|  |  |
| --- | --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ | |
|  | |
| Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Пермский государственный национальный исследовательский университет» | |
| Кафедра математического обеспечения вычислительных систем | |
| УДК 004.032.26 | |
| **ГЕНЕРАЦИЯ СТИЛИЗОВАННЫХ ОБРАЗОВ ЛЮДЕЙ С ПОМОЩЬЮ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ**  *Курсовая работа* | |
|  | Работу выполнил студент  ПМИ-19-2 группы 3 курса механико-математического факультета  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.М. Писцов |
|  | Научный руководитель:  старший преподаватель кафедры математического обеспечения вычислительных систем  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.А. Юрков  "\_\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г. |
| Пермь 2022 | |

**Оглавление**

[Введение 3](#_Toc102015365)

[1 Обзор существующих способов генерации изображений 5](#_Toc102015366)

[1.1 Генеративно-состязательная нейронная сеть 5](#_Toc102015367)

[1.2 Комбинированные способы генерации стилизованных изображений 7](#_Toc102015368)

[2 Принципы генерации изображения и способы стилизации 9](#_Toc102015369)

[2.1 Стиль изображения 9](#_Toc102015370)

[2.2 Генерация изображения 10](#_Toc102015371)

[2.3 Функция ошибки 13](#_Toc102015372)

[2.4 Функция ошибки стиля 16](#_Toc102015373)

[3 Создание алгоритма генерации качественных изображения 19](#_Toc102015374)

[3.1 Создание текстовой подсказки 19](#_Toc102015375)

[3.2 Оптимизация генерации стилизованных портретов 20](#_Toc102015376)

[4 Тестирование различных параметров генерации стилизованных портретов людей 23](#_Toc102015377)

[4.1 Сравнение текстовых фраз 23](#_Toc102015378)

[4.2 Сравнение обучающих моделей 25](#_Toc102015379)

[4.3 Сравнение коэффициентов функции ошибки контента 25](#_Toc102015380)

[4.4 Алгоритм генерации изображений 26](#_Toc102015381)

[Заключение 28](#_Toc102015382)

[Библиографический список 30](#_Toc102015383)

[Приложение А 33](#_Toc102015384)

[Приложение Б 34](#_Toc102015385)

Введение

Стилизация изображений в последнее время приобретает все больше применений. Стилизация — воспроизведение или «имитация образной системы и формальных особенностей одного из стилей прошлого, использованных в новом художественном контексте» [[19](#Библсписок19)]. Целью стилизации считается достижение выразительности с частичным или полным отказом от достоверности изображения. Психологическая задача стилизации заключается в избавлении от обилия деталей, которые мешают целостному восприятию общей картины.

Стилизация портретов находят применение во многих сферах жизни: аватары в социальных сетях, концепт-арты, персонажи мультсериалов, персонажи на обложках книг или журналов. Упрощение деталей позволяет сконцентрироваться на сути изображения, лучше запоминать важную информацию. Основным автоматизированным способом создания стилизованных портретов является наложение фильтров в графических редакторах. Фильтры работают с пикселями изображения, что позволяет осветлить или затемнить отдельные участки, выделить контуры, изменить тон, четкость.

Недостатком такого подхода является невозможность нанесения стилей, обладающих семантической нагрузкой. Художник может своими руками создать портрет, например, в стиле гранж, фьюжн или сюрреализм с сохранением основных черт, свойственных изначальному фотопортрету. Однако этот процесс требует слишком много ресурсов.

Одним из способов автоматизировать процесс наложения семантически нагруженных стилей (далее - стиль) является применение нейронных сетей. Новые комбинации нейронных сетей, основанных на GAN (генеративно-состязательная сеть) [[1](#Библсписок1)], способны создавать фотореалистичные портреты (StyleGAN от NVDIA [[2](#Библсписок2)]), переносить стили рисования персонажей на портреты людей (BlendGAN [[13](#Библсписок13)]), наносить стиль на изображение с использованием текстовой фразы (сочетание VQGAN+CLIP [[18](#Библсписок18)]). Обширность возможностей этих архитектур обусловлена способностью нейронной сети выделять объекты, обладающие семантически большой значимостью, разделять объекты на менее значимые элементы (кривые, контуры), находить связи между объектами на всем изображении.

