УДК 004.89

Система синтеза видео на основе технологии DeepFake

М.О. Лукащук[*1], С.А. Зори[*2]

[*1] магистрант, Донецкий национальный технический университет,
mikhail.lukashchuk@mail.ru

[*2] д.т.н., доцент, Донецкий национальный технический университет,
ik.ivt.rec@mail.ru, OrcID: 0000-0003-4018-234X, SPIN-код: 3565-6330

Аннотация

В статье представлен краткий обзор существующих методов синтеза видео с использованием технологий DeepFake. На основе анализа современных методов предложен подход к улучшению технологии, направленный на повышение реалистичности и эффективности создаваемых видео, дано описание алгоритмов и моделей, применяемых для генерации видео. Предложенные подходы могут быть перспективными для дальнейших разработок и практического применения в области видеосинтеза и глубокого обучения.

Введение

На сегодняшний день технология DeepFake широко используется для создания реалистичных видео, где изображение лица на видео может быть изменено или заменено лицом другого человека [1]. Несмотря на возможности применения в развлекательной и образовательной сферах, технология DeepFake также вызывает обеспокоенность в отношении безопасности данных и этики. Проблема создания реалистичных видео путем синтеза лиц или замены объектов актуальна не только в связи с вопросами прав собственности и контроля над контентом, но и с точки зрения оптимизации ресурсов и улучшения качества конечного видео.

Анализ современных методов синтеза изображений и применения технологий на основе генеративных состязательных сетей (GAN) рассмотрен в [2]. Согласно этому источнику, наиболее высококачественные результаты синтеза достигаются при использовании GAN [3] и их модификаций, таких как StyleGAN [4] и CycleGAN [5], которые обладают мощным потенциалом в создании фотореалистичных изображений. Эти подходы, основанные на конкурентной системе генератора и дискриминатора, позволяют генерировать изображения, а, следовательно, и видео, которые могут точно имитировать оригинальные.

Под термином DeepFake подразумевается использование алгоритмов, позволяющих синтезировать видео с высокой степенью реализма, сохраняя функции исходного видео, но значительно усложняя процесс его анализа и обратной инженерии. Существующие инструменты для создания DeepFake, такие как DeepFaceLab [6] и FaceSwap [7], активно используют генеративные сети для оптимизации синтеза, хотя их применение требует значительных вычислительных ресурсов и времени.

На данный момент существует множество программных решений, предлагающих инструменты для создания и обработки видео DeepFake, но наиболее производительные и гибкие методы преимущественно представлены в коммерческом формате. Бесплатные решения часто ограничены базовыми функциями, а для достижения оптимальных результатов необходимы значительные вычислительные мощности.

Целью данной работы является демонстрация и улучшение комплекса алгоритмов для синтеза видео с использованием DeepFake, которые обеспечат высокое качество изображения и оптимальную производительность. Такой подход позволит использовать технологию DeepFake в безопасных и полезных приложениях, таких как виртуальная реальность и образовательные проекты.

1 Проектирование системы синтеза видео на основе технологии DeepFake

В качестве прототипа системы синтеза видео на основе DeepFake для апробации разрабатываемых алгоритмов улучшения качества работы технологии предполагается разработка программной системы, которая состоит из нескольких ключевых компонентов, каждый из которых выполняет определённые функции в процессе генерации видео. Для демонстрации взаимодействия компонентов в системе разработана диаграмма компонентов (представлена на рисунке 1). На ней показаны 8 модулей проектируемой системы. Модуль загрузки данных отвечает за загрузку и хранение исходных видео и изображений. Модуль предобработки данных включает в себя функции нормализации, масштабирования и аугментации данных. Модуль обнаружения и отслеживания лиц локализует лица на видео и отслеживает их в течение всего ролика.

