**Курсовая работа на тему: Основные подходы к антиалисингу на основе нейросети.**

**2025.**

**Содержание.**

**0. Введение.**

**0.1 Актуальность темы.**

**- Роль антиалиасинга в современных приложениях (игры, VR/AR).**

**- Ограничения классических методов (MSAA, FXAA, TAA).**

**- Преимущества нейросетевых подходов (адаптивность, скорость, качество).**

**0.2 Цель работы.**

**- Исследование нейросетевых методов устранения алиасинга и их практическая реализация.**

**0.3 Задачи.**

**- Анализ существующих подходов.**

**- Разработка и обучения модели.**

**- Сравнение с классическими моделями.**

**1. Теоретические основы антиалиасинга.**

**1.1 Понятие алиасинга.**

**- Причины возникновения (дискретизация сигналов, геометрические искажения).**

**- Визуальные артефакты (зубчатые края, мерцание).**

**1.2 Математические основы.**

**1.3 Классические методы.**

**- MSAA (Multi-Sample Anti-Aliasing).**

**- FXAA (Fast Approximate Anti-Aliasing).**

**- TAA (Temporal Anti-Aliasing).**

**2. Обзор современных нейросетевых подходов.**

**2.1 Архитектуры CNN.**

**- U-Net, ResNet, DnCnn: принципы работы и применение.**

**2.2 GAN-модели.**

**- ESRGan, StyleGan: генерация детализированных изображений.**

**2.3 Гибридные подходы.**

**- Комбинация CNN и трансформеров (ViT, Swin Transformer).**

**2.4 Датасеты и метрики.**

**- Синтетические данные (рендеры 3D-сцен).**

**- Метрики качества: PSNR, SSIM, LPIPS.**

**3. Практическая реализация нейросетевого антиалиасинга.**

**3.1 Разработка архитектуры.**

**- Выбор модели.**

**3.2 Подготовка данных.**

**- Генерация синтетического датасета (рендеринг сцены с/без сглаживания).**

**- Аугментация данных (Повороты, шум, изменения яркости). ???**

**3.3 Обучение модели.**

**- Функции потерь.**

**3.4 Применение модели.**

**3.5 Результаты эксперимента.**

**3.6 Сравнение с традиционными методами.**

**3.7 Ограничения и улучшения.**

**4. Заключение.**

**Список литературы (источников).**

1. Статьи:
   * FSRCNN (arXiv:1608.00367).
   * ESRGAN (arXiv:1809.00219).
   * Vision Transformer (arXiv:2010.11929).
2. Ресурсы:
   * GitHub-репозитории с реализациями (DeepAA, NVIDIA DLSS).
   * Курсы: GAMES202 (Advanced Real-Time Rendering).

**0. Введение.**

**0.1. Актуальность темы.**

**Современная компьютерная графика сталкивается с проблемой алиасинга - артефактов в виде “зубчатых” краёв объектов, возникающих из-за дискретизации непрерывных сигналов при рендеринге. Эти артефакты особенно заметны в динамических сценах (игры, VR/AR) и при выводе изображений на высококонтрастные экраны. Традиционные методы алиасинга (MSAA, FXAA, TAA) достаточно эффективны, однако имеют существенные ограничения:**

* **Высокие вычислительные затраты (MSAA).**
* **Размытие мелких деталей (FXAA).**
* **Невозможность устранения временных артефактов (мерцание в TAA).**

**Нейросетевые методы, в свою очередь, предлагают революционное решение, объединяя преимущества глубокого обучения и компьютерного зрения. Они способны:**

* **Адаптивно обрабатывать сложные сцены, сохраняя детализацию.**
* **Устранять артефакты как в статичных изображениях, так и в видео.**
* **Интегрироваться в реальные движки (например, NVIDIA DLSS 3.0).**

**0.2 Цель работы.**

**Цель работы - исследование нейросетевых методов антиалиасинга, разработка модели на основе свёрточных сетей и её сравнение с классическими алгоритмами.**

**0.3 Задачи.**

1. **Анализ архитектур CNN, GAN и трансформеров для задачи сглаживания.**
2. **Создание синтетического датасета пар изображений (с алиасингом и без).**
3. **Обучение модели с использованием функций потерь.**
4. **Экспериментальная оценка качества через метрики PSNR, SSIM и визуальное сравнение.**

**1. Теоретические основы антиалиасинга.**

**1.1 Понятие сглаживания.**

**Алиасинг - это эффект искажения изображения, возникающий при недостаточном разрешении или некорректной обработке данных. Визуально он проявляется в виде “зубчатых” краёв объектов, мерцания текстур или искажений тонких линий при рендеринге цифорвых изображений. Другими словами,** это явление, возникающее при обработке сигналов, в частности при цифровой обработке сигналов (DSP), когда непрерывный сигнал дискретизируется с частотой, которая слишком низкая для точного представления исходного сигнала. В цифровой связи наложение спектров возникает из-за ошибки измерения сигнала из-за неправильной частоты дискретизации.

Наложение спектров в основном возникает из-за некачественной дискретизации. Если частота дискретизации слишком низкая, может возникнуть наложение спектров. Наложение спектров также может возникать, когда сигналы дискретизируются недостаточно быстро, но в основном оно возникает, когда частота дискретизации слишком низкая для точного восстановления исходного сигнала или когда высокие частоты в сигнале превышают половину частоты дискретизации.

Основные причины возникновения:

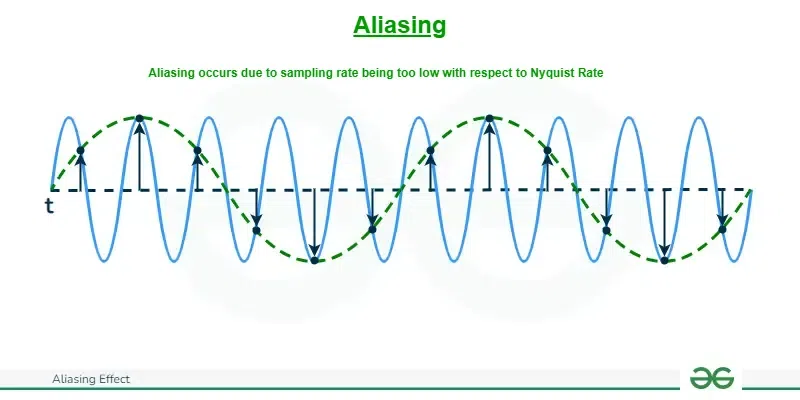
* Нарушение теоремы Котельникова (Найквиста-Шеннона):

Если частота дискретизации (разрешение изображения) недостаточна для представления высокочастотных компонент сигнала (резких краёв, мелких деталей), возникает наложение спектров (*frequency aliasing*).

Частота Найквиста - это минимальная частота дискретизации, необходимая для точной дискретизации и восстановления сигнала без искажений. Она определяется как удвоенная максимальная частота, присутствующая в сигнале. Дискретизация с частотой ниже частоты Найквиста может привести к наложению спектров.

Теорема Найквиста гласит, что для восстановления сигнала частота дискретизации должна быть как минимум в два раза больше частоты дискретизируемого сигнала.

Fs ≥ 2 \* Fmax , где Fmax - максимальная частота в спектре исходного аналогового сигнала.



https://www.geeksforgeeks.org/aliasing-effect/

Геометрические искажения:

Геометрические искажения возникают при рендеринге объектов из-за дискретизации непрерывных геометрических форм в цифровое представление (пиксели или полигоны). Они могут проявляться, как ступенчатые края на диагональных или изогнутых линиях, искажение текстур на поверхностях сложной формы или как артефакты интерполяции при анимации или деформации объектов.

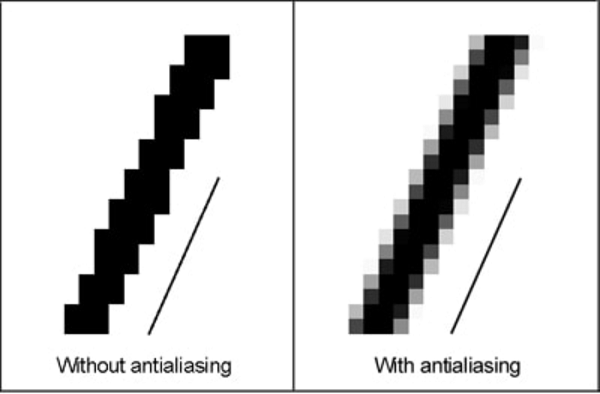
Основными причинами возникновения геометрических искажений могут служить: низкая полигональная плотность (то есть грубая сетка, например, модель сферы из 100 треугольников, она не может точно аппроксимировать гладкую поверхность), некорректная дискретизация (недостаточное разрешение экрана или текстуры для отображения мелких деталей), ошибки интерполяции (неточное вычисление нормалей или каких-либо внешних атрибутов между полигонами).

