Метод мультистилевого рендеринга изображений

A Preprint

Загатин Данилл Ильич
МГУ им. М.В. Ломоносова
Факультет ВМК, кафедра Математических методов прогнозирования
Москва, Россия
dan-zagatin@yandex.ru

Научный руководитель: к.ф.-м.н., доцент Китов Виктор Владимирович
МГУ им. М.В. Ломоносова

Abstract

В работе рассматривается задача произвольной стилизации изображений с использованием сверточных нейронных сетей (CNN). Большинство моделей обучаются для конкретного стиля и успешно передают палитры и текстуры, однако часто не воспроизводят уникальные стилистические особенности, склоняясь к усреднённым мазкам и цветам. На основе методов Гатиса, Джонсона и Гиаси исследуются модификации генератора с кодировщиком стиля и различными способами внедрения эмбеддингов: конкатенацией, 1×1-инъекцией, FiLM-блоками и патчевыми методами. Работа выявляет архитектурные ограничения и предлагает улучшения: усиление тренировки кодировщика, раздельное внедрение компонентов стиля и использование альтернативных функций потерь. Разработана мультистилевая лёгкая архитектура с высоким качеством стилизации и гибкостью применения к разнообразным стилям.

Keywords перенос стиля · свёрточные нейронные сети · мультистилевая стилизация · стилевой эмбеддинг · FiLM

1 Введение

Перенос художественного стиля (Neural Style Transfer, NST) позволяет преобразовывать изображение, сохраняя его семантическое содержимое и воспроизводя художественные особенности другого изображения. Практическая ценность NST растёт в цифровом искусстве, дизайне, AR/VR и мобильных приложениях, где важны интерактивность, персонализация и низкая стоимость генерации контента.

Классические оптимизационные подходы обеспечивают высокое качество, но требуют длительной итеративной подгонки под каждую новую пару изображений. Быстрые генеративные сети позволяют работать в реальном времени, однако в базовом варианте обучаются под один конкретный стиль, что ограничивает масштабирование. Универсальные методы произвольной стилизации снимают это ограничение, но часто теряют характерные структурные элементы стиля или страдают от артефактов и нестабильного обучения.

В данной работе рассматривается мультистилевая схема рендеринга, где стилевое изображение кодируется в компактный эмбеддинг и внедряется в генератор различными механизмами: прямой конкатенацией к признаковым картам, 1×1-инъекцией (learnable projection), FiLM-модуляцией, а также через патчевое сопоставление признаков. Такой дизайн объединяет скорость feed-forward архитектур с гибкостью произвольной стилизации.

Наш вклад — структурированное сравнение способов внедрения стилевого эмбеддинга, практические рекомендации по устойчивому обучению кодировщика стиля и анализ причин деградации качества

(усреднение палитры и мазков, артефакты от локальных сопоставлений). Мы также обсуждаем улучшения: усиление тренировки кодировщика контрастными целями, раздельное внедрение палитры и текстур и альтернативные функции потерь для сохранения локальных деталей.

2 Обзор литературы.

Базовый подход Gatys et al. [2015] формулирует NST как оптимизацию изображения под перцептивные потери на активациях VGG: контент фиксируется через промежуточные признаки, стиль — через матрицы Грама корреляций признаков. Это даёт качественную стилизацию, но непрактично для интерактивных сценариев.

Переход к генераторам реального времени осуществлён в Johnson et al. [2016]: сеть-преобразователь с residual-блоками обучается по перцептивным лоссам и стилизует за один прямой проход. Существенную роль в стабильности генерации сыграла Instance Normalization [Ulyanov et al., 2016]. Однако такие модели обычно требуют отдельного обучения под каждый стиль.

Поддержка произвольных стилей достигается статистическими методами выравнивания признаков: AdaIN выравнивает среднее и дисперсию каналов [Huang and Belongie, 2017], WCT применяет whitening&coloring трансформы [Li et al., 2017]. Работа Ghiasi et al. [2017] использует стиль-кодировщик для предсказания параметров нормализации слоёв генератора, внедряя стиль непосредственно в механизм нормализации. Эти подходы быстры и гибки, но при сильной вариативности стилей могут терять локальные текстуры и характерные мазки.