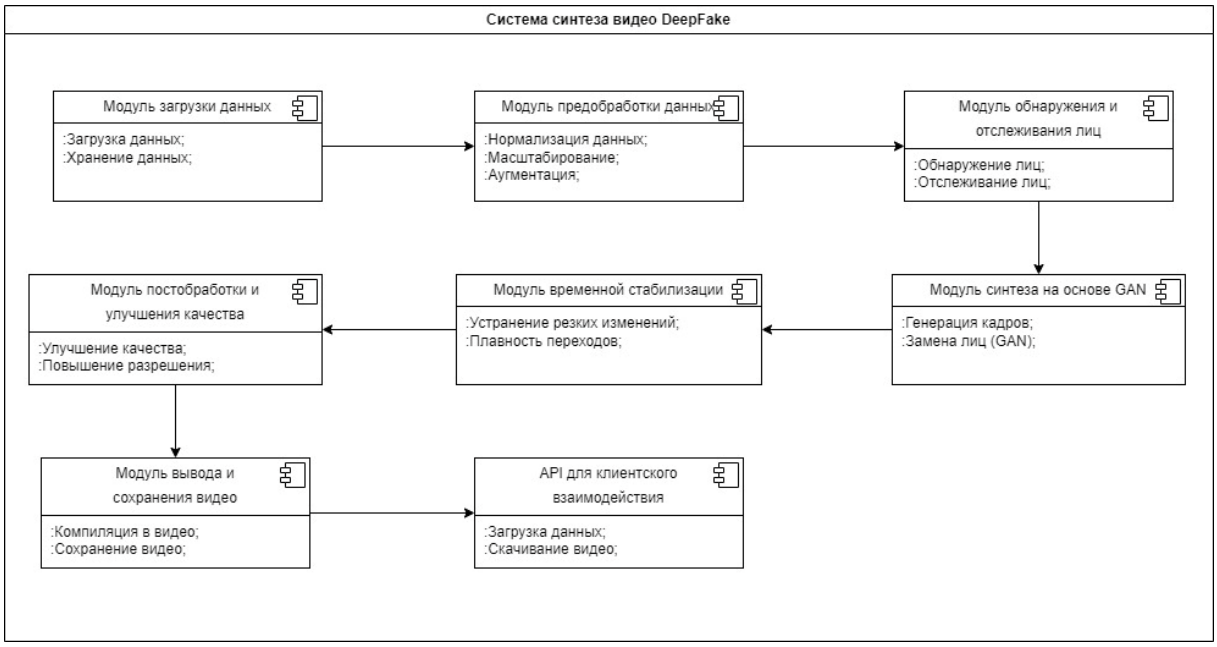

Рисунок 1 – Диаграмма компонентов системы синтеза видео

Модуль синтеза на основе GAN выполняет генерацию новых кадров или замену лиц на основе GAN-алгоритмов. Модуль временной стабилизации устраняет резкие изменения между кадрами для повышения плавности. Модуль постобработки и улучшения качеств: улучшает разрешение и качество изображений. Модуль вывода и сохранения видео компилирует обработанные кадры в видеофайл и сохраняет его. API для клиентского взаимодействия обеспечивает доступ к системе через API, позволяя загружать и скачивать видео.

Процесс синтеза видео включает последовательное выполнение операций по подготовке данных, синтезу и постобработке. На начальном этапе модуль предобработки готовит исходные данные. После этого генеративная сеть выполняет преобразования для синтеза видео. Постобработка стабилизирует изображение, и готовый видеоролик сохраняется. Исходя из спроектированной диаграммы компонентов, была разработана диаграмма последовательности выполнения операций, которая представлена на рис. 2.