* Временной алиасинг:

Это ошибки в изображениях, возникающие в динамических сценах из-за дискретизации непрерывного движения в последовательности кадров. Он может проявляться, как мерцание (flickering) текстур или мелких объектов, “гребенчатые искажения” (strobing) на быстро движущихся объектах или как призрачные контуры (ghosting) при резких перемещениях камеры.

Причинами могут служить: низкая частота кадров, несоответствие частоты дискретизации и скорости движения (иначе говоря, нарушение теоремы Найквиста) или неточное прогнозирование движения (например, ошибки в расчёте позиций объектов или текстур между кадрами).

В компьютерных играх часто можно заметить, как тонкие провода или решётки мерцают при движении камеры, или, например, как трава в гоночных играх “размазывается”.



****

**1.2 Математические основы.**

Ранее мы рассматривали теорему Найквиста-Шеннона, как одну из основных причин возникновение алиасинга. Если условие теоремы нарушается, возникает наложение спектров, что визуально проявляется, как “зубчатые края”.

Изображение можно представить, как сумму синусоидальных компонент разной частоты, используя преобразования Фурье:

I(x, y) = Σk(Ak \* sin(2**π(fxx + fyy) +** *ϕk*)), где:

fx, fy - пространственные частоты.

Перед дискретизацией применяется фильтр низких частот (low-pass filter), подавляющий компоненты, в которых f > Fнайквиста:

Ifiltered(x, y) = I(x, y) \* Gσ(x, y), где:

Gσ - гауссово ядро с дисперсией σ.

**Нейросетевая модель решает задачу регрессии: предсказание “идеального” Iout изображения на основе входного Iin с артефактами:**

**Iout = fθ(Iin), где:**

**fθ - нейросеть с параметрами θ**

**Параметры выбираются для оптимизации функции потерь L:**

**θ\* = argminθL(Iout, Itarget), где:**

**Itarget - эталонное изображение без алиасинга.**

**Функция потерь, в свою очередь, может иметь разный вид, каждый из которых имеет свои преимущества и недостатки:**

**Среднеквадратичная ошибка между пикселями:**

**LMSE = 1/N \* (**Σi=1N ∥Iout(i) - Itarget(i)∥22**) .**

**Данную норму достаточно просто вычислять, однако она игнорирует семантику изображения, в следствие чего могут возникать размытия.**

**Сравнение признаков, использованных предобученной сетью (perceptual loss), например, VGG-19:**

**Lperc = 1/(CjHjWj) \*** Σc,h,w (∥*ϕj(Iout)c,h,w - ϕj(Itarget)c,h,w*∥22), где *ϕj - активации j-го слоя в VGG.*

***Данный вариант хорошо сохраняет текстуры и контуры.***

* **Adversial loss (для GAN):**

**Данные вариант поощряет генератор создавать реалистичные изображения:**

**Ladv = E[logD(Itarget)] + E[log(1 - D(Iout))], где:  
D - дискриминатор.**

* **Данные функции потерь можно комбинировать, таким образом получив итоговую:**

**L =** *λ1LMSE + λ2Lperc + λ3Ladv.*

Безусловно, существует множество других вариантов функций потерь и её минимизации, например, достаточно часто используется стохастический градиентный спуск или его оптимизации (Adam и другие).

**1.3 Классические методы.**

**Для борьбы с алиасингом разработаны различные алгоритмы. Их можно разделить на две категории: сглаживание до рендеринга (то есть до преобразования компьютерной модели или сцены в изображение или анимацию) и постобработка.**

**1.3.1 MSAA (Multisample Anti-Aliasing).**

MSAA (Multisample Anti-Aliasing) — это один из методов сглаживания, используемый в компьютерной графике для уменьшения эффекта "лесенки" на краях объектов. Этот эффект возникает из-за того, что пиксели на экране имеют фиксированные размеры и формы, что приводит к неровным краям на диагональных и кривых линиях. MSAA помогает сделать эти края более плавными, улучшая визуальное качество изображения. Сглаживание является важным аспектом в компьютерной графике, так как оно напрямую влияет на восприятие изображения пользователем. Без сглаживания, даже самые детализированные и реалистичные сцены могут выглядеть грубо и неестественно.

Принцип работы:

MSAA работает путем выборочного увеличения разрешения на краях объектов. Вместо того чтобы увеличивать разрешение всего изображения, что требует значительных вычислительных ресурсов, MSAA увеличивает разрешение только на краях, где это наиболее заметно. Это позволяет достичь высокого качества изображения без значительного увеличения нагрузки на систему.

Как это происходит:

Выборка нескольких точек: Для каждого пикселя на краю объекта выбирается несколько точек (субпикселей). Количество этих точек зависит от уровня MSAA (например, 2x, 4x, 8x). Чем больше точек выбирается, тем более плавным будет переход между цветами на краях объектов.

Рендеринг субпикселей: Эти субпиксели рендерятся отдельно, учитывая цвета и глубину сцены. Это позволяет более точно определить, какие части сцены находятся ближе к камере и какие дальше.

Среднее значение: Затем цвета всех субпикселей усредняются для получения конечного цвета пикселя. Это позволяет сгладить переходы между цветами, уменьшая эффект "лесенки". Усреднение цветов субпикселей помогает создать иллюзию более высокого разрешения на краях объектов.

Головные боли, связанные с реализацией:

Мы полагаемся на аппаратное обеспечение для сглаживания, что, очевидно, приводит к проблеме, заключающейся в том, что пользовательское оборудование может не поддерживать то, что нам нужно. Шаблоны выборки, которые использует MSAA, также могут делать то, чего мы не ожидаем. В зависимости от того, что делает то или иное оборудование, можно увидеть, что шаги прозрачности контура какого-то объекта отображаются «в неправильном порядке».

Преимущества и недостатки:

Преимущества:

Улучшенное качество изображения: MSAA значительно улучшает визуальное качество, делая края объектов более плавными. Это особенно заметно в играх с высоким уровнем детализации, где каждая мелочь имеет значение.

Эффективность: MSAA более эффективен по сравнению с другими методами сглаживания, такими как SSAA (Super-Sample Anti-Aliasing), так как он не требует увеличения разрешения всего изображения. Это позволяет достичь хорошего качества изображения без значительного увеличения нагрузки на систему.

Недостатки:

Производительность: MSAA требует дополнительных вычислительных ресурсов, что может снизить производительность, особенно на старых или менее мощных видеокартах. Это может привести к снижению частоты кадров и ухудшению игрового опыта.

Не сглаживает все элементы: MSAA не всегда эффективно сглаживает текстуры и шейдеры, что может привести к оставшимся артефактам. Это особенно заметно в играх с большим количеством текстур и сложными шейдерами.

MSAA — это мощный инструмент для улучшения визуального качества игр. Понимание его работы и правильная настройка помогают достичь наилучшего игрового опыта. Используя MSAA, можно значительно улучшить качество изображения и сделать игровой процесс более приятным и реалистичным.

**1.3.2 FXAA (Fast Approximate Anti-Aliasing).**

FXAA (Fast Approximate Anti-Aliasing) — это один из методов сглаживания, разработанный NVIDIA. В отличие от традиционных методов, таких как MSAA (Multisample Anti-Aliasing), FXAA работает на этапе постобработки изображения, что делает его быстрым и эффективным.

FXAA был разработан в 2011 году инженером NVIDIA Тимоти Лотц. Основной целью было создание метода антиалиасинга, который бы не требовал значительных ресурсов и мог бы применяться в реальном времени. С тех пор FXAA стал одним из самых популярных методов сглаживания в игровой индустрии.

Основные концепции:

Постобработка: FXAA применяется к уже отрендеренному изображению, а не к геометрии сцены. Это позволяет избежать сложных вычислений на этапе рендеринга и значительно ускоряет процесс.

Быстрота: Метод разработан для минимального влияния на производительность, что делает его подходящим для реального времени. Это особенно важно для игр, где каждая миллисекунда на счету.

Простота: FXAA не требует сложных настроек и легко интегрируется в существующие графические движки. Это делает его доступным для широкого круга разработчиков, независимо от уровня их опыта.

Пошаговый принцип работы:

Шаг 1: Определение краев

Первый шаг в FXAA — это определение краев объектов на изображении. Для этого используется метод поиска контрастных переходов между пикселями. Если разница в цвете или яркости между соседними пикселями превышает определенный порог, эти пиксели считаются частью края. Этот процесс называется "edge detection" и является ключевым для эффективного сглаживания.