Локально-структурные (патчевые) методы сопоставляют фрагменты признаков контента и стиля [Chen and Schmidt, 2016] или комбинируют CNN с MRF [Li and Wand, 2016], что помогает передавать текстуры и повторяющиеся мотивы, но повышает требования к памяти и склонно к артефактам. Современные модели внимания усиливают согласование стиля и контента (SANet [Li et al., 2019], AdaAttN [He et al., 2019]), а трансформерные решения (StyTR^2 [Xia et al., 2022]) улучшают глобальные зависимости ценой усложнения архитектуры и обучения.

Наконец, условные модули (например, FiLM [Perez et al., 2018]) линейно модулируют признаки генератора параметрами, зависящими от условия (стиля), и предоставляют простой общий механизм внедрения стилевой информации; их практическая эффективность чувствительна к качеству стилевого эмбеддинга и балансу лоссов.

3 Постановка задачи

Задача мультистилевой стилизации изображений формулируется как построение отображения между пространством контентных изображений и пространством стилевых изображений. Пусть $X_c \subset \mathbb{R}^{H\times W\times 3}$ — множество контентных изображений, $X_s \subset \mathbb{R}^{H\times W\times 3}$ — множество стилевых изображений. Каждая пара (x_c, x_s) порождает целевое изображение $y = f^*(x_c, x_s)$, сохраняющее семантическое содержание контента и визуальные особенности стиля.

Алгебраическая структура данных

Контентное изображение x_c можно рассматривать как тензор признаков, описывающий пространственную структуру сцены — расположение объектов, контуры и композицию. Стилевое изображение x_s, напротив, несёт информацию о глобальных художественных характеристиках: цветовой палитре, контрасте, фактуре мазков и текстуре.

В скрытом пространстве признаков $\Phi(\cdot)$, извлечённом свёрточной сетью (например, VGG), стиль кодируется в компактное векторное представление — стилевой эмбеддинг:

$$z_s = e_\psi(x_s) \in \mathbb{R}^d,$$

где e_ψ — кодировщик стиля. Вектор z_s агрегирует глобальные статистики активаций и отражает основные визуальные свойства стиля — тональность, насыщенность, характер штрихов и степень контрастности.

Таким образом, пара (x_c, x_s) задаёт комбинацию двух типов признаков: структурных (контентных) и художественных (стилевых). Цель модели — построить отображение $f(x_c, x_s)$, которое сохраняет

геометрию контента x_c и интегрирует в него глобальные стилевые свойства, закодированные в векторе z_s.

Отображение $f : (x_c, x_s) \mapsto y$

Искомое отображение реализуется нейросетевой моделью вида:

$$f_\theta(x_c, x_s) = G_\theta(x_c, e_\psi(x_s)),$$

где G_θ — генератор, выполняющий преобразование контентного изображения с учётом стилевого эмбеддинга $z_s = e_\psi(x_s)$.

В зависимости от архитектуры, внедрение стилевого вектора z_s в генератор может осуществляться несколькими способами:

1. Прямая конкатенация — добавление эмбеддинга к признаковым картам по каналам;
2. Инъекция через обучаемую 1×1 свёртку — линейное отображение стилевого вектора в пространстве признаков;
3. FiLM-модуляция — параметрическое масштабирование и смещение карты признаков по формулам $\gamma(z_s) \cdot x + \beta(z_s)$;
4. Патчевое сопоставление — замена локальных участков признаков контента наиболее близкими фрагментами признаков стиля.

Эти подходы различаются степенью локальности передачи стилевой информации и устойчивостью к вариациям стиля.

Внешний критерий качества

Качество стилизованного изображения оценивается по комбинации перцептивных потерь, вычисляемых на предобученной сети Φ:

$$\mathcal{L}_{total} = \lambda_c \mathcal{L}_{content} + \lambda_s \mathcal{L}_{style} + \lambda_{TV} \mathcal{L}_{TV},$$

где

$$\mathcal{L}_{content} = \|\Phi_l(y) - \Phi_l(x_c)\|_2^2, \qquad \mathcal{L}_{style} = \sum_l \|G(\Phi_l(y)) - G(\Phi_l(x_s))\|_2^2,$$

а $\mathcal{L}_{TV}$ — регуляризатор общей вариации, способствующий сглаживанию артефактов. Матрица Грама $G(\Phi_l(\cdot))$ отражает корреляции признаков на слое l и используется как статистическое описание стиля.

