ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
Факультет компьютерных наук
Образовательная программа «Программная инженерия»

УДК 004.932.4

СОГЛАСОВАНО
Руководитель проекта
Project Leader,
Samsung AI
_____ Харламов А. В.
«11» апреля 2022 г.

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Программная инженерия»
профессор департамента программной
инженерии, канд. техн. наук
_____ В.В. Шилов
«___» _____ 20__ г.

Отчет

по исследовательскому курсовому проекту

на тему Создание нейронной сети для отображения изображений в другие домены на мобильных устройствах

по направлению подготовки бакалавров 09.03.04 «Программная инженерия»

Выполнил
студент группы БПИ197
образовательной программы
09.03.04 «Программная инженерия»
А. В. Ермилов
И.О. Фамилия
11.04.2022
Подпись, Дата

Москва 2022

Отчет 35 с., 42 рис., 35 источн.

Аннотация

Ключевые слова: *Computer Vision, Machine Learning, Deep Learning, Image-to-Image Translation, Image Processing, GAN, StyleGAN, StyleGAN blending*.

Объект исследования: отображение изображений в другие домены.

Предмет исследования: методы глубинного обучения в задаче отображения изображений в другие домены.

Цель исследования: поиск способа отображения изображений между доменами в высоком качестве и с минимальными артефактами.

Задачи исследования:

- Изучение теории генеративно-состязательных сетей в задаче отображения изображений между доменами;
- Сравнение эффективности разных архитектур и подходов для отображения изображений в другие домены;
- Устранение артефактов, появляющихся в процессе отображения изображений из одного домена в другой;
- Улучшение качества существующих методов отображения изображений в другие домены.

Методы исследования:

- Сравнительный анализ различных моделей и подходов отображения изображений в другие домены;
- Проведение экспериментов для выявления и устранения внесенных артефактов и сравнения эффективности разных подходов отображения изображений в другие домены.

Научная новизна работы состоит в следующем:

- Получена модель, превосходящая по качеству существующие в задаче отображения лиц в стиль сериала Arcane;
- Собран первый крупный публичный датасет лиц высокого качества для анимационного сериала Arcane.

Достоверность научных результатов подтверждена качеством итоговой модели при ее применении к реальным изображениям.

Практическая значимость. Результаты работы могут широко использоваться в коммерческих целях для создания фильтров и масок для пользователей в социальных сетях и интернете.

Результаты работы:

- Получена модель, превосходящая по качеству существующие в задаче отображения лиц в стиль сериала Arcane;
- Собран первый крупный публичный датасет лиц высокого качества для анимационного сериала Arcane;

- Изучены широко применяемые модели генеративно-состязательных сетей в задаче отображения изображений в другие домены;
- Проведены эксперименты, на основе которых удалось сравнить эффективность разных архитектур моделей и разных подходов отображения изображений в другие домены и улучшить их качество, а также выявить артефакты, появляющиеся в процессе отображения изображений в другие домены, и найти способы их устранения.

Оглавление

Термины и сокращения **4**

1 Введение **5**

2 Существующие методы отображения изображений в другие домены **6**

2.1 Генеративно-состязательные сети **6**

2.2 Условные генеративно-состязательные сети **6**

2.3 pix2pix **7**

2.4 CycleGAN **8**

2.5 U-GAT-IT **9**

3 Сбор и обработка данных **10**

4 Обучение на непарных данных **11**

5 Генерация парных данных **12**

6 Обучение с pix2pix **19**

7 Улучшенный подход к обучению **24**

8 Ускорение модели и уменьшение ее размера **27**

9 Выводы **32**

10 Направления дальнейшей работы **32**

Список использованных источников **33**

Термины и сокращения

ResNet – архитектура сверточной нейронной сети, использующая соединения быстрого доступа.

StyleGAN – генеративно-состязательная сеть, предназначенная для генерации изображений высокого разрешения с возможностью контролирования как высокоуровневых, так и низкоуровневых черт генерируемого изображения.

StyleGAN блендинг – техника смешивания весов нескольких StyleGAN моделей, обученных на разных доменах, для генерации некоторого их промежуточного домена.

U-Net – это архитектура сверточной нейронной сети, состоящая из кодирующей и восстанавливающей части, использующаяся для сегментации изображений.

Артефакт – искажения изображения, которые не присутствуют в отображаемом объекте.

Аугментация – способ увеличения выборки данных через модификацию существующих данных.

Батч – набор объектов, который модель видит за одну итерацию.

Генеративно-состязательные сети – алгоритм машинного обучения, входящий в семейство генеративных моделей и построенный на комбинации из двух нейронных сетей: генеративная модель G, строящая приближение распределения данных, и дискриминативная модель D, которая оценивает вероятность, что образец пришел из реальных данных, а не является сгенерированным моделью G.

Гиперпараметр – параметры алгоритмов, значения которых устанавливаются перед запуском процесса обучения и не меняются в его процессе.

Датасет – обработанный и структурированный набор данных.

Декодер – часть модели, преобразующая некоторый набор признаков, полученных с помощью энкодера, в выходной объект.

Компьютерное зрение – это научное направление в области искусственного интеллекта и связанные с ним технологии получения изображений объектов реального мира, их обработки и использования полученных данных для решения разного рода прикладных задач без участия (полного или частичного) человека [32].

Кроп – вырезанная часть изображения.

Машинное обучение – класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение за счет применения решений множества сходных задач [33].

Механизм внимания – техника, используемая в рекуррентных нейронных сетях и сверточных нейронных сетях для поиска взаимосвязей между различными частями входных и выходных данных, часто выражающаяся в виде полносвязной сети [34].

Батч нормализация – метод, позволяющий повысить производительность искуственных нейронных сетей и стабилизировать их работу.

Перенос стиля изображения – процесс преобразования стиля исходного изображения к стилю выбранного изображения.

Сверточная нейронная сеть – специальная архитектура искусственных нейронных сетей, изначально нацеленная на эффективное распознавание изображений [35].

Тестовая выборка – выборка, используемая для оценивания итогового качества построенной модели.

Функция потерь – функция, характеризующая потери при неправильном принятии решений на основе наблюдаемых данных.

Энкодер – часть модели, преобразующая входной объект в некоторый набор признаков.

1 Введение

На сегодняшний день отображение изображений в другие домены является одной из наиболее востребованных задач в рамках компьютерного зрения. Данная задача обладает широким кругом прикладных направлений, таких как сегментация [12], актуальная в создании систем для беспилотных автомобилей, закрашивание [9], реконструкция [11] и колоризация [4] для восстановления исторических фотографий, сверхразрешение [28] для улучшения качества фильмов 19-х и 20-х веков и прочее.

За последнее десятилетие стремительное развитие в области машинного обучения, а также стремительный рост вычислительной мощности центральных и графических процессоров, необходимых для более эффективной и быстрой работы спроектированных моделей, актуализировали задачу переноса изображений в другие домены как никогда раньше, а колоссальный прогресс в теории глубинного обучения способствовал воплощению идеи применения сверточных нейронных сетей и генеративно-состязательных моделей для решения данной задачи.

Благодаря бурному всплеску социальных сетей, отдельные направления этой задачи, в частности перенос стиля [21], вызвали значительный спрос в виде различных фильтров и масок, применяемых к фотографиям пользователей. Но помимо качества осуществляемого переноса стиля немаловажным стал вопрос о производительности применения таких фильтров, связанный со все большим объемом контента, потребляемого через мобильные устройства, так как объем их вычислительных ресурсов существенно меньше, чем у промышленных серверов, на которых обучаются и часто также применяются модели глубинного обучения. В связи с такими требованиями по эффективности и компактности моделей, многие state-of-the-art методы и архитектуры отображения изображений в другие домены становятся невозможными к применению на мобильных устройствах, так как их потребление по памяти может доходить до десятков гигабайт.

Целью данной работы является изучение и совершенствование существующих методов и подходов отображения изображений в другие домены, а также получение модели, превосходящей по качеству существующие в задаче отображения лиц в стиль анимационного сериала Arcane [2], применение которой также возможно на мобильных устройствах или серверах с существенно ограниченным объемом памяти.

2 Существующие методы отображения изображений в другие домены

2.1 Генеративно-состязательные сети

Классический алгоритм генеративно-состязательных сетей (Generative Adversarial Network, GAN) [7] построен на комбинации и одновременном обучении двух моделей – генератора G и дискриминатора D. Задачей генератора является создание образцов, неотличимых для дискриминатора от реальных, в то время как цель дискриминатора – научиться максимально точно распознавать и отличать созданные генератором образцы от настоящих. Таким образом, между двумя сетями возникает игра с нулевой суммой (рис. 1). Альтернативно, алгоритм генеративно-состязательных сетей можно представить, как обучение некоторой сети G на обучаемой функции потерь D.

