ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ

УДК 004.89 **DOI 10.52928/2070-1624-2025-45-2-2-8**

WAVESTYLEGAN: ВЕЙВЛЕТ-ГЕНЕРАТИВНАЯ СОСТЯЗАТЕЛЬНАЯ СЕТЬ

В. А. ВОРОБЕЙ, канд. физ.-мат. наук, доц. А. Э. МАЛЕВИЧ
(Белорусский государственный университет, Минск)

U. Varabei ORCID https://orcid.org/0009-0006-9604-8894
A. Malevich ORCID https://orcid.org/0000-0001-8716-8655

Разработана модель генеративной состязательной сети WaveStyleGAN для работы с изображениями на основе семейства архитектур StyleGAN. Ключевыми особенностями предложенной архитектуры являются переход к обработке вейвлет-признаков изображений, использование в дискриминаторе самомодулируемых сверток, а также модифицированных блоков Fast Fourier Convolution. Внесенные изменения позволили уменьшить сложность и размер модели по сравнению с базовыми версиями. Полученная модель была обучена на наборе данных человеческих лиц FFHQ в разрешении 1024×1024 и смогла сохранить высокое качество генерации изображений при значительно уменьшенном количестве итераций обучения. Время работы обученной сети на CPU сократилось примерно втрое по сравнению с оригинальной моделью, что существенно расширяет возможности по ее встраиванию в окружения, где отсутствует доступ к вычислениям на графическом процессоре.

Ключевые слова: *генеративные состязательные сети, генерация изображений, дискретное вейвлет-преобразование, вейвлеты, нейронные сети.*

Введение. За последние несколько лет модели генерации изображений достигли значительного прогресса и теперь зачастую вызывают восхищение качеством создаваемых картинок: детализированностью, стилем, согласованностью отдельных частей и предметов в кадре. Наиболее популярными моделями для создания изображений являются диффузионные сети (Stable Diffusion 3.5 [1], DALL-E 3 [2], Midjourney и др.), так как они обеспечивают отличное качество генерации, а кроме того, пользователь при взаимодействии с ними может на естественном языке описывать то, что должно быть на картинке, как это должно выглядеть, какой требуется визуальный стиль. Поскольку таким моделям в процессе обучения предлагалось огромное количество информации о самых разных областях окружающего мира, то они не ограничены какой-то определенной узкой сферой при обработке входящих текстовых запросов. Однако у такого класса сетей есть и ряд существенных недостатков: огромное количество обучаемых параметров, медленная скорость работы и обучения, самих моделей в принципе может не быть в открытом доступе (семейство DALL-E, Midjourney), а воспользоваться ими можно лишь через сторонние сервисы без непосредственного доступа к самим сетям. Если же по какой-то причине необходима модель в некой определенной области, важно ее быстродействие и доступны данные со значительным количеством примеров (хотя иногда может быть достаточно и 500–1000 экземпляров), то в таком случае целесообразнее рассмотреть генеративные состязательные сети (Generative Adversarial Networks). Их качество может даже превосходить качество диффузионных сетей, они работают значительно быстрее и могут быть локально обучены за разумное время (если сравнивать с архитектурами генерации изображений общего назначения), а в дальнейшем и встроены в устройства пользователей при необходимости. Золотым стандартом среди моделей этого класса являются архитектуры семейства StyleGAN и их варианты: StyleGAN2 [3], StyleGAN3 [4], StyleGAN-XL [5] и др. Однако и у них имеются недостатки, которые выражаются в большом времени обучения, не самой высокой скорости работы (хотя и значительно быстрее диффузионных сетей). В данной работе при помощи вейвлетов и ряда других модификаций показано, как можно добиться более быстрой сходимости, а также сократить время работы конечной модели.

Описание модели. Реализованная модель построена на основе базовой версии архитектуры StyleGAN2 с некоторыми изменениями, которые приведены ниже.