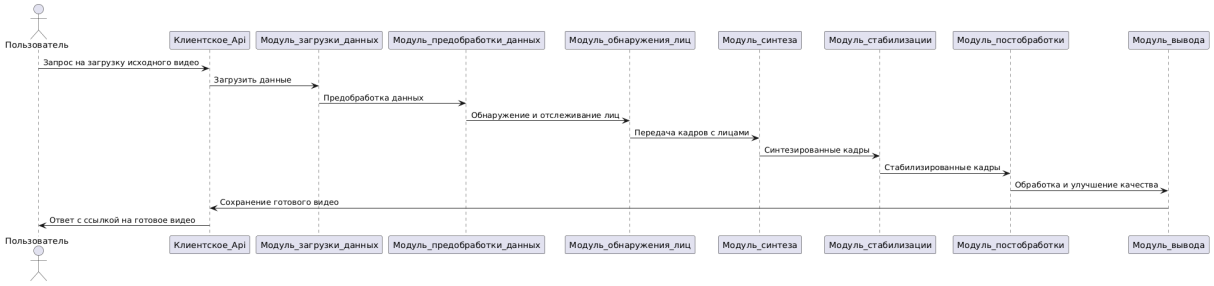

Рисунок 2 – Диаграмма последовательности выполнения операций

Важным элементом системы является физическое расположение основных компонентов системы, включая сервер для вычислений, клиентское устройство пользователя, а также хранилище данных. Для работы системы требуются высокопроизводительные ресурсы, такие как GPU, а также серверные мощности для хранения данных. Основные компоненты системы, включая сервер с GAN-архитектурой, расположены на облачных серверах. Для отображения физического расположения основных элементов системы была разработана диаграмма развёртывания (см. рис. 3).

2 Алгоритм GAN для синтеза видео

Далее будет рассмотрен алгоритм синтеза видео, наиболее актуальным на данный момент является подход с применением технологий на основе генеративных состязательных сетей (GAN). Алгоритм генеративно-состязательных сетей (GAN) для синтеза видео включает два ключевых компонента: генератор и дискриминатор. Генератор обучается создавать реалистичные кадры на основе шума или изображения, а дискриминатор пытается отличить сгенерированные кадры от реальных.

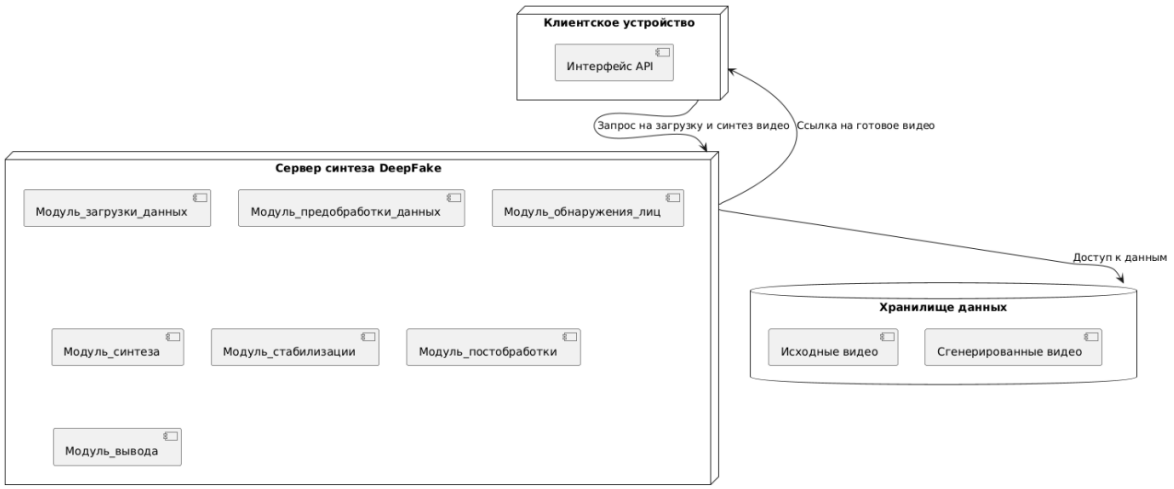

Рисунок 3 – Диаграмма развёртывания

Эти два компонента взаимодействуют друг с другом: генератор учится «обманывать» дискриминатор, создавая все более реалистичные изображения, а дискриминатор старается повысить точность распознавания реальных и сгенерированных кадров.

В контексте алгоритма GAN случайный шум — это вектор случайных чисел, который используется в качестве входных данных для генератора. Это своего рода "платформа" для генерации новых образцов данных, например, изображений или видеокадров. Генератор принимает случайный шум (чаще всего это просто вектор случайных чисел, например, сгенерированных с помощью нормального распределения) и преобразует его в нечто осмысленное, например, «фейковое» изображение или видеокадр. Суть заключается в том, что этот шум не имеет смысла сам по себе, но с помощью нейронной сети генератор обучается создавать из него реалистичные данные. Важно, что случайный шум не представляет собой какие-то заранее определенные данные (например, изображение или видео), а скорее его роль — это создание исходной случайности, на основе которой нейросеть «учится» генерировать осмысленные данные. Каждое изменение случайного шума может привести к созданию разных, но похожих по стилю или содержанию объектов.