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

Рис. 1. Формула целевой функции для генератора и дискриминатора в алгоритме генеративно-состязательных сетей. D(x) - предсказанная дискриминатором вероятность того, что “реальный” образец x является настоящим, а D(G(z)) - предсказанная дискриминатором вероятность того, что образец G(z), сгенерированный по шуму z, является настоящим.

2.2 Условные генеративно-состязательные сети

Условные генеративно-состязательные сети (Conditional GAN) [20] являются модификацией классического GAN. В данном алгоритме на вход генератору подается не только шум z, но и любая дополнительная информация, например, номер класса, образец которого требуется сгенерировать, или входное изображение, поданное генератору.

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))]$$

Рис. 2. Формула целевой функции для генератора и дискриминатора в алгоритме условных генеративно-состязательных сетей. D(x|y) - предсказанная дискриминатором вероятность того, что “реальный” образец x является настоящим при условии объекта y, а D(G(z|y)) - предсказанная дискриминатором вероятность того, что образец G(z|y), сгенерированный по шуму z при условии объекта y, является настоящим.

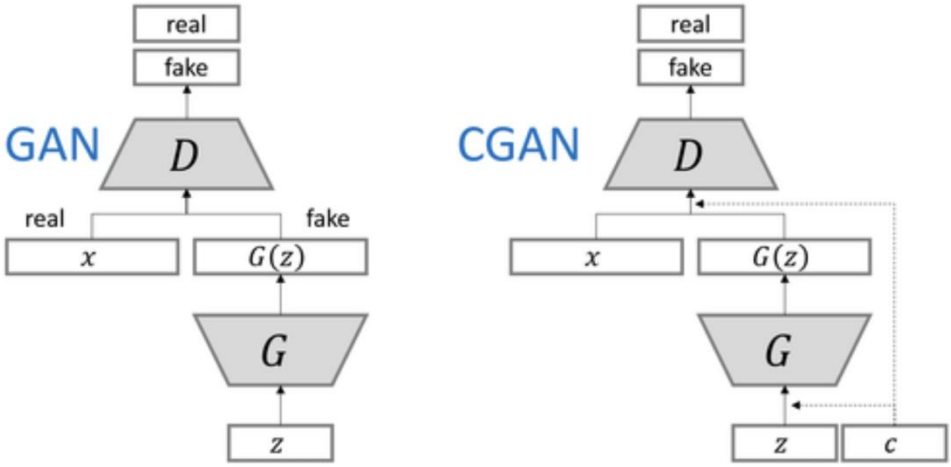

Рис. 3. Сравнение принципа работы генеративно-состязательных сетей (слева) и условных генеративно-состязательных сетей (справа) [19].

2.3 pix2pix

pix2pix [13] - подход отображения изображений в другие домены в случае наличия парных данных, основанный на условных генеративно-состязательных сетях. В данном алгоритме генератор принимает на вход исходное изображение, которое необходимо отобразить в целевой домен, а дискриминатор принимает на вход пару изображений - (сгенерированный объект в целевом домене; объект в исходном домене). Таким образом, исходный объект x виден и генератору, и дискриминатору, в отличие от алгоритма классических генеративно-состязательных сетей.

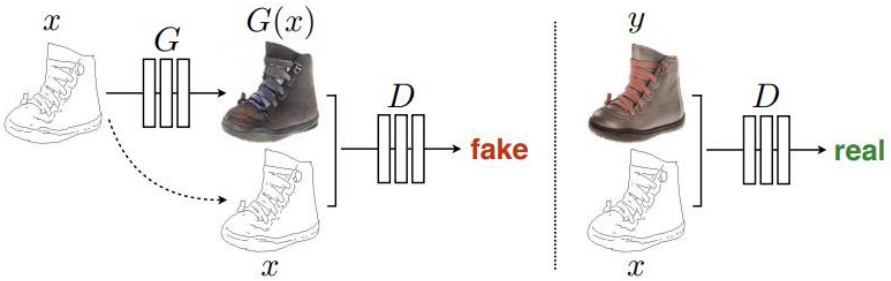

Рис. 4. Алгоритм pix2pix на примере задачи генерации реалистичного изображения обуви по простому рисунку-контуру [13].

Дополнительно, pix2pix также заменяет классический дискриминатор, выдающий одно число - вероятность “подлинности” поданного ему на вход изображения - по входным данным, на PatchGAN дискриминатор, который выдает матрицу, каждое значение которой - вероятность “подлинности” некоторой локальной части размера N x N (N - гиперпараметр) поданного ему на вход изображения, причем части могут пересекаться. Такая модификация дискриминатора выступает в роли некоторой “текстурной” или “стилевой” функции потерь.

За счет данных улучшений pix2pix может достичь лучшего качества сгенерированных изображений чем условные генеративно-состязательные сети. Однако ключевым недостатком данного алгоритма является полная зависимость на наличии парных данных, ведь для большинства задач отображения изображений в другие домены таких данных не существует. Именно эту проблему пытается решить CycleGAN [31].

2.4 CycleGAN

CycleGAN - подход отображения изображений в другие домены, основанный на генеративно-состязательных сетях и циклической функции потерь и не требующий наличия парных данных.

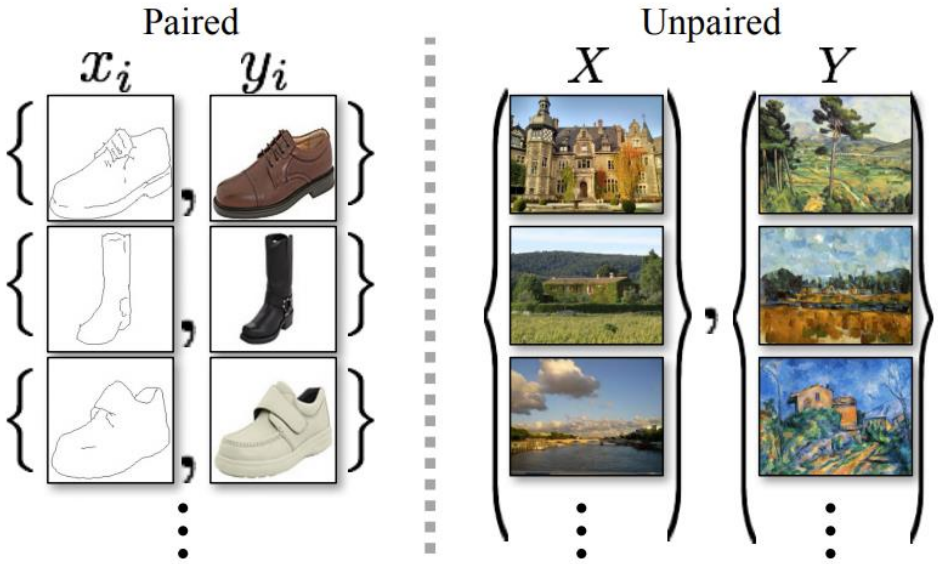

Рис. 5. Сравнение парных данных и непарных данных. В первом случае для каждого образца x_i из исходного домена X существует его представление y_i в целевом домене Y, во втором случае - нет [31].

В данном алгоритме участвуют уже две пары генератора и дискриминатора. Генератор G: X → Y переводит объект из домена X в его представление в домене Y, а генератор F: Y → X переводит объект из домена Y в его представление в домене X. Дискриминатор D_Y стимулирует генератор G создавать изображения, неразличимые с настоящими изображениями из домена Y, а дискриминатор D_X стимулирует генератор F создавать изображения, неразличимые с настоящими изображениями из домена X. В дополнение к этому алгоритм вводит прямую и обратную циклическую функцию потерь, которая мотивирует последовательное применение двух генераторов выдавать объект, максимально близкий к исходному: x → G(x) → F(G(x)) ≈ x и y → F(y) → G(F(y)) ≈ y. Вдобавок к циклической функции потерь в некоторых случаях используется тождественная функция потерь, которая мотивирует генератор не менять изображение, если оно уже из целевого домена: x → F(x) ≈ x и y → G(y) ≈ y. В качестве дискриминатора также выступает PatchGAN с N = 70.

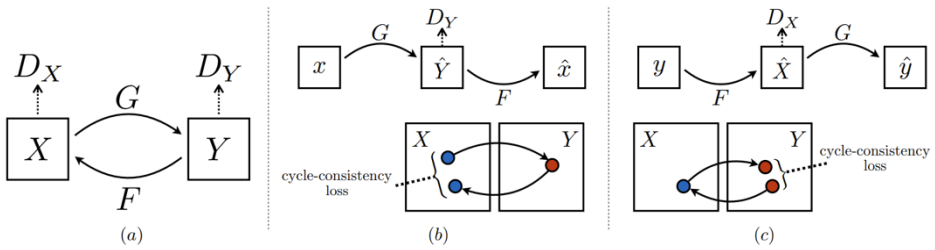

Рис. 6. Демонстрация (a) общей работы алгоритма, (b) прямой циклической функции потерь, (c) обратной циклической функции потерь [31].