Вместо обработки стандартного изображения был произведен переход к его вейвлет-образу: генератор создает не RGB-изображение, а его вейвлет-преобразование, которое в дальнейшем подается на вход дискриминатору. Таким образом, выход генератора и вход дискриминатора имеют не 3, а 12 каналов соответственно: на каждый исходный канал приходится по 4 компоненты вейвлет-разложения: LL, LH, HL, HH. Такой подход позволяет явно разделить для моделей признаки с высокими и низкими частотами, что упрощает задачу обучения сетей. Дополнительным преимуществом является возможность исключения блока

с самым высоким разрешением признаков из обеих моделей (а он потребляет значительную часть ресурсов при работе), поскольку при обратном вейвлет-преобразовании естественным образом удваивается ширина и высота тензора, а количество каналов уменьшается в 4 раза. Данный подход аналогичен тому, что использовали авторы архитектуры SWAGAN [6], которая в свою очередь также опирается на StyleGAN2, однако в отличие от простейших вейвлетов Хаара в SWAGAN в данной работе используется семейство вейвлетов CDF-9/7, реализованное через лифтинг-схему.

В генераторе были обновлены обе сети: и отображения признаков (Mapping), и синтеза изображений (Synthesis). В сети отображения размер входного скрытого вектора шума был сокращен с 512 до 192. Это обусловлено стремлением к более высокой концентрации признаков целевого изображения, а также тем, что многие данные (в частности, лица людей и, например, датасет ImageNet) эффективно описываются меньшим числом переменных. Также в данной сети количество полносвязных слоев бралось равным 3, а множитель для скорости обучения был увеличен с 0,01 до 0,03. Первый слой сети синтеза был заменен с обучаемого константного (StyleGAN2) на слой Фурье-признаков (StyleGAN3). В качестве активации для сглаживания модели и ее градиентов при обучении использовалась активация SiLU вместо LeakyReLU. Схема генератора представлена рисунке 1.