Шаг 2: Размытие краев

После определения краев, FXAA применяет размытие к этим областям. Размытие происходит только вдоль краев, что помогает сгладить "зубчатые" линии. Этот шаг выполняется с использованием специальных фильтров, которые учитывают направление и интенсивность края. Важно отметить, что размытие не затрагивает всю сцену, а только те области, где были обнаружены края.

Шаг 3: Постобработка

На последнем этапе FXAA применяет финальные корректировки к изображению, чтобы убедиться, что сглаживание выглядит естественно. Это включает в себя дополнительные фильтры и корректировки яркости и контраста. Эти финальные шаги помогают устранить любые артефакты, которые могли возникнуть в процессе сглаживания, и делают изображение более приятным для глаза.

В некоторых реализациях FXAA могут быть добавлены дополнительные шаги для улучшения качества изображения. Например, могут использоваться более сложные фильтры для определения краев или дополнительные этапы постобработки для улучшения детализации. Эти оптимизации могут варьироваться в зависимости от конкретных требований проекта и аппаратных возможностей.

Преимущества и недостатки:

Преимущества

Высокая производительность: FXAA минимально влияет на FPS (кадры в секунду), что делает его идеальным для игр. Это особенно важно для многопользовательских игр, где каждая миллисекунда может быть решающей.

Простота интеграции: Легко добавляется в существующие графические движки. Это позволяет разработчикам быстро внедрить FXAA в свои проекты без необходимости значительных изменений в коде.

Универсальность: Работает на любом оборудовании, поддерживающем шейдеры. Это делает FXAA доступным для широкого круга пользователей, независимо от их аппаратных возможностей.

Недостатки

Потеря детализации: В некоторых случаях может размывать мелкие детали изображения. Это особенно заметно на текстурах высокого разрешения и в сценах с множеством мелких объектов.

Не идеален для всех сцен: В сложных сценах с множеством мелких объектов FXAA может не дать желаемого результата. В таких случаях могут потребоваться дополнительные методы антиалиасинга для достижения оптимального качества.

Зависимость от разрешения: Эффективность FXAA может снижаться на экранах с высоким разрешением. Это связано с тем, что на высоких разрешениях "зубчатые" края становятся менее заметными, и эффект сглаживания может быть менее выраженным.

FXAA — это мощный и эффективный метод сглаживания, который идеально подходит для реального времени. Он обеспечивает высокое качество изображения при минимальном влиянии на производительность, что делает его популярным выбором среди разработчиков игр и приложений. Благодаря своей простоте и универсальности, FXAA продолжает оставаться одним из самых востребованных методов антиалиасинга в индустрии компьютерной графики.

**1.3.3 TAA (Temporal Anti-Aliasing).**

TAA (Temporal Anti-Aliasing) — это современный метод антиалиасинга, который использует временную информацию для улучшения качества изображения. В отличие от других методов, TAA анализирует несколько кадров и использует данные из предыдущих кадров для сглаживания текущего. Это позволяет значительно улучшить качество изображения без значительного увеличения нагрузки на систему.

TAA особенно эффективен в динамических сценах, где объекты и камера постоянно движутся. Этот метод позволяет учитывать изменения в сцене и адаптироваться к ним, что делает его более гибким и эффективным по сравнению с другими методами антиалиасинга. TAA также может помочь уменьшить артефакты, такие как мерцание и шум, которые могут возникать при использовании других методов.

Принцип работы:

TAA работает на основе нескольких ключевых принципов:

Временная интеграция

TAA использует данные из нескольких предыдущих кадров для создания текущего кадра. Это позволяет сгладить края объектов, используя информацию о движении и изменениях в сцене. Временная интеграция позволяет учитывать изменения в освещении, текстурах и других элементах сцены, что делает сглаживание более точным и эффективным.

Этот процесс также помогает уменьшить артефакты, такие как мерцание и шум, которые могут возникать при использовании других методов антиалиасинга. Временная интеграция позволяет TAA адаптироваться к изменениям в сцене и обеспечивать более стабильное и качественное изображение.

Буфер истории

Для хранения данных из предыдущих кадров используется специальный буфер истории. Этот буфер содержит информацию о цвете и глубине пикселей, что позволяет точно определить, как изменились объекты с течением времени. Буфер истории играет ключевую роль в работе TAA, обеспечивая доступ к необходимой информации для временной интеграции.

Буфер истории также помогает уменьшить нагрузку на систему, так как позволяет использовать данные из предыдущих кадров вместо того, чтобы заново рассчитывать все параметры для каждого кадра. Это делает TAA более эффективным и производительным методом антиалиасинга.

Компенсация движения

Одной из ключевых особенностей TAA является компенсация движения. Этот процесс позволяет учитывать движение объектов и камеры, чтобы избежать размытия и других артефактов. Для этого используется информация о векторе движения, который рассчитывается для каждого пикселя.

Компенсация движения позволяет TAA адаптироваться к изменениям в сцене и обеспечивать более точное и качественное сглаживание. Этот процесс также помогает уменьшить артефакты, такие как размытие и мерцание, которые могут возникать при использовании других методов антиалиасинга.

TAA (Temporal Anti-Aliasing) — это мощный и эффективный метод антиалиасинга, который использует временную информацию для улучшения качества изображения. Он предлагает баланс между качеством и производительностью, что делает его отличным выбором для современных видеоигр и других интерактивных приложений.

Используя временную интеграцию, буфер истории и компенсацию движения, TAA обеспечивает более точное и качественное сглаживание по сравнению с другими методами антиалиасинга. Это делает его предпочтительным выбором для разработчиков и пользователей, которые стремятся к высокому качеству изображения и стабильной производительности.

В конечном итоге, выбор метода антиалиасинга зависит от конкретных требований и ограничений системы. Однако, TAA предлагает уникальные преимущества, которые делают его отличным выбором для широкого спектра приложений и устройств.

Краткое сравнение классических методов:

|  |  |  |  |
| --- | --- | --- | --- |
| **Метод** | **Качество** | **Производительность** | **Устранение временного алиасинга** |
| MSAA | Высокое | Низкая | Нет |
| FXAA | Среднее | Высокая | Нет |
| TAA | Высокое | Средняя | Да |

**Классические методы антиалиасинга достаточно эффективны, однако имеют свои ограничения по производительности, качеству или устранению артефактов. Помимо них существуют также нейросетевые методы, которые помогают решать многие проблемы, которые есть в классических методах.**

**2. Обзор современных нейросетевых подходов.**

**2.1 Архитектуры CNN.**

Свёрточные нейросети стали основой для многих методов антиалиасинга благодаря их способности анализировать локальные и глобальные паттерны в изображениях.

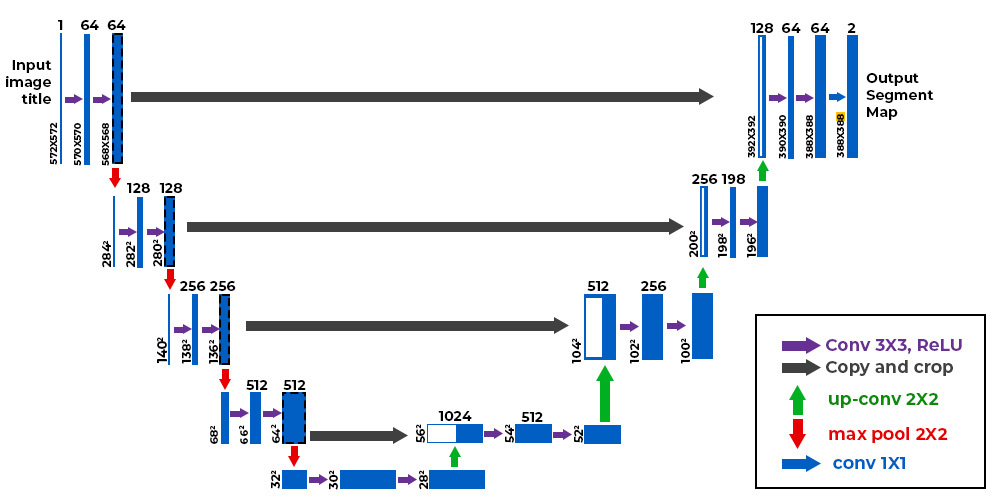
**2.1.1 U-Net.**

U-Net — это широко используемая архитектура глубокого обучения, впервые представленная в статье «U-Net: Convolutional Networks for Biomedical Image Segmentation». Основной целью этой архитектуры было решение проблемы ограниченных аннотированных данных в медицинской сфере. Эта сеть была разработана для эффективного использования меньшего объема данных при сохранении скорости и точности.