Веса — это параметры, которые определяют, как входные данные (в данном случае случайный шум или реальные данные) обрабатываются нейронной сетью, включая как генератор, так и дискриминатор. Генератор использует веса для преобразования случайного шума в изображение или кадр, то есть для того, чтобы сгенерировать изображение, которое будет максимально похожим на реальные данные. Дискриминатор, в свою очередь, использует свои веса для того, чтобы распознать, является ли кадр реальным или сгенерированным.

Для лучшего понимания, алгоритм можно представить в виде диаграммы последовательности (см. рис. 4).

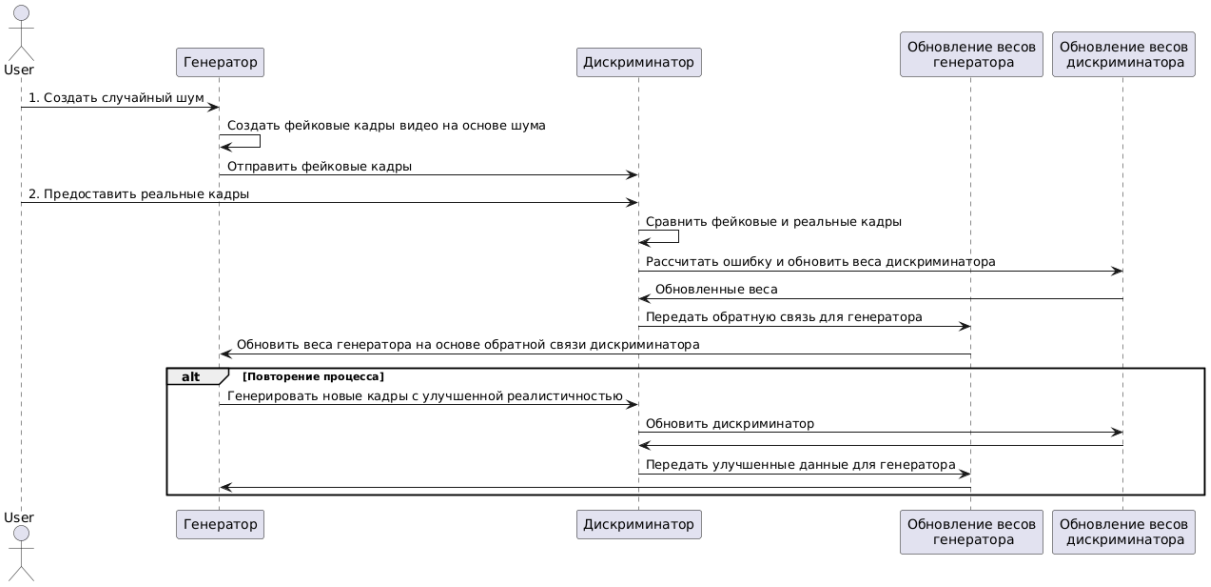

Рисунок 4 – Алгоритм генеративно-состязательных сетей (GAN)

Процесс обучения заключается в том, чтобы на основе ошибок (ошибки дискриминатора, который ошибочно принял фейк за реальное изображение) откорректировать веса сети таким образом, чтобы генератор создавал более реалистичные данные, а дискриминатор мог их лучше распознавать. Для синтеза видео на основе GAN существуют несколько подалгоритмов, ответственных за работу генератора и дискриминатора. Эти подалгоритмы включают обучение генератора, обучение дискриминатора, и постепенное улучшение кадров. Ниже представлены некоторые из них.

2.1 Алгоритм обучения дискриминатора

Дискриминатор обучается на основе двух типов данных: реальных и сгенерированных кадров. Цель дискриминатора — правильно различать эти два типа, минимизируя ошибку классификации. На рисунке 5 представлен псевдокод данного алгоритма.