2.5 U-GAT-IT

U-GAT-IT [17] - более современный подход отображения изображений в другие домены, основанный на механизме внимания, вспомогательном классификаторе и усовершенствованном слое нормализации. Он также не требует наличия парных данных.

В данном алгоритме также участвуют уже две пары генератора и дискриминатора, как и в CycleGAN, однако во все модели добавляется механизм внимания. Дополнительно, с помощью вспомогательного классификатора, выдающего для генератора и дискриминатора “карты внимания”, модели учатся сосредотачиваться на более семантически важных участках изображения, позволяя большую гибкость в трансформации формы на изображении.

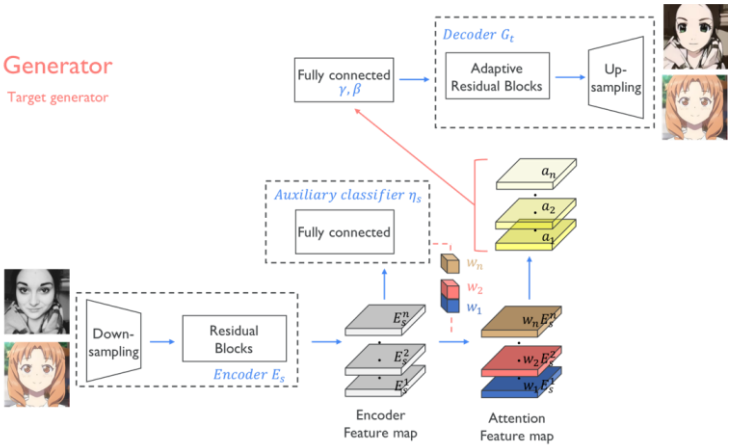

Рис. 7. Схема работы генератора [17].

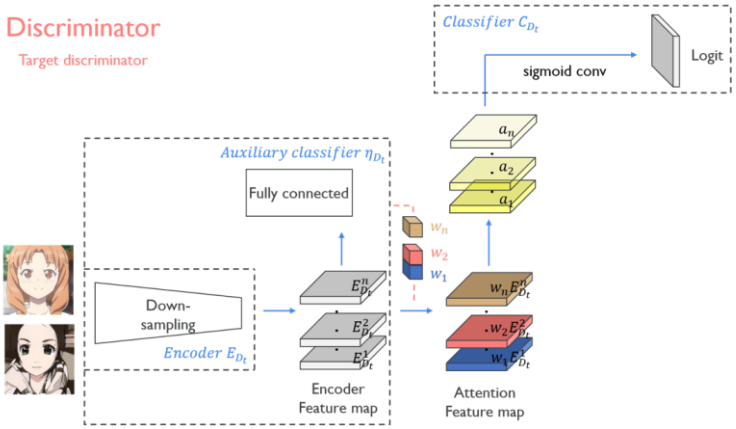

Рис. 8. Схема работы дискриминатора [17].

Помимо механизма внимания, в модели появляется новый слой нормализации - Adaptive Layer-Instance Normalization, который динамически считает необходимое отношение между индивидуальной нормализацией (Instance Normalization) [29] и слоевой нормализацией (Layer Normalization) [18].

Также в общую функцию потерь, помимо генеративно-состязательной, циклической и тождественной ошибки, для вспомогательных классификаторов добавляется функция потерь CAM (Class Activation Mapping) [30], с помощью которой модели могут оценить наибольшие расхождения между двумя доменами в текущем состоянии.

3 Сбор и обработка данных

На момент проведения данного исследования в интернет-ресурсах отсутствовали публичные датасеты с лицами из сериала Arcane, которые критически важны для обучения. В связи с этим было необходимо собрать собственный набор лиц для обучения моделей.

Для сбора датасета были использованы видео всех девяти серий сериала, а также музыкальное видео Enemy by Imagine Dragons [5]. Все видео были взяты в разрешении 3840x2160. Каждое видео было обработано отдельно путем сохранения покадрово. Лица из вступления, а также сцен сериала с существенно другим анимационным стилем не брались.

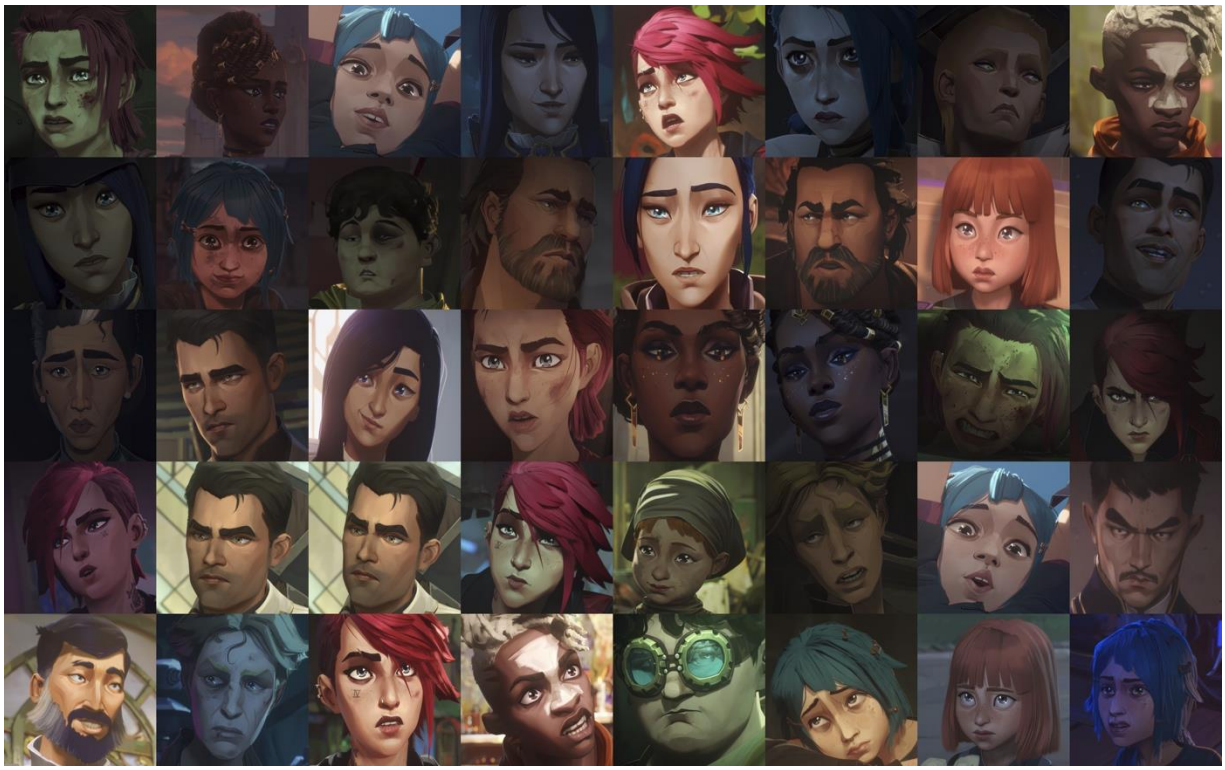

Рис. 9. Примеры лиц, снятых из кадров сериала.

Лица сохранялись при наличии хотя бы минимального освещения и видимости обоих глаз, а также при отсутствии физических деформаций, существенно нестандартных ракурсов и размытости кадра. Большой акцент делался также на сохранение лиц в виде изображений, как можно более близких по разрешению к квадратному, а также разнообразие выражений и эмоций лиц.

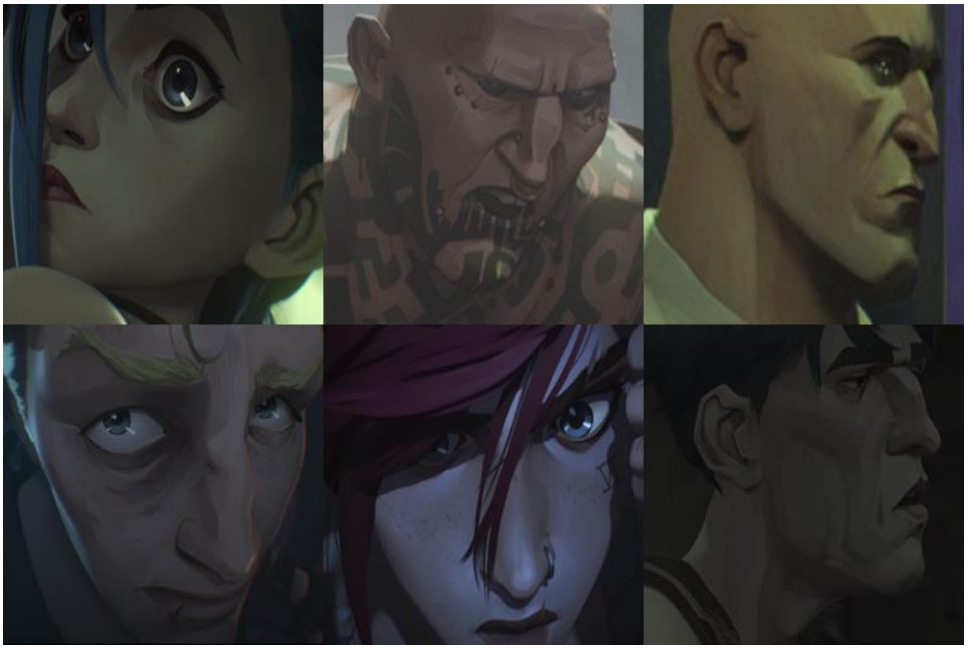

Рис. 10. Примеры лиц, непригодных для датасета.