Однако даже при использовании нейронных сетей сложно реализовать универсальный алгоритм, который генерирует стилизованные портреты с желаемым результатом. Ограничением выступает объем обучающей выборки для создания широкого спектра стилизованных изображений, возможность выделять контуры лица для генерации качественных портретов, гибкость уровня стилизации и генерации.

Целью данной работы является разработка алгоритма, реализующего возможность генерации большого спектра стилизованных портретов человека. Для достижения данной цели необходимо поставить задачи:

1. Определить критерии, которым должен соответствовать алгоритм на основе комбинации нейронных сетей для генерации стилизованных портретов.
2. Проанализировать существующие способы генерации портретов людей и стилизации, выявить их слабые и сильные стороны.
3. Определить способы модернизации подходящей комбинации нейронных сетей для достижения цели.
4. Реализовать алгоритм генерации качественных портретов людей с широким спектром возможностей.
5. Протестировать алгоритм при различных параметрах и входных данных.
6. Обзор существующих способов генерации изображений

За последний год появилось множества методов для оптимизации генерации изображений нейронными сетями. Чтобы понимать принципы стилизации и создания портретов, рассмотрим один из самых популярных и эффективных методов настройки нейронной сети для создания качественных изображений.

* 1. Генеративно-состязательная нейронная сеть

GAN (Generative Adversarial Network) представляет собой алгоритм машинного обучения без учителя, построенный на комбинации из двух нейронных сетей, одна из которых (сеть G - генератор) генерирует образцы, а другая (сеть D - дискриминатор) старается отличить правильные («подлинные») образцы от неправильных. Данный алгоритм основан на архитектуре сверточных нейронных сетей [[10](#Библсписок10)]. На [рисунке 1](#Рисунок1) представлена концептуальная схема работы GAN.

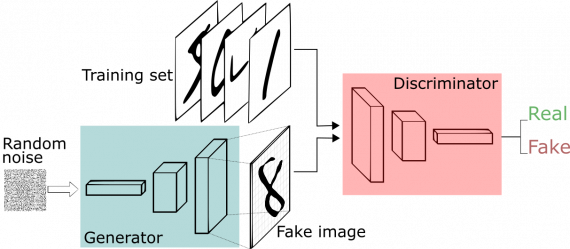


Рис. 1 - Концептуальная схема работы GAN.

Дискриминатор посчитает сгенерированный образ с высокой точностью подлинным в том случае, если он «похож» на датасет. Обе сети преследуют противоположные цели, в результате чего создается антагонистическая игра. По своей структуре генератор и дискриминатор являются частями автоэнкодера [[3](#Библсписок3)], где первый является декодером, второй - энкодером. Дискриминатор получает на вход изображение, а генератор – вектор, или тензор латентного пространства.

Тензор - объект линейной алгебры, линейно преобразующий элементы одного линейного пространства в элементы другого. Латентное пространство - абстрактное многомерное пространство, которое кодирует осмысленное внутреннее представление наблюдаемых извне событий [[4](#Библсписок4)]. В узком смысле латентное пространство можно рассматривать как репрезентацию входных данных. В контексте изображений латентное пространство можно рассматривать как набор выделенных на изображении образов объектов (нос, глаз, фоновые предметы). Сжатое представление изображения получается путем поэтапного выделения образов объектов с меньшей семантической нагрузкой, то есть классификацией объектов исходного изображения, которое состоит из пикселей.

В первую очередь дискриминатор обучается на начальном датасете для распознавания особенностей выборки изображения. Тренировка заключается в минимизации функции ошибки, которая сравнивает ожидаемый и полученный результат репрезентативного представления изображения (латентного пространства). Ошибка позволяет настроить веса сверточных слоев для улучшения результата методом обратного распространения ошибки. Обученный дискриминатор получает на вход сгенерированное изображение, генерирует его сжатое представление и сравнивает со сжатым представлением изображений из выборки. Полученная ошибка передается генератору для обновления весов.

Далее рассмотрим модели, способные генерировать стилизованные изображения.

* 1. Комбинированные способы генерации стилизованных изображений

Под моделью будем понимать конкретную связку нейронных сетей, либо нейронной сети в комбинации с определенным алгоритмом.