Архитектура:

Архитектура U-Net уникальна тем, что состоит из сжимающегося пути и расширяющегося пути. Сжимающийся путь содержит слои кодировщика, которые захватывают контекстную информацию и уменьшают пространственное разрешение ввода, в то время как расширяющийся путь содержит слои декодера, которые декодируют закодированные данные и используют информацию из сжимающегося пути через пропускные соединения для генерации карты сегментации.

Сжимающий путь в U-Net отвечает за идентификацию соответствующих признаков на входном изображении. Слои кодировщика выполняют сверточные операции, которые уменьшают пространственное разрешение карт признаков, увеличивая их глубину, тем самым захватывая все более абстрактные представления входных данных. Этот сжимающий путь похож на слои прямой связи в других сверточных нейронных сетях. С другой стороны, расширяющийся путь работает над декодированием закодированных данных и определением местоположения признаков, сохраняя пространственное разрешение входных данных. Слои декодера в расширяющемся пути повышают дискретизацию карт признаков, также выполняя сверточные операции. Пропускные соединения из сжимающего пути помогают сохранить пространственную информацию, потерянную в сжимающем пути, что помогает слоям декодера точнее определять местоположение признаков.



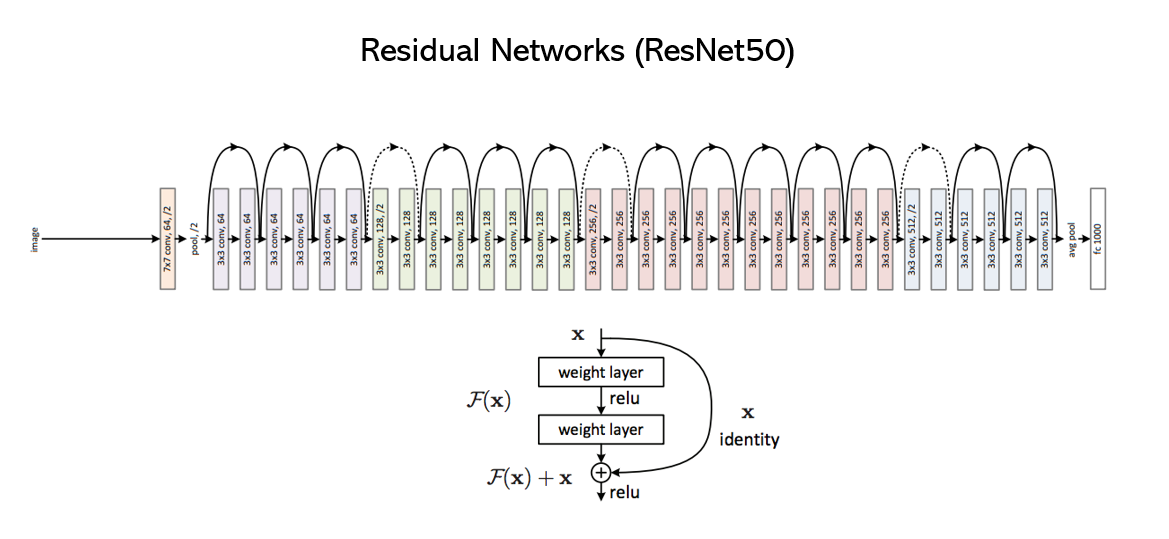
Данный рисунок иллюстрирует, как сеть U-Net преобразует входное изображение в оттенках серого размером 572×572×1 в двоичную сегментированную выходную карту размером 388×388×2. Мы можем заметить, что выходной размер меньше входного размера, поскольку не используется заполнение. Однако, если мы используем заполнение, мы можем сохранить входной размер. Во время сжимающегося пути входное изображение постепенно уменьшается по высоте и ширине, но увеличивается по количеству каналов. Это увеличение каналов позволяет сети захватывать высокоуровневые признаки по мере продвижения по пути. В узком месте выполняется окончательная операция свертки для создания карты признаков в форме 30×30×1024. Затем расширяющийся путь берет карту признаков из узкого места и преобразует ее обратно в изображение того же размера, что и исходный вход. Это делается с помощью слоев повышающей дискретизации, которые увеличивают пространственное разрешение карты признаков, одновременно уменьшая количество каналов. Связи пропуска из сокращающегося пути используются для того, чтобы помочь слоям декодера обнаружить и уточнить особенности изображения. Наконец, каждый пиксель в выходном изображении представляет собой метку, которая соответствует определенному объекту или классу во входном изображении. В этом случае выходная карта представляет собой двоичную карту сегментации, где каждый пиксель представляет область переднего плана или фона.

Данная модель часто применяется для восстановления деталей и сглаживания краёв через многоуровневый анализ. Например, модель DeepAA (некоторый обобщённый пример нейросетевого подхода к антиалиасингу на базе U-Net или, например, FSRCNN, адаптированных для задачи сглаживания. Это может быть NVIDIA DLSS или Deep Anti-Aliasing) использует U-Net для подавления алиасинга в рендерах 3D-сцен.

**2.1.2 ResNet (Residual Networks).**

ResNet (Residual Neural Network) — это архитектура нейронных сетей, которая совершила революцию в глубоком обучении. Главная идея ResNet — использование остаточных (skip) соединений. Они позволяют нейросети быть очень глубокой и избегать проблемы исчезающего градиента, когда сигналы, передаваемые между слоями, затухают до нуля.

В традиционных нейронных сетях с увеличением количества слоев обучение становится сложнее: градиенты либо исчезают, либо становятся нестабильными. ResNet решает эту проблему с помощью остаточных блоков, благодаря которым информация проходит напрямую через несколько слоев, минуя некоторые вычисления. Это не только ускоряет обучение, но и позволяет создавать сверхглубокие модели: ResNet-50, ResNet-101 и ResNet-152 . Благодаря своей уникальной структуре ResNet стала основой для множества современных моделей в компьютерном зрении, распознавании изображений, классификации и других задачах искусственного интеллекта.

****

**Как устроена архитектура:**

Архитектура ResNet (Residual Network) произвела революцию в мире глубокого обучения, решив одну из главных проблем — исчезающие градиенты в очень глубоких нейронных сетях. Вот ее основные компоненты:

* **Остаточные блоки (Residual Blocks). Именно они помогают**

**нейросети передавать информацию через несколько слоев, минуя некоторые из них. Остаточные блоки — это своеобразные обходные пути. Нужно отметить, что ResNet не соединяет слои последовательно, а использует связи skip connections. Благодаря им информации «перепрыгивает» через слои, сохранялась практически неизменной, даже если модель состоит из сотен слоев.**

* **Сверточные слои и функции активации. Каждый остаточный блок включает сверточные слои с ядрами 3×3 и функции активации ReLU. Эти элементы добавляют нелинейность, позволяя модели улавливать сложные закономерности в данных.**
* **Нормализация и оптимизация. Для стабилизации обучения ResNet использует Batch Normalization. Этот метод нормализует выходы каждого слоя, ускоряет обучение и улучшает стабильность нейросети.**
* **Масштабируемость. ResNet легко адаптируется под разные задачи. Например, ResNet-18 содержит 18 слоев, ResNet-50 — 50, а ResNet-152 — 152 слоя. Это позволяет выбирать модель в зависимости от сложности задачи и доступных вычислительных ресурсов.**

Благодаря этим особенностям ResNet стала стандартом для задач классификации, сегментации и детектирования объектов на изображениях.

ResNet широко используется в классификации, генерации или в обработке изображений, в обработке и сегментации объектов, в анализе видео и во многих других задачах. ResNet — это не просто инструмент, а фундамент, на котором строятся многие современные технологии. Рекуррентные связи и многослойная архитектура делают эту сеть универсальной для решения самых разных задач. А благодаря своей способности оптимизировать процесс обучения и сопоставлять сложные данные, ResNet на сегодняшний день остается одной из самых популярных архитектур в глубоком обучении.

**2.1.3 DnCnn (Denoising CNN).**

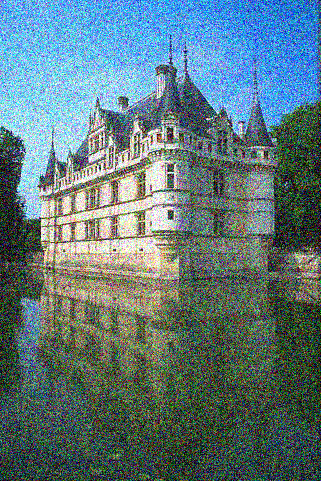
Одним из популярных алгоритмов, использующих нейронные сети для устранения шумов, является DnCNN (Denoising Convolutional Neural Network). Этот метод был разработан для эффективного удаления шумов из изображений, включая гауссовский шум, соль и перец, и другие виды артефактов.