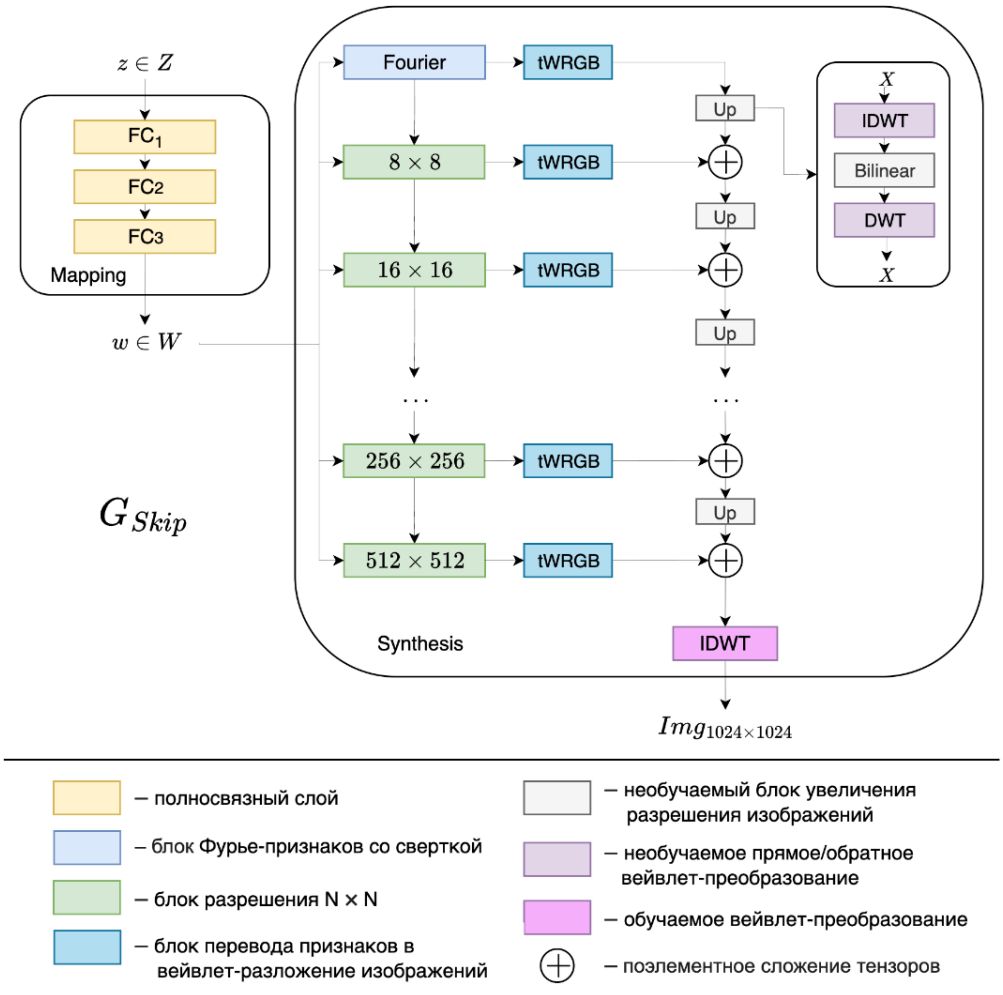

Рисунок 1. – Архитектура генератора модели WaveStyleGAN

Архитектура дискриминатора была заменена с ResNet на Skip по аналогии со SWAGAN. Для нормализации сигнала, распространяемого по сети, применялись самомодулируемые свертки [7]. Также для более полной обработки признаков и дополнительного учета спектральной информации сигнала был модифицирован и встроен блок Fast Fourier Convolution (FFC) [8]. В него были добавлены самомодулируемые свертки и вейвлеты. Общая схема блока FFCv2 приведена на рисунке 2. Наконец, в выходные линейные слои была добавлена спектральная нормализация для повышения стабильности обучения. Итоговая схема дискриминатора показана на рисунке 3.