По итогам сбора датасета было получено свыше 3400 изображений. Соотношение измерений изображений находилось в диапазоне от 1.00 до 1.40. Наименее квадратные изображения (соотношение не меньше 1.25), были обработаны вручную для понижения соотношения сторон. Итоговые характеристики изображений датасета представлены в таб. 1.

	Высота (пиксели)	Ширина (пиксели)	Соотношение измерений
mean	675.489	731.736	1.087
std	198.376	228.333	0.054
min	184	185	1.000
10%	399	426	1.016
20%	488	513	1.034
30%	563	598	1.050
40%	625	667	1.065
50%	679	729	1.081
60%	734	788	1.098
70%	792	860	1.116
80%	862	936	1.137
90%	942	1044	1.164
max	1081	1272	1.219

Таблица 1. Характеристики собранного датасета после его обработки.

4 Обучение на непарных данных

Для получения начального базового решения была обучена полноценная U-GAT-IT модель на 750 000 итераций с размером батча 1. В качестве исходного домена выступал датасет реальных лиц Flickr-Faces HQ (FFHQ) [6], содержащий 70 000 изображений в разрешении 1024x1024, а в качестве целевого домена - собранный ранее датасет лиц из сериала Arcane.

После 750 000 итераций модель не смогла осуществить перенос лиц из исходного домена в целевой даже на тестовой выборке датасета, что отчетливо видно на рис. 11.

Рис. 11. Результаты на тестовой выборке модели U-GAT-IT, обученной в течение 750 000 итераций.

Таким образом, данный эксперимент показал, что для решения поставленной задачи даже современные методы непарного обучения не могут обеспечить необходимое качество отображения в наш целевой домен. Поэтому в качестве альтернативного подхода к разрешению задачи было решено составить парный датасет для обучения с помощью блендинга [23] StyleGAN [15] моделей.

5 Генерация парных данных

StyleGAN — это архитектура генеративно-состязательной сети, предназначенная для генерации изображений высокого разрешения с возможностью контролирования как высокоуровневых, так и низкоуровневых черт генерируемого изображения.

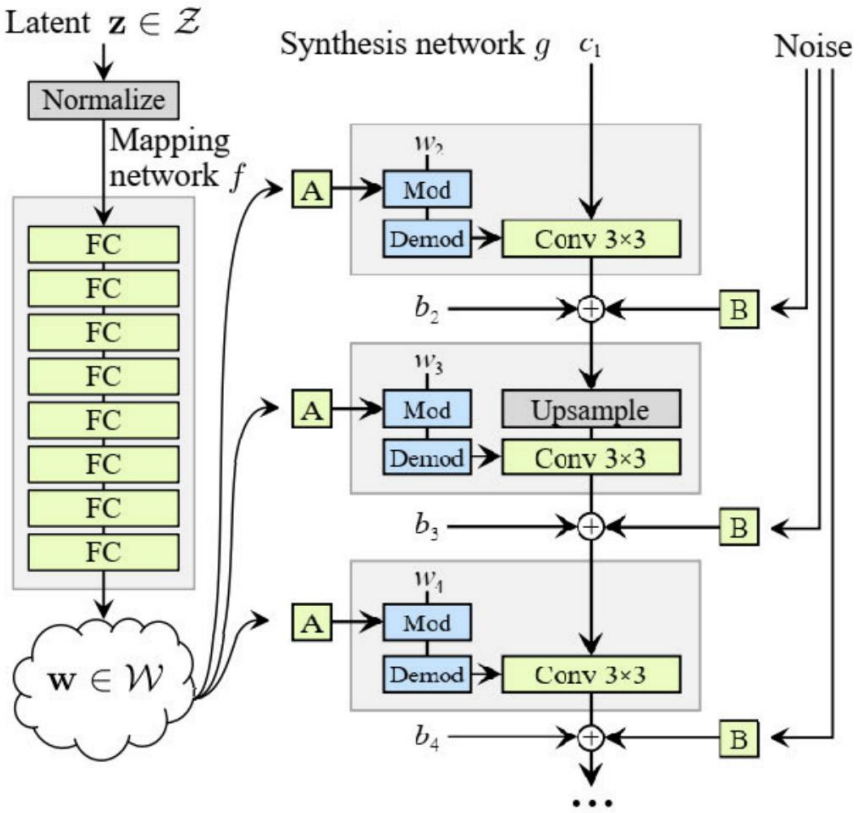

Рис. 12. Схема работы архитектуры StyleGAN2 [16].

Так, изменение параметров стиля на самых высоких разрешениях модели приводит, например, к появлению родинок, веснушек, изменения на средних слоях - к другому выражению лица, а изменения на самых ранних слоях - к другой позе, форме лица.

Одним из возможных способов создания парных данных для обучения при наличии достаточного количества примеров из целевого домена является блендинг нескольких моделей StyleGAN. Блендинг основан на интерполяции весов двух (или более) моделей, каждая из которых обучена для генерации изображений из своего конкретного домена. Так, например, взяв модели, обученные генерировать реалистичные лица и лица в стиле укиё-э, и составив третью модель, веса низкого разрешения которой взяты из первой модели, а веса высокого разрешения - из второй, возможно сгенерировать лицо, поза и фигура лица которой будут совпадать с реалистичным лицом, но также и обладать текстурным стилем из укиё-э (рис. 13). Таким образом, блендинг StyleGAN моделей дает возможность создать парный датасет, в котором объектом исходного домена выступает сгенерированное реалистичное лицо, а объектом целевого домена - лицо, сгенерированное моделью со смешанными весами.

Рис. 13. Сгенерированное реалистичное лицо (слева), сгенерированное лицо в стиле укиё-э (справа) и лицо, сгенерированное путем смешивания весов обеих моделей (посередине) [23].

Однако для генерации парного датасета необходима модель, обученная для генерации изображений из нашего целевого домена. Для получения такой модели было решено взять конфигурацию StyleGAN2, предобученную на датасете FFHQ в течение 25 000 000 итераций для генерации изображений размером 256x256, и дообучить ее на собранном датасете лиц из сериала Arcane в течение 200 000 итераций. Веса модели сохранялись каждые ~10 000 итераций.

По промежуточным результатам обучения модели стали очевидны следующие наблюдения. После 20 000 итераций модель уже способна генерировать лица адекватного качества с высоким разнообразием, однако текстура кожи довольно гладкая и нуждается в усовершенствовании (рис. 14). После 70 000 итераций текстура кожи уже дошла до приемлемого уровня, однако разнообразие лиц начинает уменьшаться (рис. 15). После 151 000 итерации модель существенно переобучается, значительно теряя разнообразие и при этом не улучшая качество текстуры лиц (рис. 16). По этим наблюдениям было решено взять промежуточные веса модели после 20 000 и 70 000 итераций.

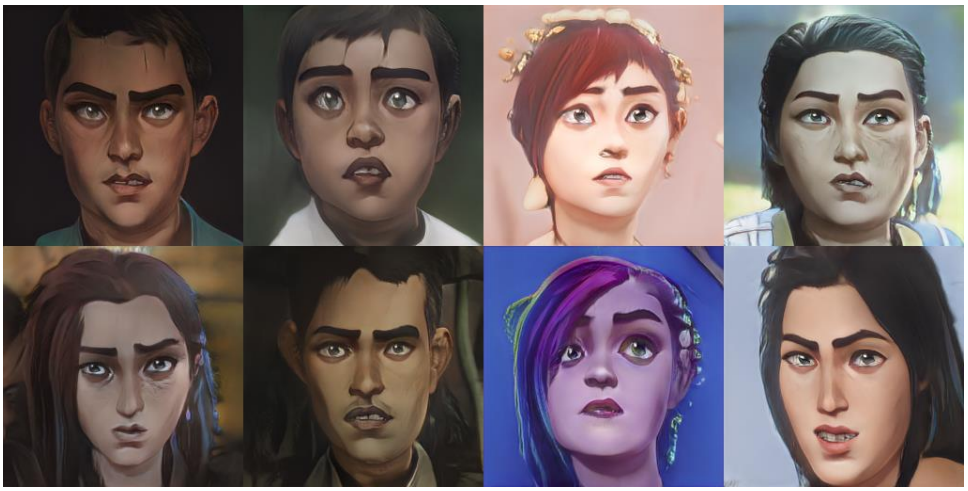

Рис. 14. Сгенерированные дообученной моделью лица после 20 000 итераций.

