. Hall, and J. Collomosse, “Arty shapes,” in Proceedings of the Fourth Eurographics conference on Computational Aesthetics in Graphics, Visualization and Imaging. Eurographics Association, 2008, pp. 65–72.

[20] M. Zhao and S.-C. Zhu, “Portrait painting using active templates,” in Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering. ACM, 2011, pp. 117–124.

[21] H. Winnemoller, S. C. Olsen, and B. Gooch, “Real-time video ¨ abstraction,” in ACM Transactions On Graphics (TOG), vol. 25, no. 3. ACM, 2006, pp. 1221–1226.

[22] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in Proceedings of the IEEE International Conference on Computer Vision. IEEE, 1998, pp. 839–846.

[23] B. Gooch, E. Reinhard, and A. Gooch, “Human facial illustrations: Creation and psychophysical evaluation,” ACM Transactions on Graphics, vol. 23, no. 1, pp. 27–44, 2004.

[24] L.-Y. Wei, S. Lefebvre, V. Kwatra, and G. Turk, “State of the art in example-based texture synthesis,” in Eurographics 2009, State of the Art Report, EG-STAR. Eurographics Association, 2009, pp. 93–117.

[25] A. A. Efros and T. K. Leung, “Texture synthesis by nonparametric sampling,” in Proceedings of the IEEE International Conference on Computer Vision, vol. 2. IEEE, 1999, pp. 1033–1038.

[26] L.-Y. Wei and M. Levoy, “Fast texture synthesis using treestructured vector quantization,” in Proceedings of the 27th annual conference on Computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 479–488.

[27] B. Julesz, “Visual pattern discrimination,” IRE transactions on Information Theory, vol. 8, no. 2, pp. 84–92, 1962.

[28] D. J. Heeger and J. R. Bergen, “Pyramid-based texture analysis/synthesis,” in Proceedings of the 22nd annual conference on Computer graphics and interactive techniques. ACM, 1995, pp. 229– 238.

[29] J. Portilla and E. P. Simoncelli, “A parametric texture model based on joint statistics of complex wavelet coefficients,” International journal of computer vision, vol. 40, no. 1, pp. 49–70, 2000.

[30] L. A. Gatys, A. S. Ecker, and M. Bethge, “Texture synthesis using convolutional neural networks,” in Advances in Neural Information Processing Systems, 2015, pp. 262–270.

[31] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[32] G. Berger and R. Memisevic, “Incorporating long-range consistency in cnn-based texture generation,” in International Conference on Learning Representations, 2017.

[33] A. Mahendran and A. Vedaldi, “Understanding deep image representations by inverting them,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5188–5196.

[34] “Visualizing deep convolutional neural networks using natural pre-images,” International Journal of Computer Vision, vol. 120, no. 3, pp. 233–255, 2016.

[35] A. Dosovitskiy and T. Brox, “Inverting visual representations with convolutional networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4829– 4837.

[36] “Generating images with perceptual similarity metrics based on deep networks,” in Advances in Neural Information Processing Systems, 2016, pp. 658–666.

[37] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. WardeFarley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Advances in neural information processing systems, 2014, pp. 2672–2680.

[38] X. Xie, F. Tian, and H. S. Seah, “Feature guided texture synthesis (fgts) for artistic style transfer,” in Proceedings of the 2nd international conference on Digital interactive media in entertainment and arts. ACM, 2007, pp. 44–49.

[39] M. Ashikhmin, “Fast texture transfer,” IEEE Computer Graphics and Applications, no. 4, pp. 38–43, 2003.

[40] A. Mordvintsev, C. Olah, and M. Tyka, “Inceptionism: Going deeper into neural networks,” 2015. [Online]. Available: https://research.googleblog.com/2015/06/ inceptionism-going-deeper-into-neural.html

[41] C. Li and M. Wand, “Combining markov random fields and convolutional neural networks for image synthesis,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2479–2486.

[42] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for realtime style transfer and super-resolution,” in European Conference on Computer Vision, 2016, pp. 694–711.

[43] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky, “Texture networks: Feed-forward synthesis of textures and stylized images,” in International Conference on Machine Learning, 2016, pp. 1349–1357.

[44] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” ArXiv e-prints, Nov. 2015.

[45] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6924–6932.

[46] X. Huang and S. Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1501–1510.

[47] C. Li and M. Wand, “Precomputed real-time texture synthesis with markovian generative adversarial networks,” in European Conference on Computer Vision, 2016, pp. 702–716.

[48] V. Dumoulin, J. Shlens, and M. Kudlur, “A learned representation for artistic style,” in International Conference on Learning Representations, 2017.

[49] T. Q. Chen and M. Schmidt, “Fast patch-based style transfer of arbitrary style,” in Proceedings of the NIPS Workshop on Constructive Machine Learning, 2016.

[50] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman, “Controlling perceptual factors in neural style transfer,” in Proceedings of the IEEE Conference

[51] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, & Yoshua Bengio. (2014). Generative Adversarial Networks.