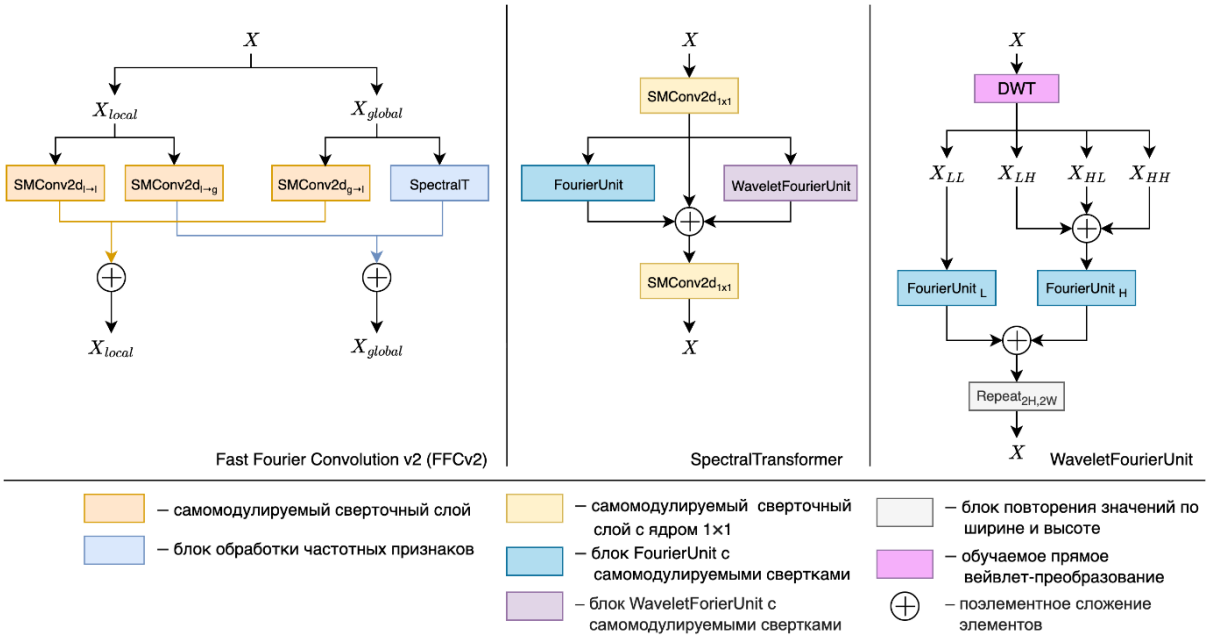

Рисунок 2. – Архитектура блока FFCv2

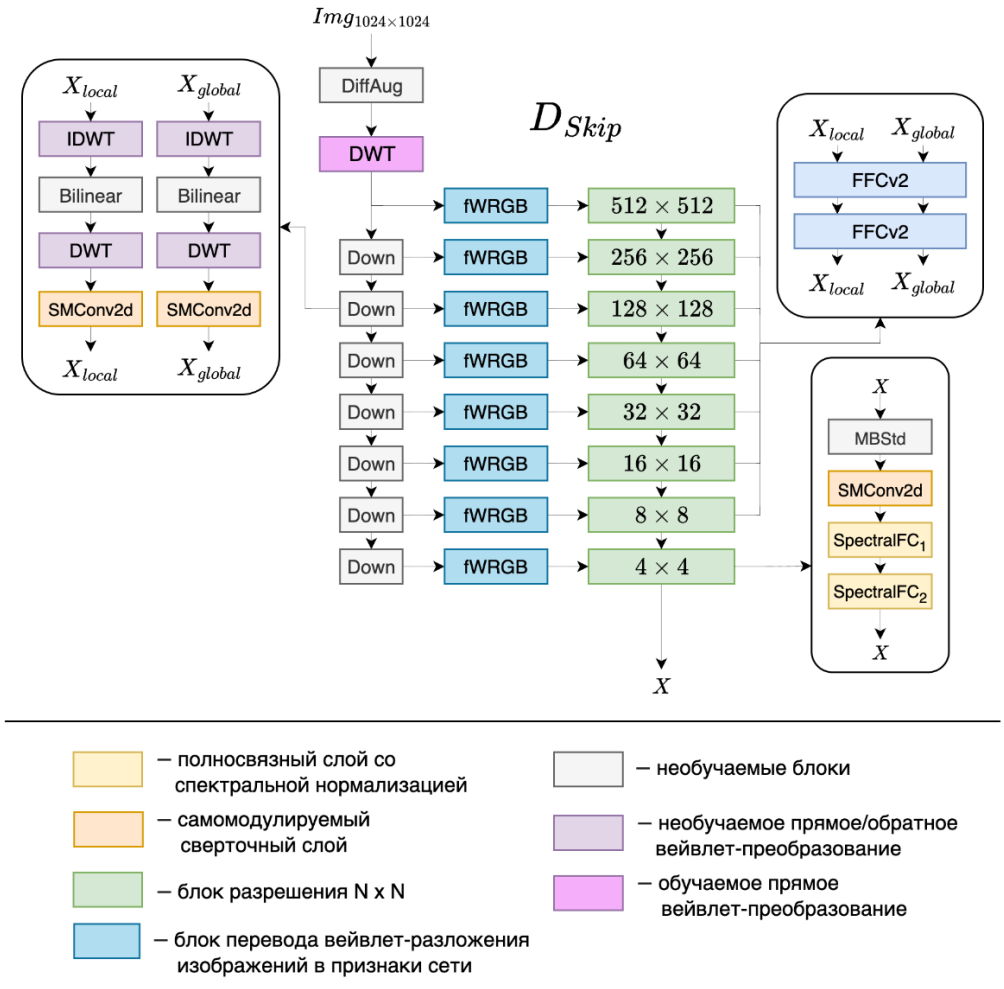

Рисунок 3. – Архитектура дискриминатора модели WaveStyleGAN

Для увеличения разнообразия данных, используемых в обучении, помимо стандартного отражения относительно вертикальной оси применялся модуль дифференцируемых аугментаций DiffAug [9]. С учетом использования вейвлетов, а также выходного формата модели генератора входная картинка для дискриминатора сначала проходит через обратное вейвлет-преобразование, к ней применяются аугментации в пространстве RGB, а затем она возвращается в исходный формат при помощи прямого вейвлет-преобразования.

Среди прочих значимых изменений следует отметить уменьшение количества фильтров в обеих моделях, начиная с разрешения 64×64, по сравнению с базовыми моделями StyleGAN2 и StyleGAN3-T, что позволило увеличить скорость тренировки, снизить расход видеопамяти, а также повысить быстродействие уже обученной модели.

Таким образом, все внесенные изменения не только не увеличили, а наоборот заметно уменьшили вычислительную сложность генератора, что дает положительный эффект при длительном использовании обученной модели. Возможность конвертации модели генератора в другие форматы (например, ONNX, CoreML и TensorFlow Lite) также была сохранена, поскольку блоки FFCv2, которые внутри себя используют комплексные числа и преобразование Фурье, применялись лишь в дискриминаторе.