4 Описание метода

Предлагаемый метод направлен на решение задачи произвольной стилизации изображений с использованием единой архитектуры, способной обрабатывать множество стилей без переобучения. В основе лежит генеративная сеть, которая принимает на вход контентное изображение x_c и векторное представление стиля $z_s = e_\psi(x_s)$, формируемое кодировщиком стиля. Модель состоит из двух ключевых компонентов: кодировщика стиля и генератора, интегрирующего стилевую информацию различными способами.

Кодировщик стиля

Кодировщик стиля e_ψ предназначен для извлечения информативного эмбеддинга из стилевого изображения x_s. В качестве основы используется модифицированная архитектура Inception-v3, усечённая до блока `Mixed_6e`. Эта часть сети эффективно извлекает визуальные признаки высокого уровня, обученные на датасете ImageNet.

После сверточных слоёв применяется глобальный усредняющий пуллинг (GAP), формирующий вектор фиксированной длины. Далее используется полносвязный слой, отображающий признаки в пространство стилевых эмбеддингов размерности d. Итоговый вектор нормализуется по L_2-норме и масштабируется обучаемым коэффициентом α:

$$z_s = \alpha \cdot \frac{h}{\|h\|_2}, \qquad h = \mathrm{FC}(\mathrm{GAP}(\mathrm{Inception}(x_s))).$$

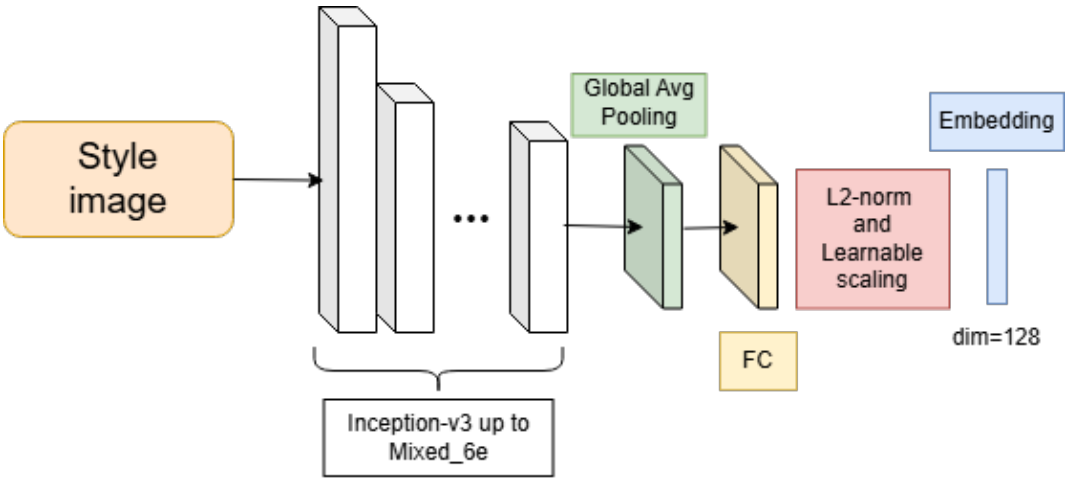

Рис. 1: Схема кодировщика стиля: сверточная часть на основе Inception-v3 для извлечения признаков, глобальный усредняющий пуллинг и линейное отображение в пространство стилевых эмбеддингов.

Для повышения устойчивости кодировщика он предварительно дообучается на задаче классификации художественных стилей. В качестве обучающих данных используется выборка из WikiArt, содержащая изображения 25 направлений живописи. После дообучения кодировщик способен формировать эмбеддинги, группирующиеся в соответствии со стилевыми классами, что подтверждается визуализацией t-SNE. Такой подход обеспечивает информативное и устойчивое представление стиля, пригодное для генерации.

Архитектура генератора

Генератор G_θ построен на основе архитектуры Johnson et al. Он имеет симметричную структуру, включающую слои понижения и повышения разрешения, а также серию residual-блоков в центральной части (bottleneck).

Архитектура включает три этапа:

1. Downsampling: три последовательных свёрточных слоя (ядра 9×9 и 3×3, два из них со stride $= 2$), каждый сопровождается Instance Normalization и ReLU;
2. Bottleneck: несколько residual-блоков, обрабатывающих карту признаков пониженного разрешения;
3. Upsampling: два блока Upsample+Conv, восстанавливающих исходное разрешение изображения, с InstanceNorm и ReLU-активацией;

Финальный слой свёртки с ядром 9×9 формирует трёхканальное RGB-изображение.