StyleGAN. Данную модель представила группа исследователей NVDIA в 2018 году как способ генерации качественных фотореалистичных портретов людей. Как основа, она была использована в алгоритме переноса стиля JoJoGAN [[14](#Библсписок14)]. Идея переноса стиля была заимствована из модели Adain (Adaptive Instance Normalization) [[5](#Библсписок5)]. На вход подаются 2 изображения: первое необходимо стилизовать, а второе содержит стиль для переноса. Изначальной проблемой переноса стиля было постоянное переобучение для каждого нового стиля. Принцип работы заключаются в выделении латентных блоков свертки, которые содержат стиль, и тех, которые содержат контент изображения. В данном случае функцией ошибки будет разница латентных пространств на слоях, отвечающих за стиль. Разработчики объединили данный подход со StyleGAN, что позволило создавать более качественные стилизованные изображения. Гибкость достигается за счет изменения коэффициента стилизации – меру соответствия изображения со стилем, в следствие итоговое изображение будет больше или меньше «похожим» на оригинальное. Однако для каждого стиля необходимо иметь второе изображение, которое содержит применяемый стиль. Такой недостаток не позволит генерировать широкий спектр изображений с разными стилями.

DALL-E и CLIP. В начале 2021 года компания OpenAI выпустили 2 нейронных сети – DALL-E и CLIP [[6](#Библсписок6),[7](#Библсписок7)]. Они выполняют противоположные задачи: первая предсказывает текстовое описание по изображению, вторая генерирует изображения по текстовому описанию. DALL-E представляет собой мультимодальную реализацию GPT (Generative Pre-Trained Transformer) [[8](#Библсписок8)]. Под мультимодальной реализацией предполагается объединение нескольких моделей в одну. В отличие от моделей, основанных на GAN, данная создает более абстрактные изображения. В GAN существует более жесткая привязка к обучающей выборке, этому способствует принцип работы дискриминатора. DALL-E генерирует сразу несколько изображений, а CLIP их ранжирует в зависимости от их соответствия текстовому запросу. Такая модель способна сгенерировать портрет человека с определенным стилем, все зависит от правильности ввода текстовой фразы. Из недостатков можно выделить невозможность использования входного изображения, излишняя абстрактность при создании портрета, большие временные затраты на генерацию изображения.

VQGAN (Vector Quantized Generative Adversarial Network) – комбинированная нейронная сеть, отличительной чертой которой является использования дискретного скрытого представления изображения, в то время, как обычная сеть GAN содержит латентное пространство непрерывных значений. При таком подходе сравнивать латентные пространства изображений проще – достаточно вычислить между ними расстояние. Такая сеть, как и DALL-E, использует Трансформер [[9](#Библсписок9)] для поиска связей между фрагментами изображения. Сочетание GAN, трансформера и дискретного латентного пространства позволяет добиться более качественной генерации изображений. Существенным недостатком является отсутствие метода наложения стилей, низкое качество генерации лиц людей.

Исходя из существующих способов стилизации, VQGAN, несмотря на свои недостатки, является хорошей основой для добавления новых алгоритмов. Например, комбинация VQGAN+CLIP позволяет генерировать стилизованное изображение на основе входного при помощи выделения стиля из заданной текстовой фразы (подсказки). Прежде всего необходимо определить, каким образом эти две сети могут взаимодействовать. Для этого рассмотрим по отдельности каждую нейронную сеть.

1. Принципы генерации изображения и способы стилизации
   1. Стиль изображения

Стиль - визуальное отражение определенного образа с использованием специальных приемов. Чтобы понимать, как можно наложить стиль на изображение, необходимо разграничить понятия стиля изображения и контента изображения в контексте сверточных сетей. Для этого рассмотрим процесс выделения образов объектов на изображения [[10](#Библсписок10)].

Двумерная свертка изображения представляет собой последовательное умножение ядра (матрицы весов) заданного размера на участки изображения с последующим суммированием. Результатом является двумерная матрица признаков меньшего размера. Ядро – двумерная матрица. Значения ядра задают распознаваемый признак, набор ядер на каждом этапе свертки задают фильтр. Фильтр скользит по областям карты представления изображения для выделения признаков. Размер ядра определяет количество признаков, которые нейронная сеть может распознать. Пулинг – сокращение пространственного представления изображения для уменьшения количества параметров и объема вычислений путем выбора наибольшего числового значения выбранного фрагмента изображения, соответствующего размеру ядра пулинга.