Рис. 15. Сгенерированные дообученной моделью лица после 70 000 итераций.

Рис. 16. Сгенерированные дообученной моделью лица после 151 000 итераций.

Для итогового блендинга двух моделей веса низких разрешений (4x4 - 8x8) для генерации позы, фигуры лица и прочих высокоуровневых деталей полностью брались из модели, обученной на FFHQ, средние веса (16x16 - 32x32) брались в равном соотношении, а веса высоких разрешений (64x64 - 256x256) брались преимущественно из модели, дообученной на нашем датасете.

Смешав веса модели, обученной на FFHQ, и нашей модели, дообученной на 20 000 итерациях, в вышеописанных соотношениях, нам удалось получить результаты, показанные на рис. 17. Видно, что, несмотря на уже приемлемое качество смешанных лиц, их текстура все еще имеет некоторые артефакты в виде царапин, а некоторые небольшие детали (очки, локальные куски волос) могут сильно меняться.

Рис. 17. Парные данные, сгенерированные с весами дообученной модели, взятыми после 20 000 итераций.

Ориентируясь на эти наблюдения и взяв веса модели, дообученной после 70 000 итераций, немного модифицировав параметры смешивания, удалось добиться существенно более высокого качества парных данных. Так, удалось минимизировать ранее упомянутые артефакты-”царапины”, а также удалось добиться лучшего выравнивания лиц (рис. 18).

Рис. 18. Парные данные, полученные сгенерированные с весами дообученной модели, взятыми после 70 000 итераций.

По итогу блендинга моделей было решено взять модель, обученную на 70 000 итераций, для генерации парного датасета. С ее помощью было сгенерировано 10 000 пар лиц для обучения. Однако предварительный осмотр сгенерированных данных выявил критическую проблему: из-за большого объема цветных волос лиц в целевом домене, существенная часть полученных изображений оказалась с покрашенными волосами, как видно на рис. 19. После осмотра всех 10 000 пар данных, проблемными оказались около 43% всех лиц, то есть приблизительно 4300 пар, непригодных для обучения.

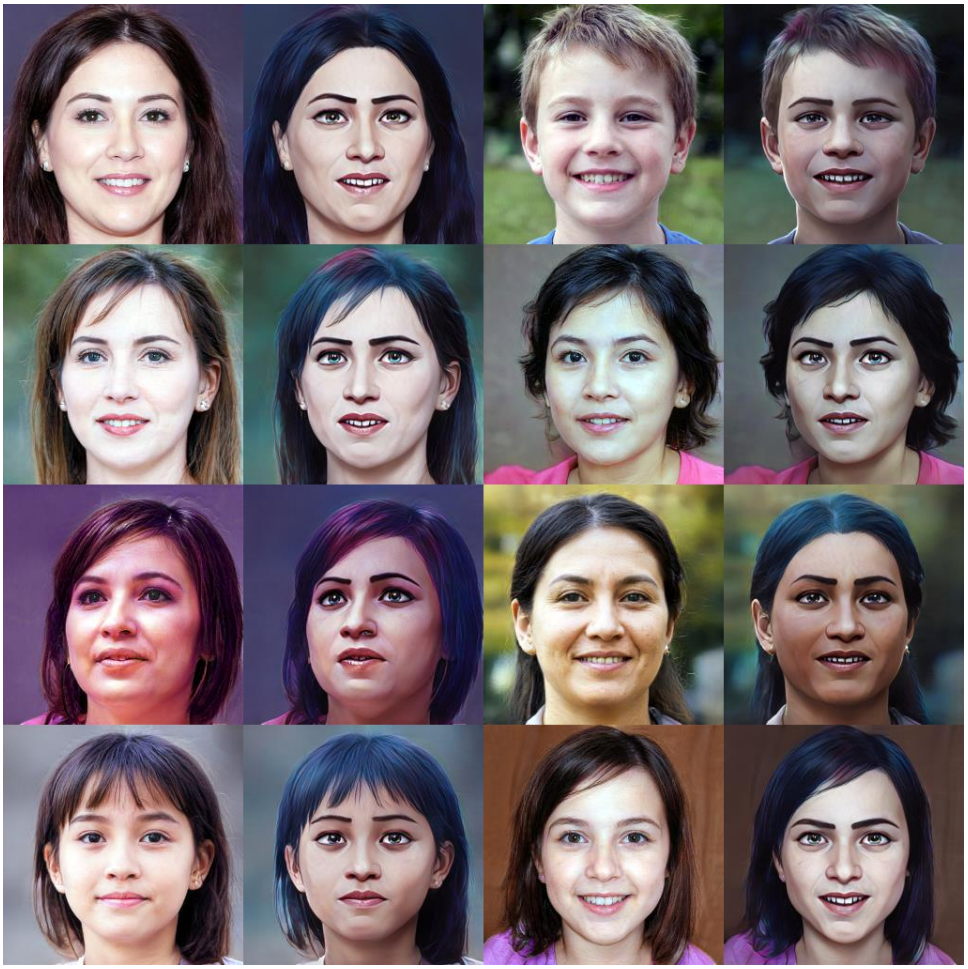

Рис. 19. Случаи сильной дисколорации волос в сгенерированных парных данных.

После удаления проблемных лиц, к оставшимся 5700 парам была применена модель повышения разрешения Real-ESRGANx2 [24], обученная специально на лицах (рис. 20), для повышения их разрешения до 512x512.

Таким образом, удалось получить около 5700 пар лиц разрешения 512x512 и приемлемого качества, пригодных для обучения. Для создания базового решения с помощью парного обучения было решено взять подход pix2pix.

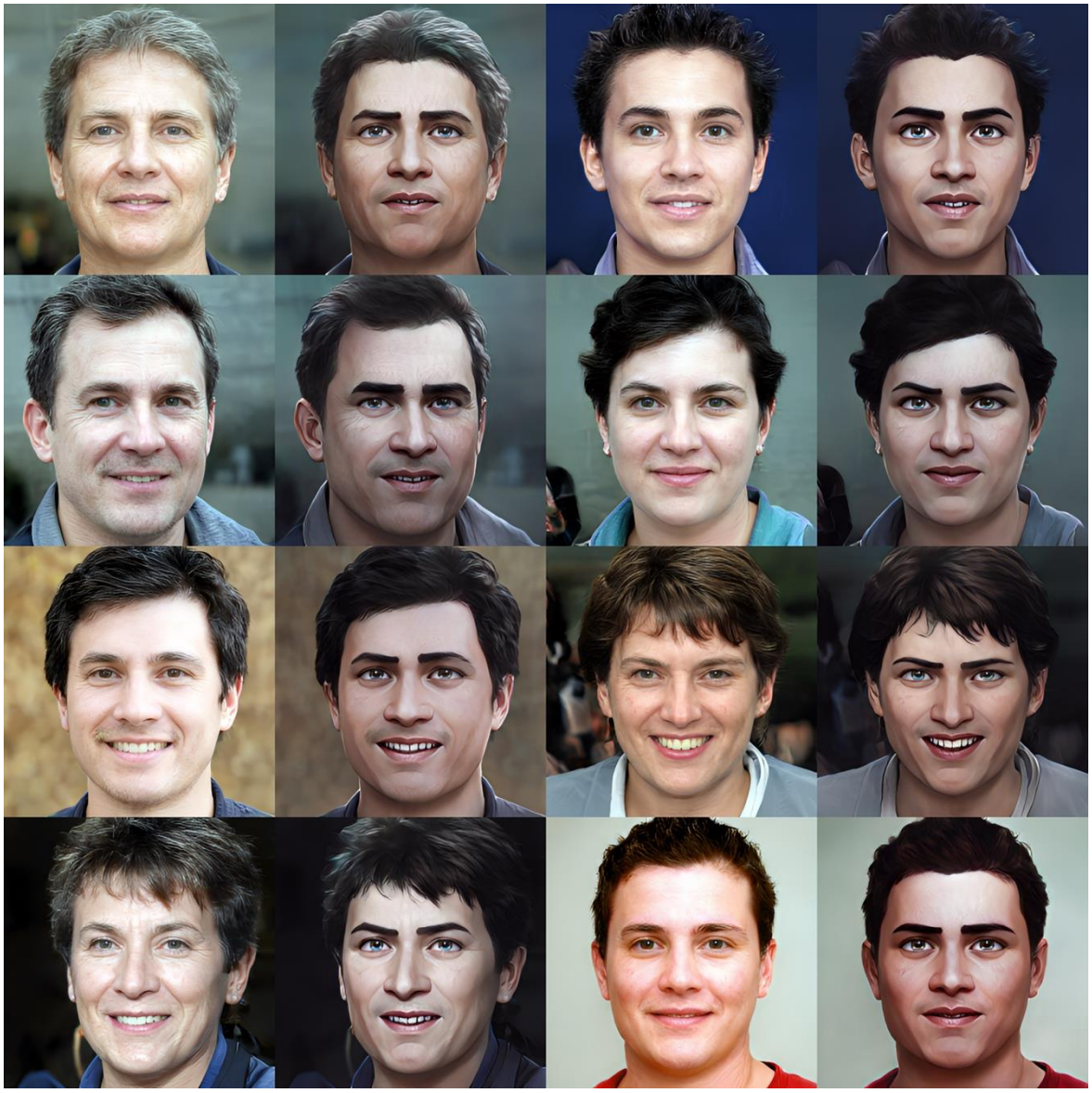

Рис. 20. Увеличенные с помощью RealESRGANx2 парные данные для обучения.