Обучение модели. Модель была реализована с использованием фреймворка PyTorch. Обучение проходило на сервере с двумя видеокартами RTX3090 24 Гб. Для увеличения эффективности тренировки также использовались скомпилированные CUDA-плагины, предложенные авторами оригинальных моделей StyleGAN.

Размер пакета (батча) для каждой видеокарты брался равным 24, в качестве оптимизаторов для обеих сетей использовался Adam со скоростью обучения 0,002, параметры моментов устанавливались равными 0 и 0,99 соответственно. Функция потерь и ее параметры были аналогичны той, что использовалась при обучении исходной модели StyleGAN2. Чтобы на начальных этапах дискриминатору было сложнее отличать сгенерированные данные от настоящих, применялась техника размытия картинок, аналогичная той, что используется в моделях StyleGAN3: на протяжении первых 200 000 изображений вход в дискриминатор проходил через фильтр Гаусса, при этом параметр σ уменьшался с 10 до 0 линейно в соответствии с числом обработанных картинок. Для повышения стабильности обучения регуляризационная функция потерь для генератора на этом этапе не применялась (аналогично StyleGAN-XL).

Общее количество обработанных изображений в разрешении 1024×1024 составило 9,99 млн, а время обучения – 10 дней и 5 часов. На каждой видеокарте использовалось 20,1 Гб памяти. Средняя скорость обработки 1000 картинок составила 85,1 секунды, что соответствует пропускной способности в 11,82 изображения в секунду при обучении модели.

Результаты. Реализованная модель была обучена на датасете человеческих лиц FFHQ в разрешении 1024×1024 (использовались все 70 000 примеров). Изображения, полученные финальной версией модели, показаны на рисунке 4, а наиболее удачные из них – на рисунке 5. Качество генерации можно дополнительно повысить, сблизив вход сети синтеза генератора со средним значением, накопленным во время обучения (метод truncation trick), хотя в этом случае несколько уменьшается разнообразие генерируемых изображений. Такие примеры для $\psi = 0{,}75$ показаны на рисунке 6 (нижний ряд). Для сравнения, верхний ряд этого же рисунка иллюстрирует результаты без использования усреднения ($\psi = 1$).

Поскольку в большинстве случаев из обученных моделей в постоянное использование переходит только генератор (дискриминатор нужен лишь в обучении), то и наибольший интерес представляет сравнение сетей непосредственной генерации изображений различных архитектур. Результаты такого сравнения приведены в таблице. Для замера скорости работы моделей использовался процессор AMD Ryzen 9 5900 и видеокарта RTX 3090.

Таблица. – Сравнение генераторов моделей

Модель	Обучаемые параметры, М	Необучаемые параметры, К	Скорость работы, PyTorch CPU, мс	Скорость работы, PyTorch GPU, мс
StyleGAN2	30,37	2797	703	18,5
StyleGAN3-T	22,32	2,4	16278	40,3
StyleGAN3-R	15,10	5,6	31312	44,8
WaveStyleGAN	23,04	701,7	233	27,5

Следует отметить, что крайне низкая скорость работы моделей StyleGAN3 на CPU обусловлена использованием специальных CUDA-ядер, недоступных при отсутствии GPU-ускорителя, что существенно затрудняет встраивание этих моделей в системы, функционирующие исключительно на CPU. Потенциально это может создать значительные сложности даже при работе на GPU вне среды PyTorch, так как высокоэффективные реализации соответствующих блоков будут заменены на стандартные функции и объекты из фреймворка PyTorch при экспортировании модели в требуемый формат.

Рисунок 4. – Случайные сгенерированные изображения

Рисунок 5. – Лучшие сгенерированные изображения

Рисунок 6. – Изображения, полученные методом truncation trick (ψ = 0,75)