Последовательное применение свертки и пулинга позволяет выделить более сложные признаки с большей семантической значимостью. В таком контексте определим контент изображения, как совокупность наиболее семантически сложных признаков. Тогда стиль в контексте нейронных сетей – совокупность признаков, семантическая значимость которых меньше, чем у контента. Имея различные наборы признаков изображения (результаты этапов свертки), мы можем вычленять различные представления изображения, которые притом не будут зависеть от конкретного расположения объектов на изначальном изображении.

Понимая, как происходит выделение образов изображения, необходимо рассмотреть принцип генерации изображения из образов.

* 1. Генерация изображения

Генерация заключается в последовательном применении операции обратной свертки, или деконволюции. Рассмотрим 2 нейронные сети: сверточная [[10](#Библсписок10)] и зеркально отображенная сверточная, которые представлены на [рисунке 2](#Рисунок2):

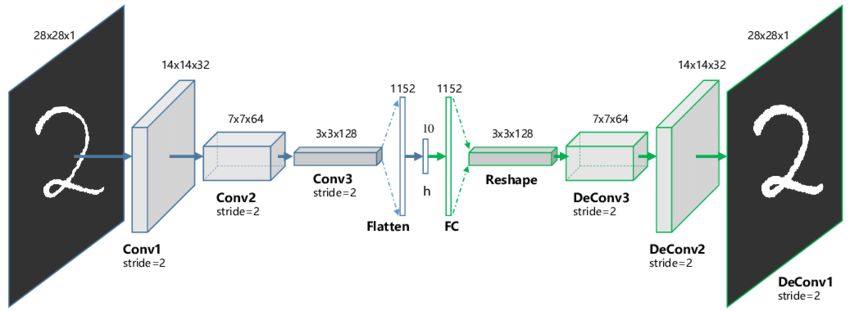


Рис. 2 - Концептуальная схема работы автоэнкодера.

Будем считать наиболее сжатое латентное пространство изображения в рамках конкретной архитектуры сверточной нейронный сети z – пространством. На рисунке 2 блок h является z – пространством. Если развернуть левую часть сети (до блока h) зеркально и полученной сети подать на вход блок h, на выходе получим некотором изображение. Данная нейронная сеть называется вариационный автоэнкодер (VAE) [[3](#Библсписок3)], в которой есть энкодер, генерирующий из изображения z-пространство, и декодер, генерирующий из z-пространства изображение. Задача оптимизации автоэнкодера заключается в настройке сети таким образом, чтобы сгенерированное изображение полностью соответствовало начальному. Генератор GAN работает по схожей схеме, что и декодер, однако на вход получает гауссовский шум. Будем рассматривать обученную нейронную сеть в том смысле, что веса тензоров промежуточных скрытых блоков уже настроены для генерации изображений. Далее под обучением будем понимать настройку весов z-пространства таким образом, чтобы на выходе генератора получилось желаемое изображение.

Чем отличается генератор VQGAN? Как говорилось ранее, данная архитектура имеет дискретный тензор скрытого пространства, а именно – дискретное z – пространство. Оно представляет собой кодовую книгу (codebook). К поступающим в процессе обучения тензорам применяется метод k-средних. Из множества групп выделяется центроида и заносится в кодовую книгу. Каждая группа характеризует объект (шляпа, город, луна и т.д.). Далее в процессе свертки тензоры латентного пространства сравниваются с эталонными векторами (находится минимум путем вычисления модуля расстояния). На концептуальном уровне эталонные векторы наиболее точно описывают объекты на изображении. Процесс создания кодовой книги называется векторное квантирование [[11](#Библсписок11)]. Автоэнкодер, в котором предусмотрено взаимодействие с кодовой книгой, называется вариационный автоэнкодер с векторным квантированием (VQVAE).

Имея описание объектов изображения в виде набора дискретных образов, мы можем найти долгосрочные связи между фрагментами изображения. В этом заключается механизм скользящего внимания (Attention) [[12](#Библсписок12)], используемый в модели Трансформер. На [рисунке 3](#Рисунок3) представлена концептуальная схема работы VQGAN.