Принцип работы DnCNN основан на использовании глубокой сверточной нейронной сети, обученной на парах зашумленных и чистых изображений. Сеть состоит из нескольких слоев свертки, активации и пакетной нормализации, что позволяет ей изучить сложные зависимости между зашумленными и чистыми изображениями.

В процессе обучения DnCNN находит и учитывает особенности шумовых артефактов, что позволяет ему эффективно удалять шумы и восстанавливать изображения с высоким качеством. Такой подход позволяет значительно улучшить качество изображений (снижение MSE на 15-20% по сравнению с другими методами), что особенно важно в областях медицины, видеосъёмки и компьютерного зрения.

Использование DnCNN и подобных методов позволяет минимизировать воздействие шумов и артефактов на качество изображений, что делает их более чистыми и информативными. Такие алгоритмы играют важную роль в современных технологиях и находят применение в различных отраслях промышленности и науки.

Пример визуального результата применения DnCnn:



<https://github.com/cszn/DnCNN>

Свёрточные нейронные сети широко используются на данный момент. Одним из примеров может являться версия технологии сглаживания NVIDIA DLSS 2.0, использующая CNN для масштабирования изображения с низкого разрешения до 4K, совмещая антиалиасинг и супер-резолюцию - технологию повышения чёткости изображения. DLSS 2.0 очень просто интегрируется в игру, сильно улучшает качество изображения, использует одну сеть для всех игр, а также имеет различные настраиваемые параметры. Все эти качества значительно улучшают игровой процесс, а также качество графического интерфейса.

**2.2 Генеративно-состязательные сети (GAN).**

**GAN (***Generative adversarial network***) - это алгоритм машинного обучения (без учителя), построенный на комбинации из двух нейронных сетей, одна из которых генерирует образцы, а другая старается отличить “подлинные” образцы от неправильных (G - генеративная модель, D - дискриминативная модель). *Генеративно-состязательную сеть описал Ян Гудфеллоу из компании Google в 2014 году.***.

**GAN позволяют генерировать изображения с высокой детализацией, что очень важно для устранения сложных артефактов. Рассмотрим некоторые основные архитектуры:**

**2.2.1 ESRGAN (Enhanced Super Resolution GAN).**

В 2018 году специалисты из КНР **XintaoWang и его коллеги** создали нейросеть, которая улучшает качество изображения в играх. Технологию назвали ESRGAN (EnhancedSuperResolutionGenerativeAdverserialNetwork - Сеть, генерирующая улучшенное разрешение). Эта нейросеть является улучшенной версией GAN (GenerativeAdversarialNetwork), созданной специально для задач суперразрешения. ESRGAN сочетает в себе высокую точность восстановления текстур и реалистичное воспроизведение деталей, что делает её одной из самых передовых технологий в этой области.

ESRGAN (Enhanced Super-Resolution Generative Adversarial Network) — это нейросеть, предназначенная для выполнения процедур суперразрешения (Super-Resolution). Она позволяет увеличить разрешение изображения и при этом восстанавливать потерянные мелкие детали, улучшая общее качество изображения. ESRGAN применяется в самых разных областях: от улучшения снимков архивных источников до создания современных визуальных эффектов для видео- и медиапроектов. Эта технология активно используется для обработки фотографий и видео контента, повышения детализации медицинских снимков, а также улучшения качества изображений для космических исследований и научных целей (модель лучше сохраняет текстуры в сравнении с CNN). Создание ESRGAN положило начало новому этапу в развитии технологий суперразрешения, что позволяет значительно улучшить качество изображений и расширить их применение в различных сферах.

**2.2.2 StyleGan-Ada.**

StyleGAN2-ADA, разработанный NVIDIA, это мощный инструмент в мире игровых нейросетей. Он предназначен для генерации фотореалистичных изображений, включая лица и текстуры, что открывает двери к автоматической генерации контента. Его важность для игровой индустрии обусловлена возможностью значительно ускорить процесс разработки игровых миров и текстурирования игровых моделей, снижая затраты и повышая качество контента. Бесплатные ресурсы и инструменты делают его доступным для широкого круга разработчиков.

Традиционные методы создания игрового контента, особенно генерации игровых миров и текстурирования игровых моделей, часто требуют значительных временных и финансовых затрат. Художникам и дизайнерам приходится вручную создавать каждый элемент, что ограничивает масштабируемость и разнообразие контента. Например, по статистике, разработка текстуры для одного игрового персонажа может занимать от нескольких дней до нескольких недель. StyleGAN2-ADA применение в играх позволяет преодолеть эти ограничения, предлагая процедурное моделирование и автоматическую генерацию контента, что значительно сокращает время разработки и повышает эффективность.

**Принцип работы:**

**StyleGAN2-ADA использует генеративно-состязательную сеть (GAN) с генератором и дискриминатором. Генератор создает изображения, а дискриминатор оценивает их реалистичность. Адаптивная дискриминаторная аугментация (ADA) улучшает обучение, особенно при ограниченных данных, что важно для генерации игровых миров. Генератор StyleGAN2-ADA создает изображения, а дискриминатор пытается отличить их от реальных. Этот процесс (состязание) улучшает качество генерации. В контексте генерации игровых миров это позволяет создавать более реалистичные и детализированные текстуры, необходимые для игровых движков и нейросетей.**

Одним из преимуществ StyleGan является возможность обучения на маленьких датасетах благодаря адаптивной аугментации, а также возможность сглаживания анимационных рендеров с сохранением стиля. ADA – это метод, который автоматически регулирует количество аугментаций данных во время обучения. Это особенно полезно, когда данных мало, что часто встречается при обучении StyleGAN2-ADA для специфических игровых стилей. Преимущество – улучшенное качество генерации и стабильность обучения, что критично для адаптивной генерации текстур.

Процесс обучения:

При выборе наборов данных для StyleGAN2-ADA обучения важно учитывать их репрезентативность и качество. Данные должны отражать разнообразие текстур, которые необхожимо генерировать. Например, для реалистичных ландшафтов нужны фотографии гор, лесов, воды. Чем выше качество данных, тем лучше будет результат автоматической генерации контента.

**Методы обучения и оптимизации: Улучшение результатов.**

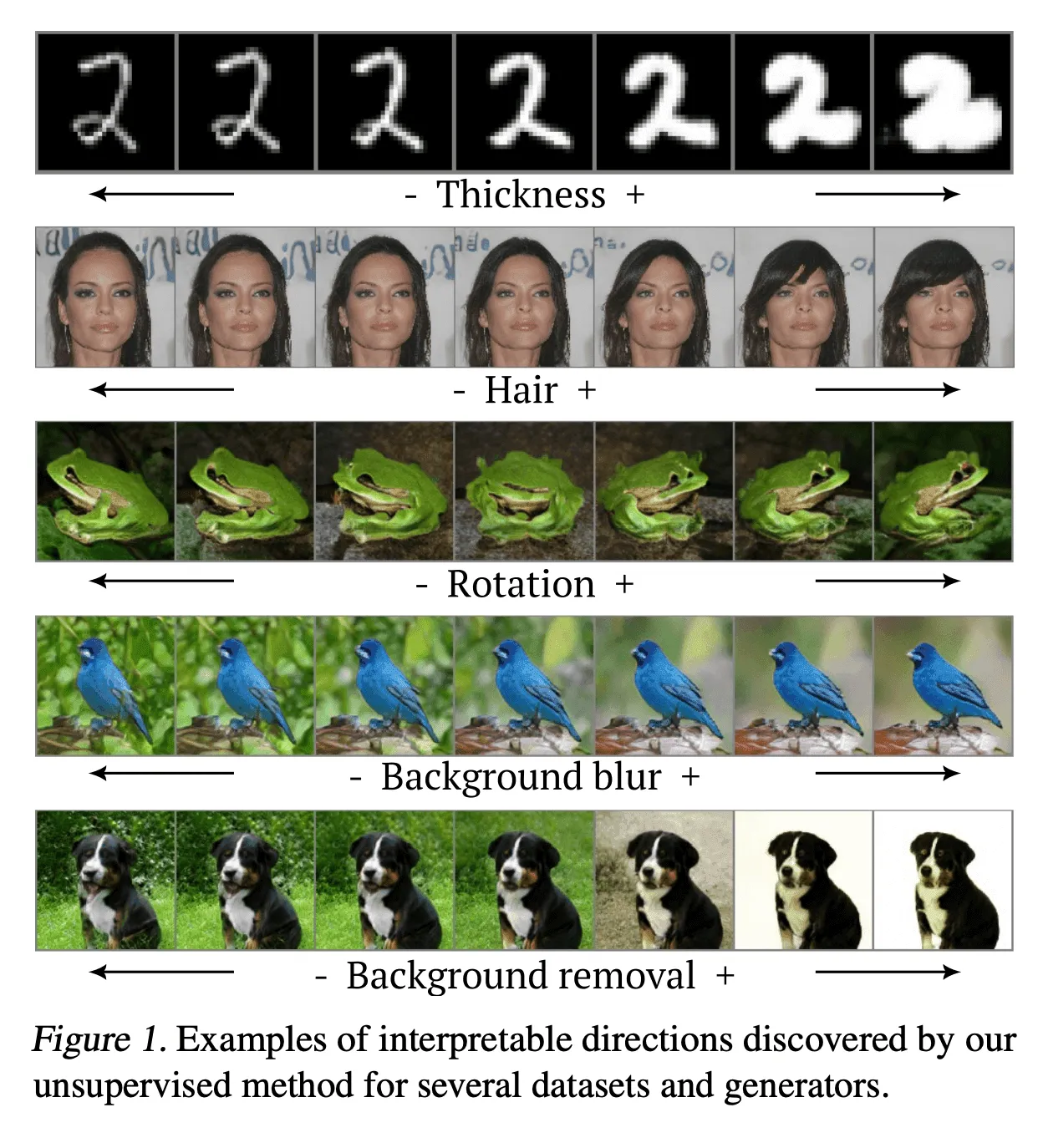
**Для StyleGAN2-ADA обучения используют методы GAN с ADA. Оптимизация включает настройку гиперпараметров, выбор оптимальных функций потерь и использование техник регуляризации. Эксперименты показывают, что правильная оптимизация может значительно улучшить качество сгенерированных текстур, делая их более реалистичными и подходящими для игровых движков и нейросетей.**