Заключение. В данной работе была реализована и успешно протестирована на наборе данных FFHQ в разрешении 1024×1024 модель вейвлет-генеративной состязательной сети WaveStyleGAN. Отличительные особенности предложенной архитектуры включают использование вейвлетов, а также модификацию дискриминатора путем добавления самомодулируемых сверток и модифицированных блоков FFC. К менее значительным изменениям можно отнести замену активации генератора, его входного блока, а также сокращение слоев в сети отображения скрытых признаков. Данные изменения позволили уменьшить в 2 раза число каналов в сверточных слоях обеих моделей, начиная с разрешения 64×64 и выше, а также убрать блоки в целевом разрешении изображений, на которые приходится значительная часть вычислений. Совокупность указанных изменений обеспечила примерно трехкратное ускорение работы обученной сети генератора на CPU, что значительно расширяет возможности по встраиванию моделей генеративных сетей в различные сервисы и мобильные устройства, обеспечивая заметный прирост производительности по сравнению с базовыми моделями семейства StyleGAN. На GPU предложенная модель работает немного медленнее базового варианта, однако есть основания полагать, что на графических процессорах мобильных устройств может быть получен прирост производительности, сопоставимый с тем, что наблюдается в настольных вариантах при переходе от CPU к GPU, так как мобильные GPU сами по себе гораздо слабее и их архитектура больше оптимизирована именно под работу готовых моделей, а не их обучение. Также следует отметить, что качество обученной сети сохранилось на высоком уровне, при этом для его достижения было проведено в 2,5 раза меньше тренировочных итераций по сравнению с исходными моделями.

ЛИТЕРАТУРА

1. Scaling Rectified Flow Transformers for High-Resolution Image Synthesis / P. Esser, S. Kulal, A. Blattmann et al. // arXiv.org. – 2024. – DOI: 10.48550/arXiv.2403.03206.
2. DEsignBench: Exploring and Benchmarking DALL-E 3 for Imagining Visual Design / K. Lin, Z. Yang, L. Li et al. // arXiv.org. – 2023. – DOI: 10.48550/arXiv.2310.15144.
3. Analyzing and Improving the Image Quality of StyleGAN / T. Karras, S. Laine, M. Aittala et al. // arXiv.org. – 2019. – DOI: 10.48550/arXiv.1912.04958.
4. Alias-Free Generative Adversarial Networks / T. Karras, M. Aittala, S. Laine et al. // arXiv.org. – 2021. – DOI: 10.48550/arXiv.2106.12423.
5. Sauer A., Schwarz K., Geiger A. StyleGAN-XL: Scaling StyleGAN to Large Diverse Datasets // arXiv.org. – 2022. – DOI: 10.48550/arXiv.2202.00273.
6. SWAGAN: A Style-based Wavelet-driven Generative Model / R. Gal, D. Cohen, A. Bermano, D. Cohen-Or // arXiv.org. – 2021. – DOI: 10.48550/arXiv.2102.06108.
7. LiteVAE: Lightweight and Efficient Variational Autoencoders for Latent Diffusion Models / S. Sadat, J. Buhmann, D. Bradley et al. // arXiv.org. – 2024. – DOI: 10.48550/arXiv.2405.14477.

8. Chi L., Jiang B., Mu Y. Fast Fourier Convolution // Advances in Neural Information Processing Systems 33 (NeurIPS 2020). – 2020. – URL: https://papers.nips.cc/paper_files/paper/2020/hash/2fd5d41ec6cfab47e32164d5624269b1-Abstract.html (date of access: 07.09.2025).
9. Differentiable Augmentation for Data-Efficient GAN Training / S. Zhao, Z. Liu, J. Lin et al. // arXiv.org. – 2020. – DOI: 10.48550/arXiv.2006.10738.

Поступила 08.09.2025

WAVESTYLEGAN: WAVELET-GENERATIVE ADVERSARIAL NETWORK

U. VARABEI, A. MALEVICH
(Belarusian State University, Minsk)

In this paper a novel generative adversarial network for images WaveStyleGAN that is based on StyleGAN-like architectures, was developed. Key features of the model suggested are processing of wavelet features of images, usage of self-modulated convolutions and modified blocks of Fast Fourier Convolutions in the discriminator. The changes implemented helped to reduce model complexity and its size when compared to the base models' versions. The model was trained on a dataset of human faces FFHQ in 1024×1024 resolution. It was able to keep a high quality of generated images with considerable decrease in a number of training iterations. Additionally, inference time on CPU was reduced by up to 3 times when compared to the original model, which significantly expands its capabilities for deployments to environments which don't provide access to computations on GPU.

Keywords: *generative adversarial networks, images generation, discrete wavelet transform, wavelets, neural networks.*