Далее, при описании функций и алгоритмов, будем опираться на исследование [8] Бинарная кросс-энтропия — это функция потерь, которая используется для оценки качества работы моделей, решающих задачи бинарной классификации.

```
Input: Реальные кадры из обучающего набора данных и сгенерированные кадры от генератора
Output: Обновленные веса дискриминатора

1. Для каждого батча кадров:
    a. Получить реальные кадры X_real и присвоить метки Y_real = 1
    b. Получить сгенерированные кадры X_fake от генератора и присвоить метки Y_fake = 0
    c. Объединить X_real и X_fake в общий обучающий набор X и метки Y
    d. Вычислить предсказание D(X) с использованием дискриминатора
    e. Вычислить ошибку дискриминатора: loss_D = BCE(Y, D(X)),
       где BCE - бинарная кросс-энтропия
    f. Обновить веса дискриминатора, используя градиентный спуск с loss_D
```

Рисунок 5 – Псевдокод алгоритма обучения дискриминатора

Задача бинарной классификации заключается в том, чтобы классифицировать объекты на две категории, например, «истинное» или «ложное», «1» или «0», «реальное» или «сгенерированное». Для одного предсказания функция потерь, основанная на бинарной кросс-энтропии, рассчитывается по формуле 1:

$$L = -(y * \log(\hat{y}) + (1 - y) * \log(1 - \hat{y})) \quad (1)$$

где y — истинное значение (0 или 1);

$\hat{y}$ — предсказанное моделью значение вероятности (от 0 до 1).

Если истинное значение y=1, то функция потерь становится $-\log(\hat{y})$, что означает, что чем ближе $\hat{y}$ к 1, тем меньше значение функции потерь. Цель обучения модели — минимизировать среднее значение бинарной кросс-энтропии для всех обучающих данных, что будет означать, что модель делает правильные предсказания с высокой уверенностью.

Градиентный спуск — это алгоритм оптимизации, который используется для нахождения минимума функции потерь (например, бинарной кросс-энтропии). Цель градиентного спуска — настроить параметры модели (веса), чтобы минимизировать ошибку предсказания. Сначала вычисляется градиент функции потерь по каждому параметру модели. Градиент указывает направление, в котором функция потерь растет быстрее всего. Параметры модели обновляются в направлении, противоположном градиенту. Предыдущие шаги повторяются многократно, пока функция потерь не станет минимальной или почти минимальной.

2.2 Алгоритм обучения генератора

Цель генератора — создать кадры, которые дискриминатор примет за реальные. Обучение генератора происходит по обратной связи, полученной от дискриминатора, который указывает, насколько реалистичными получились кадры.

На рисунке 6 представлен псевдокод данного алгоритма. Генератор обновляет свои параметры, стремясь уменьшить способность дискриминатора отличать реальные данные от фейковых. Его задача — «обманывать» дискриминатор, чтобы тот посчитал фейковые данные реальными. Если генератор успешен, дискриминатор начинает терять способность различать, какие данные — настоящие, а какие — поддельные.

Эта «соревновательная игра» продолжается до тех пор, пока генератор не научится создавать данные, которые выглядят очень правдоподобно, и дискриминатор перестает с лёгкостью их отличать.

2.3 Алгоритм постепенного улучшения кадров

Для создания реалистичного видео важно не только качество отдельных кадров, но и плавность переходов между ними. Алгоритм постепенного улучшения кадров помогает улучшить временную стабильность, чтобы сгенерированные кадры плавно перетекали друг в друга. На рисунке 7 представлен псевдокод данного алгоритма.

```
Input: Случайный шум или изображение-источник
Output: Обновленные веса генератора

1. Для каждого батча:
    a. Сгенерировать фейковые кадры X_fake из случайного шума или изображения с использованием генератора
    b. Получить предсказание дискриминатора D(X_fake) для сгенерированных кадров
    c. Установить целевые метки Y_fake = 1 (поскольку генератор хочет «обмануть» дискриминатор)
    d. Вычислить ошибку генератора: loss_G = BCE(Y_fake, D(X_fake))
    e. Обновить веса генератора, используя градиентный спуск с loss G
```

Рисунок 6 – Алгоритм обучения генератора

```
Input: Сгенерированные кадры X_fake в порядке последовательности
Output: Обновленные кадры X_smooth с уменьшением резких изменений

1. Инициализировать предыдущий кадр как X_prev = X_fake[0]
2. Для каждого кадра X_t в последовательности X_fake:
    a. Вычислить разницу delta = |X_t - X_prev|
    b. Если delta > threshold (порог допустимых изменений):
        - Установить X_t = (X_t + X_prev) / 2, чтобы уменьшить разницу
    c. Обновить X_prev = X_t
3. Вернуть сглаженные кадры X_smooth
```

Рисунок 7 – Алгоритм постепенного улучшения кадров

3 Итоговый алгоритм синтеза видео

На основании описанных выше алгоритмов, можно представить общий алгоритм работы системы, который, исходя из анализа научных работ, является наиболее актуальным на момент написания статьи. Алгоритм представлен на рисунке 8.