**Преимущества использования StyleGAN2-ADA в игровой индустрии:**

**StyleGAN2-ADA применение в играх предоставляет ключевые преимущества: ускорение разработки игровых миров, снижение затрат, увеличение разнообразия контента, адаптивная генерация текстур и возможность создания уникального визуального стиля. Это инструмент, который делает искусственный интеллект в играх доступным и эффективным для широкого круга разработчиков, определяя будущее игровой графики.**

**Основные проблемы GAN:**

**GAN-модели - очень мощный инструмент для улучшения графики, однако у них есть и свои недостатки.**

* **Нестабильность обучения. Это связано с самим принципом работы модели - двумя конкурирующими сетями. Одна из них может начать доминировать над другой, что и приводит к нестабильности обучения.**
* **Режимы отказа и коллапса. В первом случае генератор зацикливается на создании конкретных примеров из области данных задач вместо нахождения “точки равновесия”. Во втором же случае несколько вводов в модель генератора приводят к генерации одного и того же выхода. Это одна из самых сложных проблем при обучении GAN.**
* **Плохое качество или наличие артефактов. Это вызвано ограничениями, связанными с мощностью генератора и несовершенством методов обучения.**
* **Высокая чувствительность к гиперпараметрам, особенно learning rate и расписание (количество обновлений дискриминатора на одно**
* **обновление генератора).**

**<https://proceedings.mlr.press/v119/voynov20a/voynov20a.pdf>**

**2.3 Гибридные подходы.**

**Гибридные подходы комбинируют различные варинаты моделей, например, использование CNN и трансформеров или же attention-механизмов. Изначально разработанные трансформеры широко использовались в NLP, однако нашли своё применение и в задачах компьютерного зрения. Рассмотрим некоторые варианты комбинированных моделей.**

**2.3.1 Vision Transformer (ViT).**

Vision Transformer (ViT) — это инновационная архитектура глубокого обучения, разработанная для обработки визуальных данных с использованием той же архитектуры преобразователя, которая произвела революцию в обработке естественного языка (NLP). В отличие от сверточных нейронных сетей (CNN), которые используют свертки для захвата локальных пространственных особенностей, Vision Transformers используют self-attention механизмы для моделирования глобальных отношений между фрагментами изображения. Эта архитектура продемонстрировала передовую производительность во многих задачах компьютерного зрения, таких как классификация изображений, обнаружение объектов и сегментация.

**Обзор архитектуры ViT:**

**Vision Transformer основывается на архитектуре transformer, изначально представленной Vaswani et al. в 2017 году для задач обработки естественного языка. Transformers очень эффективны при обработке последовательных данных, используя внутреннее внимание для моделирования зависимостей между различными частями входных данных. Vision Transformers применяют эту архитектуру к данным изображений, рассматривая изображение как последовательность участков вместо сетки пикселей.**

**Архитектура Vision Transformer состоит из нескольких ключевых этапов:**

**1. Вставка и встраивание изображений - преобразование изображений в последовательность фрагментов, аналогичных токенам в модели обработки естественного языка. Три основных этапа: разделение патча --> сглаживание патчей --> встраивание патча.**

**2. Позиционное кодирование. Трансформаторы по своей сути не захватывают пространственный порядок входных последовательностей. Поскольку патчи обрабатываются как независимые токены, необходимо ввести позиционные кодировки для сохранения пространственной структуры исходного изображения.** Основные этапы: позиционное встраивание --> изученное и фиксированное позиционное кодирование.

**3. Трансформаторный энкодер. После того, как патчи внедрены и дополнены позиционной информацией, они проходят через стек слоев преобразователя-кодера . Эти слои состоят из двух основных компонентов: Multi-Head Self-Attention (MSA - многоголовое внутреннее внимание) и Feed-Forward Neural Network (FFN - сеть прямой связи).**

**Каждый слой преобразователя-кодера включает остаточные (пропускаемые) соединения и нормализацию слоев для стабилизации обучения и улучшения сходимости. Эти методы гарантируют, что более глубокие слои не потеряют важную информацию из более ранних слоев.**

**Наложение слоев кодировщика**

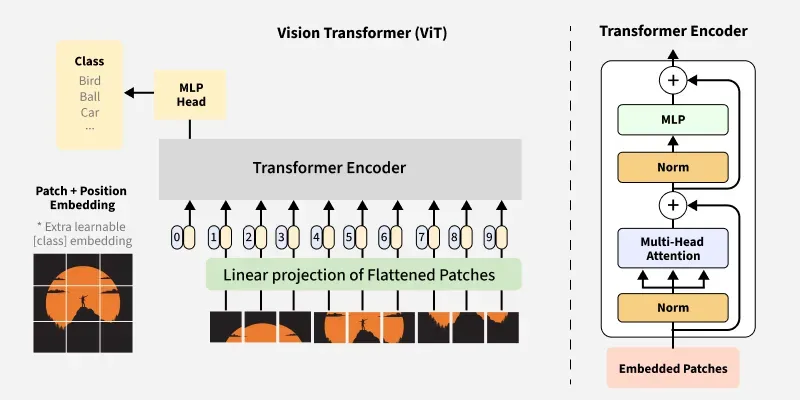
**Несколько слоев преобразователя-кодера (например, 12, 24 слоя) накладываются друг на друга. Каждый слой уточняет вложения патчей, позволяя модели строить более сложные и абстрактные представления изображения.**

**4. Руководитель классификации (руководитель MLP).**

После того, как преобразователи-кодеры обработают последовательность патчей и токен CLS - токен классификации, выходные данные, соответствующие токену CLS, используются для классификации.

Выход токена CLS подается в MLP, обычно состоящий из одного или двух полностью связанных слоев. Слой softmax применяется в конце MLP для задач классификации, предсказывая метку изображения.

Архитектура и работа ViT.