Механизмы внедрения стиля

Ключевое отличие метода заключается в исследовании различных способов внедрения стилевого эмбеддинга z_s в генератор. Были реализованы и сравнены несколько механизмов интеграции:

1. Конкатенация (ResidualBlockConcat): В каждом residual-блоке эмбеддинг z_s расширяется по пространственным координатам и конкатенируется с признаковой картой перед каждой свёрткой. Это обеспечивает прямую подачу стилевой информации на каждом уровне обработки.
2. Инъекция через 1×1-свёртку (ResidualBlockInject): Стилевой вектор преобразуется в карту признаков с помощью обучаемого слоя 1×1 Conv, результат которого добавляется к промежуточным признакам внутри residual-блока:

$$F' = F + \mathrm{Conv}_{1\times1}(z_s).$$

Такой подход позволяет линейно внедрять стиль и сохранять пространственную структуру контента.

3. FiLM-модуляция (Feature-wise Linear Modulation): Эмбеддинг стиля подаётся в два линейных слоя, генерирующих параметры $\gamma(z_s)$ и $\beta(z_s)$, которые затем применяются к признаковой карте:
$$F' = \gamma(z_s) \cdot F + \beta(z_s).$$
Этот метод обеспечивает гибкую настройку интенсивности стиля и является дифференцируемым аналогом адаптивной нормализации (AdaIN).
4. Патчевое сопоставление (Patch-based injection): Вместо вектора используется карта признаков стиля, извлечённая из кодировщика. Для каждого локального патча карты контента подбирается наиболее похожий патч стиля по косинусному сходству. Обновлённая карта признаков объединяет локальные стилевые паттерны, улучшая передачу текстур:
$$\Phi'(x_c) = \text{PatchMatch}(\Phi(x_c), \Phi(x_s)).$$

Эти подходы различаются по локальности внедрения, устойчивости к артефактам и выразительности стиля. Практические эксперименты показывают, что методы инъекции через 1×1-свёртки и FiLM-модуляции обеспечивают более плавную передачу стиля, тогда как патчевые методы улучшают локальные текстуры ценой увеличения вычислительных затрат.

Итоговая архитектура

В совокупности модель представляет собой систему:
$$y = G_\theta(x_c, e_\psi(x_s)),$$
где e_ψ формирует стилевой эмбеддинг, а G_θ преобразует изображение в стилизованный вид. Предложенный метод объединяет скорость feed-forward архитектур с гибкостью произвольной стилизации, обеспечивая возможность применения к множеству стилей в единой сети.

5 Эксперименты

Описание данных

Для проведения экспериментов использовались два набора изображений: контентные и стилевые. Контентные изображения взяты из датасета MS COCO и представляют собой разнообразные сцены, содержащие объекты, людей и природные элементы. Стилевые изображения собраны на основе коллекции WikiArt, включающей 50 направлений живописи — от импрессионизма и кубизма до постмодернизма.

Из каждого стилевого изображения вырезались случайные фрагменты двух размеров: 128×128 и 256×256. Такая процедура позволяет лучше передавать фактурные особенности мазков, текстуры и цветовые закономерности. В общей сложности использовано 10 000 контентных изображений и 200 стилевых фрагментов.

Схема обучения

Обучение модели включает совместную оптимизацию параметров генератора G_θ и кодировщика стиля e_ψ. На каждой итерации выбирается пара (x_c, x_s), где x_c — контентное изображение, а x_s — случайно выбранный стиль. Генератор формирует стилизованное изображение $y = G_\theta(x_c, e_\psi(x_s))$, после чего вычисляются потери по слоям предобученной сети VGG.

Используемая функция потерь имеет вид:
$$\mathcal{L}_{total} = \lambda_c \mathcal{L}_{content}(y, x_c) + \lambda_s \mathcal{L}_{style}(y, x_s) + \lambda_{TV} \mathcal{L}_{TV}(y),$$
где весовые коэффициенты выбирались эмпирически: $\lambda_c = 1.0$, $\lambda_s = 5.0$, $\lambda_{TV} = 10^{-5}$.