Синтез видео с использованием GAN имеет значительные достижения, но также сталкивается с рядом недостатков и сложностей, которые ограничивают его использование и требуют улучшений. В исследовании [9] выделены некоторые недостатки метода:

GAN требует значительных вычислительных ресурсов для обучения, особенно для задач синтеза видео, где каждый кадр должен быть реалистичным и плавно переходить в следующий.

GAN в синтезе видео должен не только создавать реалистичные отдельные кадры, но и обеспечивать плавные, естественные переходы между ними. Часто это приводит к проблемам, как, например, «мерцание» или «дрожание» отдельных частей изображения, что нарушает целостность видео. Такие артефакты становятся особенно заметными при синтезе длинных видео.

Генератор может создавать артефакты, такие как размытые или искаженные участки изображения. Это особенно проблематично в динамичных сценах, где модель не успевает корректно адаптировать каждый кадр.

```
Input: Набор тренировочных данных (видеокадры) или исходное видео
Output: Сгенерированное видео с плавными, реалистичными кадрами

1. Загрузка данных:
    a. Загрузить исходное видео или тренировочный набор кадров.
    b. Разделить видео на отдельные кадры, если исходные данные представляют собой видео.

2. Предобработка данных:
    a. Применить фильтрацию и нормализацию к каждому кадру для улучшения качества.
    b. Использовать метод обнаружения лиц для выделения областей с лицами на кадрах.

3. Инициализация GAN:
    a. Инициализировать веса генератора и дискриминатора случайными значениями.
    b. Установить начальные значения параметров обучения.

4. Обучение GAN:
    Пока ошибка дискриминатора не стабилизируется или не достигнуто максимальное количество эпох:
        a. Обновление дискриминатора:
            - Для каждого батча реальных и сгенерированных кадров:
                i. Подать реальные кадры и установить метки Y_real = 1.
                ii. Сгенерировать фейковые кадры с помощью генератора и установить метки Y_fake = 0.
                iii. Рассчитать ошибку дискриминатора loss_D, используя бинарную кросс-энтропию между предсказаниями и истинными метками.
                iv. Обновить веса дискриминатора с помощью градиентного спуска для минимизации loss_D.

        b. Обновление генератора:
            - Для каждого батча шума или изображения-источника:
                i. Сгенерировать фейковые кадры X_fake.
                ii. Получить предсказания дискриминатора для X_fake.
                iii. Установить метки Y_fake = 1, чтобы "обмануть" дискриминатор.
                iv. Рассчитать ошибку генератора loss_G на основе предсказаний дискриминатора.
                v. Обновить веса генератора для минимизации loss_G.

5. Постобработка кадров (сглаживание кадров):
    a. Инициализировать переменную X_prev как первый сгенерированный кадр.
    b. Для каждого кадра X_t из сгенерированной последовательности:
        i. Вычислить разницу между текущим и предыдущим кадром delta = |X_t - X_prev|.
        ii. Если delta больше допустимого порога:
            - Установить X_t = (X_t + X_prev) / 2, чтобы уменьшить резкие переходы.
        iii. Обновить X_prev = X_t.

6. Создание финального видео:
    a. Объединить сглаженные кадры в видеопоследовательность.
    b. Сохранить видео в заданном формате.

7. Вернуть сгенерированное видео.
```

Рисунок 8 – Общий алгоритм работы системы

Дискриминатор может начать «запоминать» тренировочные данные и терять способность оценивать правдоподобие новых данных, что ухудшает качество синтеза видео.

GAN часто теряют последовательность и правдоподобие в длинных видео. Для реалистичного синтеза длинных видео требуется учитывать взаимосвязь не только между соседними кадрами, но и в более широком временном контексте, что значительно усложняет задачу. Без таких зависимостей модель не может поддерживать стабильное качество по мере увеличения длины видео.