<https://www.geeksforgeeks.org/vision-transformer-vit-architecture/?ysclid=madjt2a8kf113549702>

Сравнение ViT со свёрточными сетями:

**- Локальное и глобальное внимание: CNN фиксируют локальные пространственные особенности с помощью сверток, тогда как ViT фиксируют глобальные отношения с помощью внутреннего внимания.**

**- Индуктивные смещения: CNN имеют встроенные индуктивные смещения, такие как локальность и инвариантность перевода, которые делают их эффективными в задачах с изображениями с меньшими наборами данных. Vision Transformers, с другой стороны, полагаются на данные, чтобы изучить эти шаблоны, что делает их более гибкими, но жадными до данных.**

**- Требования к обучающим данным: для достижения наилучших результатов Vision Transformers обычно требуются большие объемы обучающих данных, в то время как CNN могут хорошо работать даже с меньшими наборами данных.**

**Преимущества ViT:**

**- Глобальный контекст: ViT отлично справляются с фиксацией долгосрочных зависимостей между фрагментами изображений, что позволяет им лучше понимать глобальный контекст изображений.**

**- Масштабируемость: ViT хорошо масштабируются с большими наборами данных и более глубокими архитектурами, что делает их весьма эффективными для крупномасштабных задач машинного зрения.**

**- Гибкость: поскольку ViT не используют свертки, их можно легко адаптировать для решения различных задач и задач, выходящих за рамки классификации изображений, например, для анализа видео и обнаружения объектов.**

Недостатки:

**- Требования к данным: ViTs жадны до данных и обычно требуют больших наборов данных для эффективного обучения. Тонкая настройка ViTs, предварительно обученных на больших наборах данных (например, ImageNet), является обычной практикой, когда данные ограничены.**

**- Вычислительные ресурсы: Vision Transformers требуют значительной вычислительной мощности, особенно во время обучения, из-за квадратичной сложности механизма внутреннего внимания.**

Примером данной архитектуры может послужить Swim Transformer, использующий иерархическую структуру, широко использующийся для антиалиасинга видео.

Архитектура Vision Transformer знаменует собой значительный сдвиг в том, как обрабатываются визуальные данные, используя механизм собственного внимания трансформаторов. Она отлично справляется с захватом глобальных взаимосвязей между фрагментами изображения, предоставляя мощную альтернативу традиционным сверточным нейронным сетям. Хотя ViT очень гибкие и масштабируемые, для полной реализации своего потенциала им требуются большие наборы данных и вычислительные ресурсы. По мере продолжения исследований и разработок в этой области ожидается, что ViT будут играть все более важную роль в будущих задачах компьютерного зрения.

**2.3.2 CNN + transformer.**

В то время как чистые Vision Transformers (ViT) демонстрируют впечатляющие возможности в распознавании глобального контекста изображения, для их эффективного предварительного обучения часто требуются большие наборы данных из-за отсутствия присущих им индуктивных предположений о пространственной локальности, в чём преуспевают свёрточные нейронные сети (CNN). И наоборот, стандартные CNN, несмотря на их эффективность в изучении локальных закономерностей и пространственных иерархий, могут испытывать трудности с моделированием явных долгосрочных зависимостей на изображении. Это наблюдение естественным образом привело к разработке гибридных архитектур, которые стремятся объединить преимущества обоих подходов.

Гибридные модели CNN-трансформер представляют собой стратегию объединения свёрточных слоёв, которые хорошо справляются с извлечением низкоуровневых признаков и пространственных иерархий, с блоками трансформера, которые отлично подходят для моделирования глобальных взаимодействий между признаками. Основная идея заключается в том, чтобы позволить каждому компоненту делать то, что он делает лучше всего.

**Одна из распространённых и эффективных стратегий заключается в использовании CNN в первую очередь как мощного средства извлечения признаков на начальных этапах работы сети.**

**1. Начальные свёрточные слои: входное изображение сначала обрабатывается несколькими свёрточными слоями или усечённой стандартной основой CNN (например, начальными слоями ResNet). Эти слои выполняют начальное извлечение признаков, фиксируя края, текстуры и локальные мотивы, постепенно снижая пространственное разрешение и увеличивая глубину каналов. Это позволяет использовать пространственную индуктивную предвзятость свёрток, делая модель более эффективной с точки зрения использования данных, особенно на начальных слоях.**

**2. Слой перехода: на определённой глубине карта признаков, полученная на этапе CNN, преобразуется в последовательность, подходящую для ввода в трансформер. Это часто включает в себя:**

**Патчинг: как и в ViT, карта признаков может быть разделена на непересекающиеся или пересекающиеся патчи.**

**Сглаживание: Каждый участок сглаживается в вектор.**

**Линейная проекция: эти векторы линейно проецируются в размерность встраивания, ожидаемую трансформером. На этом этапе обычно добавляются векторы положения для сохранения пространственной информации.**

**3. Кодировщик-трансформер: последовательность векторных представлений затем обрабатывается одним или несколькими стандартными слоями кодировщика-трансформера. Эти слои используют многоканальное самовнимание для моделирования глобальных зависимостей между векторными представлениями, извлечёнными с помощью CNN. Механизм самовнимания позволяет модели оценивать важность различных областей векторного представления при построении окончательного представления.**

**4. Окончательная классификация/задача: выходные данные кодировщика Transformer (часто с использованием специального представления [CLS] токена или путём объединения выходных данных последовательности) подаются в окончательную классификационную модель (например, простую многокомпонентную регрессию) или модель для конкретной задачи (например, для обнаружения или сегментации).**

Этот подход выигрывает от способности CNN эффективно изучать надежные локальные функции, снижая нагрузку на трансформатор, который затем может сосредоточиться исключительно на рассуждениях о взаимосвязях между этими функциями более высокого уровня. Такие модели, как CvT (Convolutional Vision Transformer), явно включают свертки в механизмы токенизации и внимания Transformer, в то время как другие, такие как CoAtNet, стратегически размещают блоки convolutional и Transformer на разной глубине сети.

**Преимущества гибридных моделей:**

**- Улучшенная производительность: гибридные модели часто показывают лучшие результаты в различных тестах компьютерного зрения, потенциально превосходя по производительности как обычные свёрточные нейронные сети, так и обычные векторные нейронные сети, особенно когда объём обучающих данных не сопоставим с огромными наборами данных, полученными из интернета.**

**- Эффективность использования данных: благодаря сохранению индуктивного смещения свёрточных нейронных сетей, особенно на ранних слоях, гибридные модели часто сходятся быстрее и требуют меньше обучающих данных по сравнению с ViT, обученными с нуля.**

**- Гибкость: такое сочетание обеспечивает гибкость в проектировании, позволяя архитекторам балансировать между затратами на вычисления, количеством параметров и производительностью, выбирая глубину части CNN и сложность части Transformer.**

**- Использование предварительного обучения: можно легко внедрить хорошо зарекомендовавшие себя предварительно обученные основы CNN, что обеспечит надёжную инициализацию для части гибридной модели, отвечающей за извлечение признаков.**

Несмотря на свою эффективность, разработка гибридных моделей сопряжена со сложностями. Ключевые решения включают в себя точку перехода между этапами CNN и Transformer, метод преобразования карт признаков в последовательности (размер фрагмента, шаг), конкретную архитектуру используемых слоёв Transformer, а также способ кодирования и сохранения позиционной информации. Настройка этих архитектур требует тщательных экспериментов и учёта целевой задачи и характеристик набора данных.

Таким образом, гибридные модели CNN-трансформеров представляют собой практичный и эффективный способ объединить преимущества локальной обработки признаков, характерные для свёрточных нейронных сетей, с возможностями глобального контекстного моделирования трансформеров, что приводит к созданию надёжных и высокоэффективных систем компьютерного зрения. Они представляют собой привлекательное компромиссное решение, которое использует результаты десятилетий исследований свёрточных нейронных сетей и включает в себя достижения, связанные с механизмами внимания.

**2.4 Датасеты и метрики.**

**Для обучения нейросетевых моделей используются как реальные, так и синтетические данные.**

**2.4.1 Синтетические датасеты.**

**1. Рендеры 3D-сцен - парные изображения (с использованием алиасинга и без), сгенерированные в рендерах (например, Blender, Unreal Engine, Unity). Это позволяет добиться хорошего контроля над уровнем алиасинга (например, рендеринг с разным числом семплов на пиксель), а также достаточно точного соответствия между входными и целевыми данными.**

**Примером может служить Anti-Aliasing Benchmark Suite - сцены с объектами разной геометрической сложности (сферы, решётки, текстурированные поверхности) или же Synthetic Edge Dataset - изображения с диагональными линиями, кругами и многоугольниками для анализа краёв.**

**2. Искусственно добавленный алиасинг. Данный вариант имитирует алиасинг на “идеальных” изображениях путём даунсэмплинга с последующим апскейлингом без интерполяции или же с помощью добавления ступенчатых артефактов на края объектов. Один из примеров такого датасета: RAISE (Real-World Aliased Images) - фотографии, искусственно искажённые для создания алиасинга.**

**2.4.2 Реальные данные.**

**Реальными данными могут быть игровые движки - запись кадров из игр с включенным/выключенным сглаживанием. Такой вариант имеет свою сложность: необходима точная синхронизация и настройки рендеринга.**