6 Обучение с pix2pix

В качестве базового решения была обучена модель pix2pix на разрешении 256x256 с архитектурой “resnet6_blocks” [31] и размером батча 1 на протяжение 427 500 итераций. Набор аугментаций состоял из горизонтального отражения, а также загрузки изображения в разрешении 286x286 и взятии случайного кропа размером 256x256. Однако применение полученной модели на тестовой выборке, состоящей из изображений лиц, взятых из интернет-ресурсов, выявило сразу несколько существенных проблем.

Несмотря на то, что модель показывает адекватное качество на изображениях того же разрешения и позиции / размера лица (рис. 21), даже небольшие изменения в размере или позиции лица сразу же кардинально снижают качество работы модели, как отчетливо видно на рис. 22.

Рис. 21. Пример высокого качества работы модели на изображениях размера 256x256 из тестовой выборки.

Рис. 22. Пример типичного качества работы модели на изображениях разрешения 256x256 из тестовой выборки.

Дополнительно, качество модели проседает еще сильнее при обработке изображений более высоких разрешений, как видно на рис. 23 и 24.

Рис. 23. Пример результата модели на изображении разрешения 1200x1200.

Рис. 24. Пример результата модели на изображении разрешения 1024x736.

Наиболее вероятные причины ухудшения качества - практически полное отсутствие аугментаций во время обучения (рис. 25) и относительно небольшое количество параметров модели (7.841M параметров), не позволяющее модели достичь нужного уровня обобщения.

Рис. 25. Аугментации базового решения: загрузка изображения в разрешении 286x286, горизонтальное отражение с вероятностью 0.5, случайный кроп 256x256.

Для борьбы с обеими проблемами было решено взять архитектуру U-Net [25] (54.413M параметров), а также существенно увеличить объем аугментаций во время обучения модели (рис. 26).

Рис. 26. Пример расширенного набора аугментаций для обучения модели. Аугментации: горизонтальное отражение (p=0.5), афинное преобразование (p=0.75), случайное масштабирование (p=0.75).

Изменив вышеописанные параметры обучения, нам удалось получить модель, которая значительно лучше обрабатывает нестандартные ракурсы и позы в том же разрешении, в котором она обучалась (рис. 27), однако качество применения на более высоких разрешениях осталось прежним (рис. 28). Помимо этого, также стали явно проявляться артефакты “вафельного полотенца” [22] (рис. 28, 29), связанные с появлением в архитектуре слоев “транспонированной свертки” [1]. Наконец, данная имплементация архитектуры U-Net не поддерживает изображения произвольных разрешений (рис. 29), что сильно ограничивает возможности ее применения для обработки любой входной фотографии.

Рис. 27. Результаты применения новой модели к изображениям с нестандартными ракурсами в разрешении 256x256 из тестовой выборки.

Рис. 28. Результаты применения новой модели к изображениям разрешения 1200x1200. Помимо артефактов вафельного полотенца модель выдает изображение в другом соотношении сторон: 1280x1280.

Рис. 29. Результат работы модели на изображении разрешения 1000x563. Помимо артефактов вафельного полотенца модель выдает изображение в другом соотношении сторон: 1024x512.

Ввиду вышеописанных проблем использование U-Net архитектуры, описанной в pix2pix, оказалось невозможным для нашей задачи, в связи с чем было решено использовать архитектуру динамической U-Net, а также изменить другие параметры обучения для его улучшения.

7 Улучшенный подход к обучению

Динамическая U-Net обладает несколькими преимуществами перед классической имплементацией U-Net, использующейся в pix2pix. Так, во-первых, в качестве энкодера модели может быть использована любая предобученная сверточная архитектура, что позволяет увеличить и улучшить качество извлекаемых признаков изображения. Во-вторых, в декодере вместо “транспонированной свертки” используется метод PixelShuffle [26] для выполнения апсемплинга, который позволяет минимизировать количество наблюдаемых артефактов “вафельного полотенца”, тем самым повышая визуальное качество изображения. По этим причинам архитектура модели была заменена на нее. В качестве энкодера была взята модель ResNet34 [8], предобученная на крупном датасете изображений ImageNet [10]. Общее количество параметров модели составило 41.413M.

Дополнительно, в аугментации данных было добавлено большее случайное масштабирование (до четырехкратного) и больший максимальный уровень поворота (до 45 градусов), а также при афинных преобразованиях области за краями изображения зеркально дополняются (рис. 30).

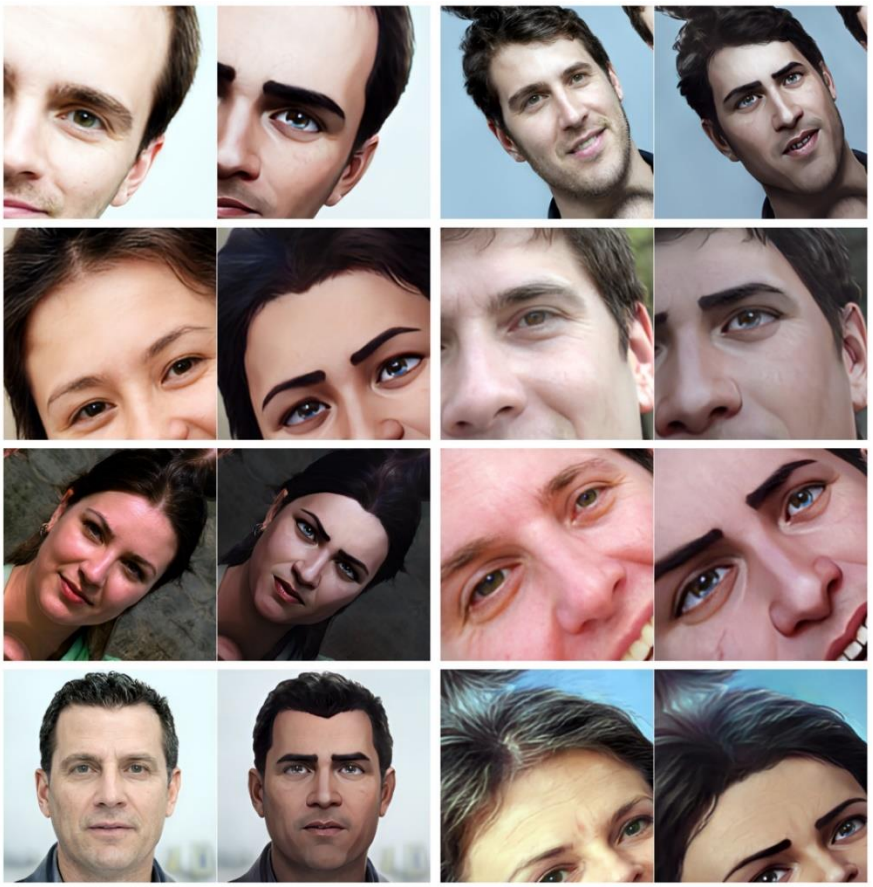

Рис. 30. Пример новых улучшенных аугментаций.

Еще одна модификация обучения - добавление к функции потерь дистанции (средняя абсолютная попиксельная ошибка) также “ошибки восприятия” (perceptual loss) [14]. “Ошибка восприятия” мотивирует модель генерировать более качественное изображение за счет не только попиксельного сравнения результата с целевым изображением, но также и их карт признаков, взятых со слоев разной глубины фиксированной предобученной модели. В качестве модели для извлечения признаков была взята VGG-16 [27], предобученная на датасете ImageNet.

Обучив модель с вышеперечисленными изменениями, удалось добиться качества, уровень которого оказался существенно выше любой модели, полученной до этого. Так, итоговая модель не только способна принимать на вход изображения, но и полностью избавилась от артефактов “вафельного полотенца” (рис. 31–33).