Для улучшения алгоритмов синтеза видео с помощью GAN, можно предложить несколько направлений и конкретных решений, позволяющих повысить качество, стабильность и управляемость результатов.

Вот ключевые подходы:

Включение временных зависимостей позволяет улучшить согласованность кадров, минимизируя артефакты на границах между ними. Для этого в архитектуру GAN добавляются слои, обрабатывающие последовательность кадров, например, рекуррентные нейронные сети [10].

Добавление второго дискриминатора, который проверяет временные зависимости между кадрами, помогает модели лучше понимать, как объекты и фон должны изменяться от кадра к кадру. Один дискриминатор проверяет качество каждого кадра, а второй — их последовательность.

Внедрение атрибутов (например, меток объектов) в генератор для управления содержимым и движением в видео. Это позволяет задать начальные условия, чтобы контролировать вид и поведение объектов, улучшая предсказуемость и управляемость.

Использование промежуточных слоев для повышения разрешения (например, от низкого к высокому) улучшает качество видео, обеспечивая плавность переходов между кадрами и детализацию.

С учётом данных предложений, на рисунке 9 представлен обновлённый алгоритм синтеза видео с использованием технологии GAN.

```
Input: Набор тренировочных данных (видеокадры) или исходное видео
Output: Сгенерированное видео с плавными, реалистичными кадрами

1. Загрузка данных:
   a. Загрузить исходное видео или набор тренировочных кадров.
   b. Если данные представлены как видео, разделить его на отдельные кадры.

2. Предобработка данных:
   a. Применить фильтрацию и нормализацию к каждому кадру.
   b. Использовать обнаружение лиц для выделения областей на кадрах.

3. Инициализация GAN с улучшенными компонентами:
   a. Инициализировать веса генератора, пространственного и временного дискриминаторов.
   b. Добавить рекуррентные нейронные сети в генератор и дискриминатор для обработки последовательности кадров.
   c. Настроить слои многоуровневой генерации, увеличивающие разрешение на каждом этапе.
   d. Установить параметры обучения (скорость обучения, количество эпох и функции потерь).

4. Обучение GAN:
   while ошибка дискриминатора не стабилизируется или не достигнуто максимальное количество эпох:

      a. Обновление дискриминаторов:
         for каждый батч реальных и сгенерированных кадров:
            1. Подать реальные кадры в пространственный дискриминатор и установить метки как реальные.
            2. Использовать временной дискриминатор для проверки последовательностей кадров.
            3. Сгенерировать фейковые кадры с помощью генератора и установить метки как фейковые.
            4. Рассчитать ошибки дискриминаторов и обновить их веса с помощью градиентного спуска.

      b. Обновление генератора:
         for каждый батч случайного шума или начального изображения:
            1. Сгенерировать последовательность кадров, используя RNN в генераторе для учета временной последовательности.
            2. Подать сгенерированные кадры на дискриминаторы для проверки пространственной и временной согласованности.
            3. Установить метки как реальные, чтобы "обмануть" дискриминаторы.
            4. Рассчитать ошибку генератора и обновить его веса для минимизации этой ошибки.

5. Постобработка кадров (сглаживание переходов):
   a. Инициализировать переменную для хранения первого сгенерированного кадра.
   b. for каждый кадр из сгенерированной последовательности:
      1. Вычислить разницу между текущим и предыдущим кадром.
      2. if разница > порог:
         - Усреднить текущий и предыдущий кадры для сглаживания перехода.
      3. Обновить переменную на текущий кадр.

6. Создание финального видео:
   a. Объединить сглаженные кадры в видеопоследовательность.
   b. Сохранить видео в заданном формате.

7. Вернуть сгенерированное видео.
```

Рисунок 9 – Итоговый алгоритм синтеза видео с использованием технологии GAN

Выводы

В работе был проведён системный анализ методов и архитектур систем синтеза видео, в частности, алгоритмов на основе генеративных состязательных сетей (GAN). Основываясь на анализе, были выявлены ключевые проблемы стандартных GAN, такие как низкая согласованность кадров и недостаточная детализация, что может приводить к визуальным артефактам и неестественным переходам между

кадрами. Для устранения этих недостатков предложены несколько усовершенствований, включая: использование рекуррентных нейронных сетей, включение дополнительного временного дискриминатора, многоуровневая генерация с постепенным увеличением разрешения и внедрение управляемых атрибутов.