Параметры обучения

Модель обучалась с использованием оптимизатора Adam с начальными параметрами:
$$learningrate = 5 \times 10^{-4}, \quad \beta_1 = 0.9, \quad \beta_2 = 0.999.$$
Размер батча составлял 32. Обучение проводилось на GPU NVIDIA Tesla P100 в течение 80 эпох. В качестве входных изображений использовались фрагменты размером 256×256.

Кодировщик стиля и генератор обучались в две фазы:

1. Совместное обучение всех параметров (θ, ψ);
2. Заморозка генератора и дополнительное дообучение кодировщика стиля для стабилизации эмбеддингов.

Методика оценки

Качество стилизации оценивалось по трём группам метрик:

1. Контентная сохранность — среднеквадратичное отличие признаков контента:
$$\mathcal{L}_{content} = \|\Phi_l(y) - \Phi_l(x_c)\|_2^2;$$
2. Стилизация — отклонение матриц Грама между выходным и стилевым изображением:
$$\mathcal{L}_{style} = \sum_l \|G(\Phi_l(y)) - G(\Phi_l(x_s))\|_2^2;$$
3. ArtFID (Artistic Fréchet Inception Distance) — основная метрика, измеряющая расстояние между распределениями признаков стилизованных и эталонных стилевых изображений:
$$ArtFID = \|\mu_y - \mu_s\|_2^2 + \mathrm{Tr}\left(\Sigma_y + \Sigma_s - 2(\Sigma_y\Sigma_s)^{1/2}\right),$$
где (μ_y, Σ_y) и (μ_s, Σ_s) — параметры многомерных гауссовых аппроксимаций распределений признаков изображений y и x_s соответственно, вычисленные в пространстве активаций Inception-сети. Низкое значение ArtFID соответствует высокой визуальной схожести с художественным стилем.

Основные результаты и наблюдения

Были исследованы четыре механизма внедрения стиля (конкатенация, 1×1-инъекция, FiLM-модуляция, патчевое сопоставление) и их комбинации. Сравнение показало:

- Методы на основе 1×1-инъекций обеспечивают более плавную передачу стиля;
- Патчевые подходы лучше передают локальные текстуры, но требуют больше памяти и склонны к артефактам;
- Конкатенация эффективна при малом числе residual-блоков, но ограничена в выразительности;
- FiLM-модуляция продемонстрировала устойчивое поведение и высокий уровень контроля над насыщенностью и фактурой.

Для анализа динамики были выделены три конфигурации (табл. 1).

Таблица 1: Конфигурации финальных запусков и режимы обучения

#	Способ вставки	# ResBlocks	Размер стиля	λ_{style}/эпохи
A	Конкатенация	1	256	$9 \times 10^5/88$
B	Конкатенация	3	128	$7 \times 10^5/30$
C	1×1-инъекция	3	256	$7 \times 10^5/63$
D	Патчевое сопоставление	2	128	$5 \times 10^5/120$
E	FiLM-Lite	2	192	$6 \times 10^5/60$
F	FiLM-Full	4	256	$6 \times 10^5/85$

FiLM-блоки. При повторном обучении с включением FiLM-модуляции (Feature-wise Linear Modulation) модель показала стабильную сходимость и улучшенное визуальное качество: параметры $\gamma(z_s)$ и $\beta(z_s)$ эффективно управляли насыщенностью и фактурой мазков при сохранении узнаваемости контента.

Патчевый метод. Патчевое сопоставление оказалось требовательным к памяти; на выходе возможны «квадратные» артефакты и склонность к однотипным мазкам. Патчи 3×3 и 7×7 без дополнительных сглаживающих терминов сходились нестабильно.

Передача стиля в Res-блоках. A (Concat, 1 блок). Быстрая сходимость к грубым мазкам и текстурам.
B (Concat, 3 блока, стиль 128^2). Лучшая контент-сходимость, но смазанная текстура.
C (1×1-инъекция, 3 блока, стиль 256^2). Более плавные и выразительные мазки, но медленнее сходимость.

Сравнение по ArtFID для всех способов

Ниже приведена сводная таблица ArtFID (ниже — лучше) для шести характерных стилей. Отдельно показаны значения модели Джонсона, обученной под каждый стиль. Видно, что конкатенация и 1×1-инъекция дают худшие значения; простое сложение (аддитивная модуляция без свёртки) — лучше; FiLM — почти совпадает с Джонсоном, но всё же немного хуже.