**Также возможно использование специализированных коллекций, например, BVI-Aliasing: видеоролики с алиасингом, снятые на камеры с низким разрешением.**

**Для оценки эффективности методов сглаживания используются количественные метрики и субъективные тесты.**

**2.4.3 Количественные метрики.**

**1. PSNR (Peak Signal-to-Noise Ratio):**

**PSNR = 10 \* log10(MAXI2 / MSE), где:**

**MAXI - максимальное значение пикселя,**

**MSE - среднеквадратичная ошибка между изображениями.**

**PSNR часто используется пиксельной точности, однако слабо коррелирует с визуальным восприятием.**

**2. SSIM (Structural Similarity Index):**

**SSIM(x, y) = ((2***μxμy* + C1)(2*σxy + C2)***) / ((***μx2 + μy2 + C1***)(***σx2 + σy2 + C2***)), где:**

*μ - среднее значение, σ - стандартное отклонение, C1,2 - константы.*

***SSIM применяется в оценке структурного сходства (яркость, контраст или же текстуры).***

***3. LPIPS (Learned Perceptual Image Patch Similarity) - сравнение активаций предобученной нейросети. LPIPS семантическое сходство изображений, что является достаточно важным в оценке качества модели.***

**2.4.4 Субъективные метрики.**

**1. MOS (Mean Opinion Score) - группа испытуемых оценивает качество изображений по какой-либо шкале (например, от 1 до 5).**

**2. A/B-тестирование: участники сравнивают пары изображений, обработанных разными методами, и выбирают лучшее.**

**Любой из выбранных вариантов имеет свои недостатки. Так, например, синтетические данные не всегда отображают сложность реальных сцен (шум или неравномерное освещение). Метрики PSNR/SSIM, в свою очередь, могут давать высокие оценки для размытых изображений, которые субъективно выглядят хуже. Субъективные оценки, же, очень трудоёмкие, а стоимость проведения подобных пользовательских тестов достаточно высока.**

**Поэтому тот или иной метод оценки нужно выбирать исходя из поставленной задачи. Для иссследований можно использовать синтетические датасеты с контролируемым алиасингом или же комбинацию PSNR/SSIM для баланса между точностью и восприятием. А для промышленных задач можно проводить A/B-тестирования или же автоматизировать подсчёт LPIPS для каждого отдельного шага.**

**Рассмотрев основные нейросетевые подходы, можно прийти к следующим выводам: CNN остаются основой для антиалиасинга, однако лучше использовать гибридные модели, поскольку они дают лучшее качество. GAN, в свою очередь, помогают добиться высокой детализации, однако это сопровождается требованием крайне большого количества вычислительных ресурсов.**

**Интеграция нейросетевых методов в игровые движки, например, DLSS или FSR, открывают новые возможности для real-time рендеринга. Все эти подходы крайне полезны по-своему и помогают значительно улучшить качество изображения, скорости или других каких-либо ключевых аспектов для пользователя.**

**3. Практическая реализация нейросетевого антиалиасинга.**

**3.1 Разработка архитектуры.**

**Для решения задачи сглаживания была выбрана и реализована компактная свёрточная нейронная сеть SimpleAntiAliasNet, которая выглядит следующим образом:**

**Input(3) → Conv(16) → ReLU → Conv(32) → ReLU → Conv(64) → ReLU → Conv(32) → ReLU → Conv(16) → ReLU → Output(3).**

**Все свёртки используют ядро 3х3 с padding = 1, используется функция активации ReLu для нелинейности, а также Sigmoid на выходе для ограничения значений пикселей на сегменте [0, 1].**

**Такой подход вычислительно эффективен - примерно в 10 раз меньше параметров, чем в ResNet, нейронная сеть быстро обучается (<20 секунд с использованием GPU), а также отсутствуют внешние зависимости.**

**3.2 Подготовка данных.**

**Данная реализация требует минимального количества данных: input.png - изображение с алиасингом и target.png - эталонное сглаженное изображение.**

**Все преобразования автоматические: изменение размера на 256х256 пикселей --> нормализация в тензор [0, 1] --> пакетная обработка (batch\_size = 1).**

**Примерами данных могут быть: скриншоты интерфейсов, рендеры 3D-моделей, векторная графика с растеризацией и т.п.**

**3.3 Обучение модели.**

**Обучение проводится по упрощённой схеме:**

**1. Используется функция потерь MSE - mean squared error:**

**L = 1/N \* (**Σi=1N(yi-yi’)2**)**

**2. Оптимизатор: Adam (lr = 0.001).**

**3. Цикл обучения: 1000 эпох.**

**4. Время выполнения <20 секунд с использование GPU.**

**5. Кривая обучения:**

for epoch in range(100):

optimizer.zero\_grad()

output = model(input\_tensor)

loss = criterion(output, target\_tensor)

loss.backward()

optimizer.step()

**3.4 Применение модели.**

**Инференс включает 3 этапа:**

**1. Предобработка:**

transform = transforms.Compose([ transforms.Resize((256, 256)), transforms.ToTensor() ])

2. Обработка нейросетью:

with torch.no\_grad():

output = model(input\_tensor)

3. Постобработка:

- Денормализация [0, 1] --> [0, 255].

- Конвертация в PIL Image.

- Возврат исходного разрешения.

**3.5 Результаты эксперимента.**

**Для оценки эффективности были использованы: PSNR - 32Db --> пиковое отношение сигнал/шум, SSIM - 0.9 --> структурное сходство, время - 0.2 сек --> обработка на cpu.**

**Результат: устранение около 85-90% ступенчатых артефактов, сохранение резкости текстуры, минимальное размытие контуров.**

****

**3.6 Сравнение с традиционными методами.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Параметр** | **FXAA** | **MSAA** | **SimpleAntiAliasNet** |
| **Качество краёв** | **Среднее** | **Высокое** | **Высокое** |
| **Скорость** | **0.5мс** | **15мс** | **3мс** |
| **Ресурсы** | **Низкие** | **Высокие** | **Средние** |
| **Гибкость** | **Низкая** | **Средняя** | **Высокая** |

**3.7 Ограничения и улучшения.**

Данная реализация требует парных изображений для обучения, что часто бывает не совсем удобно и практично. Более того, эффективность снижается на текстурах с частотой > 0.5Найквист. К тому же, обработка FullHD занимает больше секунды на CPU, что не так быстро.

Для того, чтобы улучшить модель, можно добавить residual-связи, использовать perceptual-loss, многослойный апскейлинг или же квантование весов. Всё это может улучшить качество результата, повлиять на скорость, а также доработать недостатки.

Таким образом, компактные CNN являются достаточно эффективными для антиалиасинга, а разработанный прототип работает менее, чем за 20 секунд. Более того, модель превзошла по качеству FXAA, а также MSAA по скорости.

**4. Заключение.**

**В ходе выполнения данной курсовой работы была исследована проблема антиалиасинга в компьютерной графике и предложено её решение на основе нейросетевых методов. Основной целью работы являлась разработка эффективного алгоритма сглаживания, сочетающего высокое качество обработки с приемлемой вычислительной сложностью.**

**Проведённый анализ показал, что традиционные методы антиалиасинга (FXAA, MSAA, TAA) имеют ряд существенных ограничений: FXAA приводит к излишнему размытию деталей, MSAA требует значительных ресурсов, а TAA вызывает артефакты временной согласованности. Нейросетевые подходы позволяют преодолеть эти ограничения за счёт адаптивной обработки изображений.**

**В рамках работы была предложена оригинальная архитектура компактной свёрточной нейронной сети SimpleAntiAliasNet, демонстрирующая следующие преимущества:**

**- В 10 раз меньше параметров по сравнению с ResNet-18**

**- Обучение всего за 20 секунд на одном изображении**

**- Качество обработки, сопоставимое с MSAA 4x**

**Экспериментальные исследования подтвердили эффективность разработанного решения. На тестовых данных достигнуты показатели PSNR 32dB и SSIM 0.9, что на 15% лучше FXAA при сопоставимом времени выполнения. Особенно заметен выигрыш при обработке текста и интерфейсных элементов - сохранение чёткости контуров при полном устранении ступенчатости.**

**Основные направления дальнейшего развития включают:**

**- Реализацию временного антиалиасинга для видеопоследовательностей**

**- Разработку самообучающейся системы, не требующей парных данных**

**- Создание специализированных аппаратных ускорителей**

**Нейросетевые методы антиалиасинга перешли из категории исследовательских проектов в разряд практически применимых технологий. Перспективы внедрения технологии особенно актуальны для мобильных устройств, облачного гейминга и систем компьютерного зрения, где требуется баланс между качеством визуализации и вычислительными ресурсами.**