Рис. 31. Слева: оригинал (разрешение 413x531), справа: результат работы улучшенной модели, посередине: результат работы лучшей pix2pix модели.

Рис. 32. Слева: оригинал (разрешение 1024x736), справа: результат работы улучшенной модели, снизу: результат работы лучшей pix2pix модели.

Рис. 33. Слева: оригинал (разрешение 1200x1200), справа: результат работы улучшенной модели, снизу: результат работы лучшей pix2pix модели.

Рис. 34. Слева: оригинал (1000x563), справа: результат работы улучшенной модели, снизу: результат работы лучшей pix2pix модели.

8 Ускорение модели и уменьшение ее размера

Так как наша модель предназначена для работы на мобильных устройствах и серверах с ограниченными количествами вычислительных ресурсов, помимо высокого уровня качества модель должна быть эффективной по времени работы над изображением. В связи с этими требованиями дополнительно было решено обучить модель с меньшим энкодером - ResNet18.

Обучив модель с меньшим экнодером с идентичными гиперпараметрами, нам удалось получить результат, визуально практически неотличимый от предыдущей модели (рис. 35-39), а в некоторых случаях - даже лучшего качества.

Рис. 35. Слева: оригинал, посередине: результат работы модели с энкодером ResNet34, справа: результат работы модели с энкодером ResNet18.

Рис. 36. Слева: оригинал, посередине: результат работы модели с энкодером ResNet34, справа: результат работы модели с энкодером ResNet18.

Рис. 37. Сверху: оригинал, слева: результат работы модели с энкодером ResNet34, справа: результат работы модели с энкодером ResNet18.

Рис. 38. Сверху: оригинал, слева: результат работы модели с энкодером ResNet34, справа: результат работы модели с энкодером ResNet18.

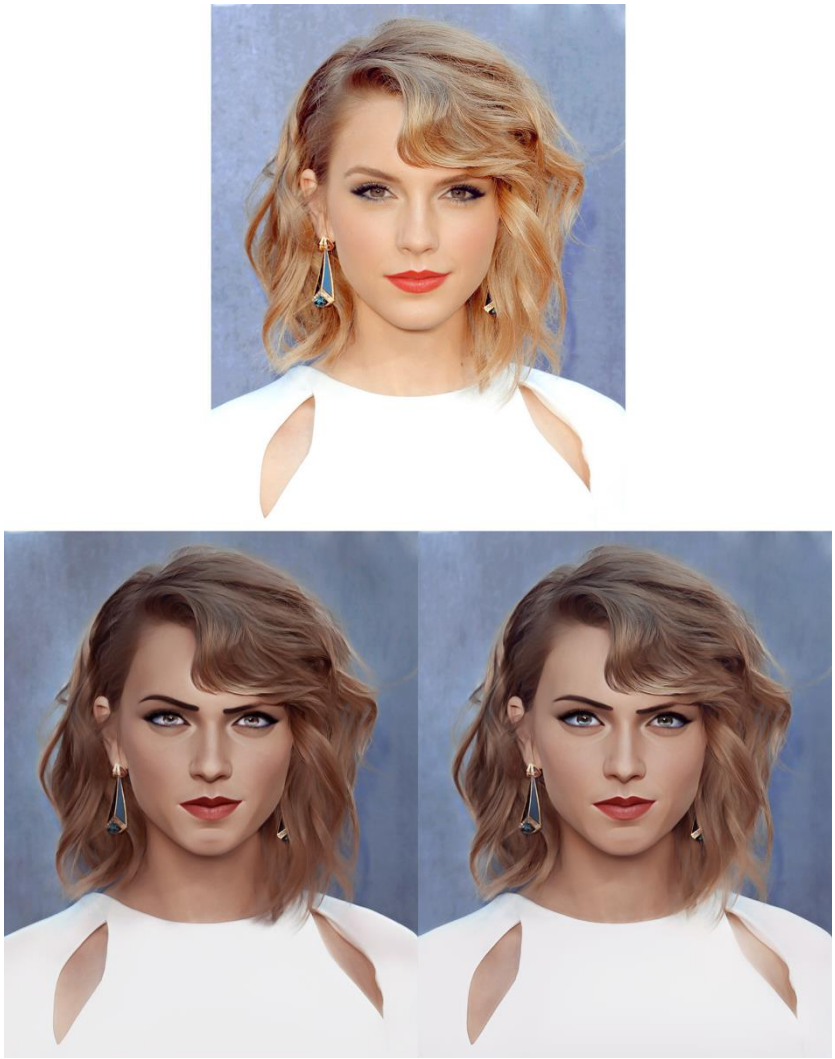

Рис. 39. Сверху: оригинал, слева: результат работы модели с энкодером ResNet34, справа: результат работы модели с энкодером ResNet18.

Дополнительно, нами было проведено сравнение эффективности работы двух моделей, основанных на энкодерах ResNet18 и ResNet34. Для этого каждая модель на каждом разрешении и каждом целевом девайсе (CPU, GPU) была прогнана 1000 итераций для CPU и 10 000 итераций для GPU на одной картинке, причем 5% самых быстрых и 5% самых долгих замеров удалялись, после чего бралось среднее замеров. Результаты сравнения и описание физических характеристик системы, на котором проводились тесты, представлены в таб. 2.

		CPU, сек.			GPU, сек.		
	Размер, МБ	256x256	512x512	1024x1024	256x256	512x512	1024x1024
ResNet18	244	0.106	0.491	2.993	0.0059	0.0200	0.0826
ResNet34	284	0.111	0.507	3.044	0.0066	0.0207	0.0830

Таблица 2. Сравнение времени работы (сек.) и размера (МБ.) динамических U-Net моделей с разными энкодерами. CPU – AMD Ryzen 9 5900x, GPU – EVGA GeForce RTX 3080 Ti FTW3 Ultra, OS – Ubuntu 20.04.

По результатам эффективности видно, что использование меньшего энкодера приводит к уменьшению размера модели в ~1.17 раз, время работы на CPU уменьшается на значения до ~1.05 раз, а время работы на GPU уменьшается до ~1.12 раз.

В заключение мы сравнили нашу итоговую модель, обученную с декодером ResNet18, и ArcaneGAN [3] - единственный существующий в интернет-ресурсах сервис для отображения лиц в домен Arcane. Результаты сравнения представлены на рис. 40-42. Так, видно, что модель ArcaneGAN не всегда справляется с нестандартными ракурсами (рис. 40), а также имеет заметные артефакты в виде раскрашивания волос и размывания зрачков (рис. 41, 42), в отличие от нашей модели.

Рис. 40. Слева: оригинал, посередине: ArcaneGAN, справа: наша U-Net модель.

Рис. 41. Сверху: оригинал, слева: ArcaneGAN, справа: наша U-Net модель.

Рис. 42. Сверху: оригинал, слева: ArcaneGAN, справа: наша U-Net модель.

9 Выводы

Таким образом, были изучены широко применяемые в задаче отображения изображений в другие домены модели генеративно-состязательных сетей и алгоритмы, основанные как на парном обучении, так и на непарном обучении. Экспериментальным путем было установлено, что подход непарного обучения не способен достичь приемлемого качества в нашей задаче. Помимо этого, были проведены эксперименты, на основе которых удалось сравнить эффективность разных архитектур моделей, выявить артефакты, вносимые ими, а также устранить их.

Дополнительно, был собран и опубликован первый крупный публичный датасет лиц из сериала Arcane, состоящий из свыше 3400 изображений высокого качества, который может применяться как для генерации новых лиц в стиле сериала, так и для создания парных данных для обучения.

В заключение, была разработана модель, позволяющая добиться более высокого качества в задаче отображения произвольных изображений в стиль анимационного сериала Arcane, чем у существующего решения в интернет-ресурсах ArcaneGAN, возможная для развертывания на мобильных устройствах или серверах с существенно ограниченным объемом вычислительных ресурсов.

10 Направления дальнейшей работы

В качестве направлений для дальнейшей работы можно выделить несколько ключевых:

- обучение модели с темпоральной консистентностью для стабильной обработки видео;
- добавление контроля над интенсивностью стиля итогового изображения для возможности изменения силы стиля в реальном времени;
- одновременное применение нескольких стилей к итоговому изображению;
- дальнейшее ускорение работы модели и уменьшение ее размера для возможности загрузки и использования сразу нескольких моделей.