Эти методы были интегрированы в общую структуру GAN, позволив сформировать улучшенный алгоритм синтеза видео. Предложенные подходы могут быть перспективными для дальнейших разработок в области видеосинтеза и глубокого обучения, а также полезными в практическом применении для задач создания контента, виртуальной реальности и других областей мультимедийных технологий.

Литература

1. Deepfake: краткая история появления и нюансы работы технологии [Электронный ресурс]. – Режим доступа: https://habr.com/ru/companies/neuronet/articles/592119/. – Загл. с экрана.

2. Узких, Г. Ю. Применение генеративно-состязательных сетей (GAN) в обработке изображений / Г. Ю. Узких // Вестник науки. – 2024. – Т. 4, № 8 (77). – С. 182-185.

3. Generative adversarial network [Электронный ресурс] // Википедия. – Режим доступа: https://en.wikipedia.org/wiki/Generative_adversarial_network. – Загл. с экрана.

4. StyleGAN: Revolutionizing AI-Driven Image Creation [Электронный ресурс]. – Режим доступа: https://www.simplilearn.com/tutorials/generative-ai-tutorial/stylegan. – Загл. с экрана.

5. Cycle Generative Adversarial Network (CycleGAN) [Электронный ресурс]. – Режим доступа: https://www.geeksforgeeks.org/cycle-generative-adversarial-network-cyclegan-2/. – Загл. с экрана.

6. DeepFaceLab - AI-Powered Face Manipulation and Editing [Электронный ресурс]. – Режим доступа: https://perchance-ai.vercel.app/free-ai-tools/deepfacelab. – Загл. с экрана.

7. Face Swap Video [Электронный ресурс]. – Режим доступа: https://faceswapvideo.ai/. – Загл. с экрана.

8. Великанов, М. С. Нейросетевой алгоритм поиска областей открытия/закрытия в видеопоследовательностях / М. С. Великанов, А. Б. Анзина, С. В. Лаврушкин, Д. С. Ватолин // International Journal of Open Information Technologies. – 2020. – Т. 8, № 3. – С. 55-62.

9. Аверченков, А. В. Анализ и применение генеративно-состязательных сетей для получения изображения высокого качества / А. В. Аверченков, А. А. Андросов, Ю. А. Малахов // Эргодизайн. – 2020. – № 4. – С. 167-175.

10. Рекуррентная нейронная сеть (RNN): виды, обучение, примеры [Электронный ресурс]. – Режим доступа: https://neurohive.io/ru/osnovy-data-science/rekurrentnye-nejronnye-seti/. – Загл. с экрана.

Лукащук М.О., Зори С.А. Система синтеза видео на основе технологии DeepFake. *В статье представлен краткий обзор существующих методов синтеза видео с использованием технологий DeepFake. На основе анализа современных методов предложен подход к улучшению технологии, направленный на повышение реалистичности и эффективности создаваемых видео, дано описание алгоритмов и моделей, применяемых для генерации видео. Предложенные подходы могут быть перспективными для дальнейших разработок и практического применения в области видеосинтеза и глубокого обучения.*

Ключевые слова: *DeepFake, синтез видео, генеративные модели, GAN, StyleGAN, алгоритмы, машинное обучение, нейронные сети*

Lukashchuk M.O., Zori S.A. Video Synthesis Methods Based on DeepFake Technology. *This article provides a brief overview of existing video synthesis methods utilizing DeepFake technology. Based on the analysis of current methods, an approach to improving the technology is proposed, aimed at increasing the realism and efficiency of the generated videos, and the algorithms and models used for video generation are described. The proposed approaches may be promising for further development and practical application in the field of video synthesis and deep learning*.

Keywords: *DeepFake, video synthesis, generative models, GAN, StyleGAN, algorithms, machine learning, neural networks*

Статья поступила в редакцию 17.05.2025
Рекомендована к публикации профессором Мальчевой Р. В.