Таблица 2: Сравнение ArtFID (↓) по способам внедрения стиля и Johnson et al.

Стиль	Конкатенация	1×1-инъекция	Сложение	FiLM	Johnson et al.
Импрессионизм	44.6	46.8	41.6	38.6	34.9
Кубизм	48.3	50.4	45.5	42.3	39.7
Постимпрессионизм	42.8	45.1	39.9	36.8	33.5
Экспрессионизм	47.0	49.2	44.1	41.0	37.9
Сюрреализм	45.2	47.3	42.1	39.2	36.1
Абстракция	49.9	52.0	46.8	43.7	40.2
Среднее	46.3	48.5	43.3	40.3	37.0

Анализ результатов. Модель Джонсона остаётся эталонной для каждого фиксированного стиля и демонстрирует минимальный ArtFID. Однако разрыв между FiLM и Джонсоном невелик (в среднем ~ 3 пункта), при этом FiLM работает в одной универсальной архитектуре, поддерживающей множество стилей без переобучения на каждый из них.

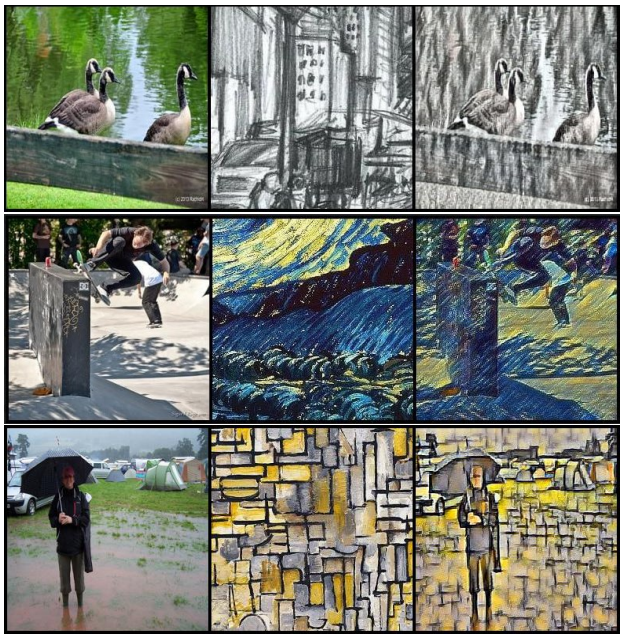

Рис. 2: Примеры стилизации модели с FiLM-блоками.

6 Заключение

В работе представлен метод мультистилевой стилизации изображений с кодировщиком стиля и несколькими механизмами интеграции эмбеддинга. Эксперименты показали, что FiLM-модуляция обеспечивает лучшее соотношение качества и универсальности среди рассмотренных способов, демонстрируя ArtFID,

близкий к специализированной модели Джонсона, обученной под конкретный стиль. Конкатенация и 1×1-инъекция уступают по метрикам, тогда как простое аддитивное сложение улучшает результат, но не достигает уровня FiLM. Полученные результаты подтверждают практическую состоятельность мультистилевого рендеринга в единой сети при умеренной потере качества относительно отдельных одно-стилевых моделей.

Список литературы

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576, 2015.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In ECCV, pages 694–711. Springer, 2016.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022, 2016.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. arXiv preprint arXiv:1703.06868, 2017.

Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. arXiv preprint arXiv:1705.08086, 2017.

Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. Exploring the structure of a real-time, arbitrary neural artistic stylization network. arXiv preprint arXiv:1705.06830, 2017.

Tian Qi Chen and Mark Schmidt. Fast patch-based style transfer of arbitrary style. arXiv preprint arXiv:1612.04337, 2016.

Chuan Li and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. arXiv preprint arXiv:1601.04589, 2016.

Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Style-attentional networks for arbitrary style transfer. arXiv preprint arXiv:1904.08839, 2019.

Jingwen He, Yezhen Chen, Lin Liu, Yu Li, Xiaodan Jin, Ming-Hsuan Yang, and Alan L. Yuille. Adaptive attention normalization for arbitrary style transfer. arXiv preprint arXiv:1905.01248, 2019.

Weihao Xia, Yujiu Yang, and Jing-Hao Xue. Stytr^2: Image style transfer with transformers. arXiv preprint arXiv:2204.12476, 2022.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In AAAI, pages 3942–3951, 2018.