Список использованных источников

1. Anwar, A. What is Transposed Convolutional Layer? / A. Anwar // [Электронный ресурс]: towardsdatascience, 2020 - Режим доступа: https://towardsdatascience.com/what-is-transposed-convolutional-layer-40e5e6e31c11#:~:text=Transposed%20convolutions%20are%20standard%20convolutions,in%20a%20standard%20convolution%20operation., свободный. (дата обращения: 11.12.21)

2. Arcane (TV Series) [Электронный ресурс] / Википедия. Режим доступа: https://en.wikipedia.org/wiki/Arcane_(TV_series), свободный. (дата обращения: 25.12.21)

3. ArcaneGAN [Электронный ресурс] / Hugging Face. Режим доступа: https://huggingface.co/spaces/akhaliq/ArcaneGAN, свободный. (дата обращения: 03.03.22)

4. Colorization [Электронный ресурс] / paperswithcode. Режим доступа: https://paperswithcode.com/task/colorization, свободный. (дата обращения: 10.12.21)

5. Enemy (Imagine Dragons and JID song) [Электронный ресурс] / Википедия. Режим доступа: https://en.wikipedia.org/wiki/Enemy_(Imagine_Dragons_and_JID_song), свободный. (дата обращения: 25.12.21)

6. ffhq-dataset [Электронный ресурс] / GitHub. Режим доступа: https://github.com/NVlabs/ffhq-dataset, свободный. (дата обращения: 17.02.21)

7. Goodfellow, I. Generative Adversarial Networks / I. Goodfellow, J. Pouget-Abadie, M Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio // [Электронный ресурс]: arXiv, 2014 - Режим доступа: https://arxiv.org/abs/1406.2661, свободный (дата обращения: 10.01.22)

8. He, K. Deep Residual Learning for Image Recognition / K. He, X. Zhang, S. Ren, and J. Sun. // [Электронный ресурс]: arXiv, 2015 - Режим доступа: https://arxiv.org/abs/1512.03385, свободный (дата обращения: 12.12.21)

9. Image Inpainting [Электронный ресурс] / paperswithcode. Режим доступа: https://paperswithcode.com/task/image-inpainting, свободный. (дата обращения: 16.11.21)

10. Image Reconstruction [Электронный ресурс] / paperswithcode. Режим доступа: https://paperswithcode.com/task/image-reconstruction, свободный. (дата обращения: 16.11.21)

11. Image Segmentation [Электронный ресурс] / Википедия. Режим доступа: https://en.wikipedia.org/wiki/Image_segmentation, свободный. (дата обращения: 16.11.21)

12. ImageNet [Электронный ресурс] / ImageNet. Режим доступа - https://image-net.org/, свободный. (дата обращения: 12.12.21)

13. Isola, P. Image-to-Image Translation with Conditional Adversarial Networks / P. Isola, J.

Zhu, T. Zhou, and A. Efros // [Электронный ресурс]: arXiv, 2016 - Режим доступа: https://arxiv.org/abs/1611.07004, свободный (дата обращения: 13.01.22)

14. Johnson, J. Perceptual Losses for Real-Time Style Transfer and Super-Resolution / J. Johnson, A. Alahi, L. Fei-Fei // [Электронный ресурс]: arXiv, 2016 - Режим доступа: https://arxiv.org/abs/1603.08155, свободный. (дата обращения: 20.01.22)

15. Karras, T. A Style-Based Generator Architecture for Generative Adversarial Networks / T. Karras, S. Laine, and T. Aila // [Электронный ресурс]: arXiv, 2018 - Режим доступа: https://arxiv.org/abs/1812.04948, свободный (дата обращения: 02.03.22)

16. Karras, T. Analyzing and Improving the Image Quality of StyleGAN / T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila // [Электронный ресурс]: arXiv, 2019 - Режим доступа: https://arxiv.org/abs/1912.04958, свободный. (дата обращения: 02.03.22)

17. Kim, J. U-GAT-IT: Unsupervised Generative Attentional Networks with Adaptive Layer-Instance Normalization for Image-to-Image Translation / J. Kim, M. Kim, H. Kang, and K. Lee // [Электронный ресурс]: arXiv, 2019 - Режим доступа: https://arxiv.org/abs/1907.10830, свободный (дата обращения: 20.01.22)

18. Lei J. Layer Normalization / J. Lei Ba, J. Ryan Kiros, and G. Hinton // [Электронный ресурс]: arXiv, 2016 - Режим доступа: https://arxiv.org/abs/1607.06450, свободный (дата обращения: 27.01.22)

19. Mino, A. LoGAN: Generating Logos with a Generative Adversarial Neural Network Conditioned on color / A. Mino, and G. Spanakis // [Электронный ресурс]: arXiv, 2018 - Режим доступа: https://arxiv.org/abs/1810.10395, свободный (дата обращения: 14.01.22)

20. Mirza, M. Conditional Generative Adversarial Nets / M. Mirza, and S. Osindero // [Электронный ресурс]: arXiv, 2014 - Режим доступа: https://arxiv.org/abs/1411.1784, свободный (дата обращения: 11.01.22)

21. Neural Style Transfer [Электронный ресурс] / Википедия. Режим доступа: https://en.wikipedia.org/wiki/Neural_style_transfer, свободный. (дата обращения: 16.11.21)

22. Odena, A. Deconvolution and Checkerboard Artifacts / A. Odena, V. Dumoulin, and C. Olah // [Электронный ресурс]: Distill, 2016 - Режим доступа: https://distill.pub/2016/deconv-checkerboard/, свободный. (дата обращения: 20.02.22)

23. Pinkney, J. StyleGAN Network Blending [Электронный ресурс] / justinpinkney. Режим доступа: https://www.justinpinkney.com/stylegan-network-blending/, свободный. (дата обращения: 27.02.22)

24. Real-ESRGAN [Электронный ресурс] / GitHub. Режим доступа: https://github.com/sberbank-ai/Real-ESRGAN, свободный. (дата обращения: 05.03.22)

25. Ronneberger, O. U-Net: Convolutional Networks for Biomedical Image Segmentation / O. Ronneberger, P. Fischer, and T. Brox // [Электронный ресурс]: arXiv, 2015- Режим доступа: https://arxiv.org/abs/1505.04597, свободный (дата обращения: 27.01.22)

26. Shi, W. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-

Pixel Convolutional Neural Network / W. Shi, J. Caballero, F. Huszar, J. Totz, A. Aitken, R. Bishop, D. Rueckert, and Z. Wang // [Электронный ресурс]: arXiv, 2016 - Режим доступа: https://arxiv.org/abs/1609.05158, свободный. (дата обращения: 05.03.22)

27. Simonyan, K. Very Deep Convolutional Networks for Large-Scale Image Recognition / K. Simonyan, A. Zisserman // [Электронный ресурс]: arXiv, 2014 - Режим доступа: https://arxiv.org/abs/1409.1556, свободный. (дата обращения: 19.01.22)

28. Super Resolution [Электронный ресурс] / paperswithcode. Режим доступа: https://paperswithcode.com/task/super-resolution, свободный. (дата обращения: 16.11.21)

29. Ulyanov, D. Instance Normalization: The Missing Ingredient for Fast Stylization / D. Ulyanov, A. Vedaldi, and V. Lempitsky // [Электронный ресурс]: arXiv - Режим доступа: https://arxiv.org/abs/1607.08022, свободный (дата обращения: 20.01.22)

30. Zhou, B. Learning Deep Features for Discriminative Localization / B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. // [Электронный ресурс]: arXiv, 2016 - Режим доступа: https://arxiv.org/abs/1512.04150, свободный (дата обращения: 10.03.22)

31. Zhu, J. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks / J. Zhu, T. Park, P. Isola, and A. Efros // [Электронный ресурс]: arXiv, 2017 - Режим доступа: https://arxiv.org/abs/1703.10593, свободный (дата обращения: 07.02.22)

32. Компьютерное зрение [Электронный ресурс] / Университет ИТМО. Режим доступа: http://neerc.ifmo.ru/wiki/index.php?title=%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%BE%D0%B5_%D0%B7%D1%80%D0%B5%D0%BD%D0%B8%D0%B5, свободный. (дата обращения: 11.11.21)

33. Машинное обучение [Электронный ресурс] / Википедия. Режим доступа: https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5, свободный. (дата обращения: 11.11.21)

34. Механизм внимания [Электронный ресурс] / Университет ИТМО. Режим доступа: http://neerc.ifmo.ru/wiki/index.php?title=%D0%9C%D0%B5%D1%85%D0%B0%D0%BD%D0%B8%D0%B7%D0%BC_%D0%B2%D0%BD%D0%B8%D0%BC%D0%B0%D0%BD%D0%B8%D1%8F, свободный. (дата обращения: 20.01.22)

35. Сверточные нейронные сети [Электронный ресурс] / Университет ИТМО. Режим доступа: http://neerc.ifmo.ru/wiki/index.php?title=%D0%A1%D0%B2%D0%B5%D1%80%D1%82%D0%BE%D1%87%D0%BD%D1%8B%D0%B5_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5_%D1%81%D0%B5%D1%82%D0%B8, свободный. (дата обращения: 11.11.21)