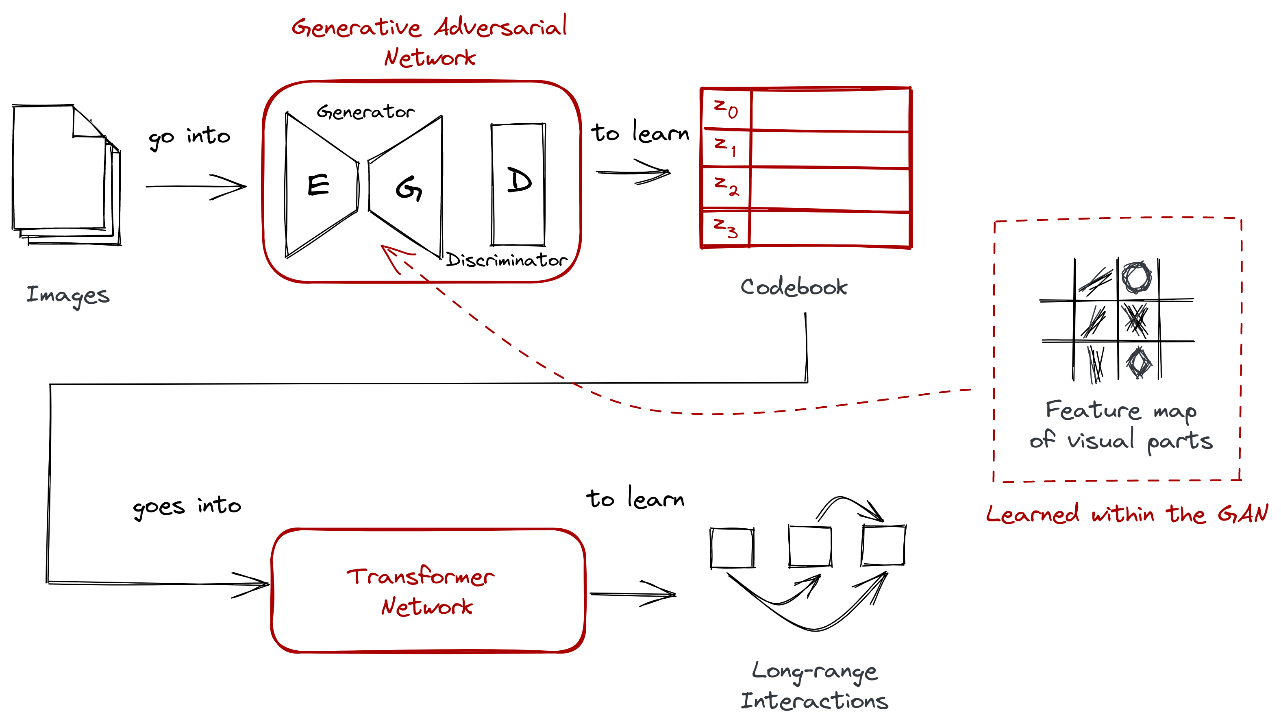


Рис. 3 - Концептуальная схема работы VQGAN.

VQGAN объединяет в себе возможность генерации точных изображений (GAN), возможность генерации и настройки кодовой книги (VQVAE) для работы Трансформера, который ищет отношения между визуальными частями изображения.

Теперь выдвинем требования, которым должен соответствовать генератор для решения поставленной задачи:

1. Целевое изображение «похоже» на изначальное. Сохранены основные контуры, благодаря чему целевое изображение будет восприниматься именно как стилизованное, а не совершенно новое.
2. Алгоритм предоставляет возможность наложения широкого спектра стилей.
3. Целевое изображения сохраняет черты лица человека.

Для каждого конкретного изображения необходимо обучить кодовую книгу путем сокращения значения функции ошибки.

* 1. Функция ошибки

Обучение кодовой книги сводится к минимизации функции ошибки от ожидаемого и полученного результатов. Прежде всего необходим понять, как происходит настройка весов нейронной сети [[15](#Библсписок15)]. Рассмотрим принцип настройки весов на примере реконструкции изображения.

Пусть – вход сети (начальное изображение) GAN, – сгенерированный образец. Введем функцию минимаксной потери , которая характеризует способность дискриминатора D различать реальные и реконструированные изображения:

*,*

где показывает вероятность, что действительно реальное изображение,

показывает вероятность, что сгенерированный образец подлинный.

VQGAN следует архитектуре энкодера-декодера (где E – энкодер, D - декодер), что напоминает принцип работы VQVAE. На [рисунке 4](#Рисунок4) представлена концептуальная схема работы генератора VQGAN.

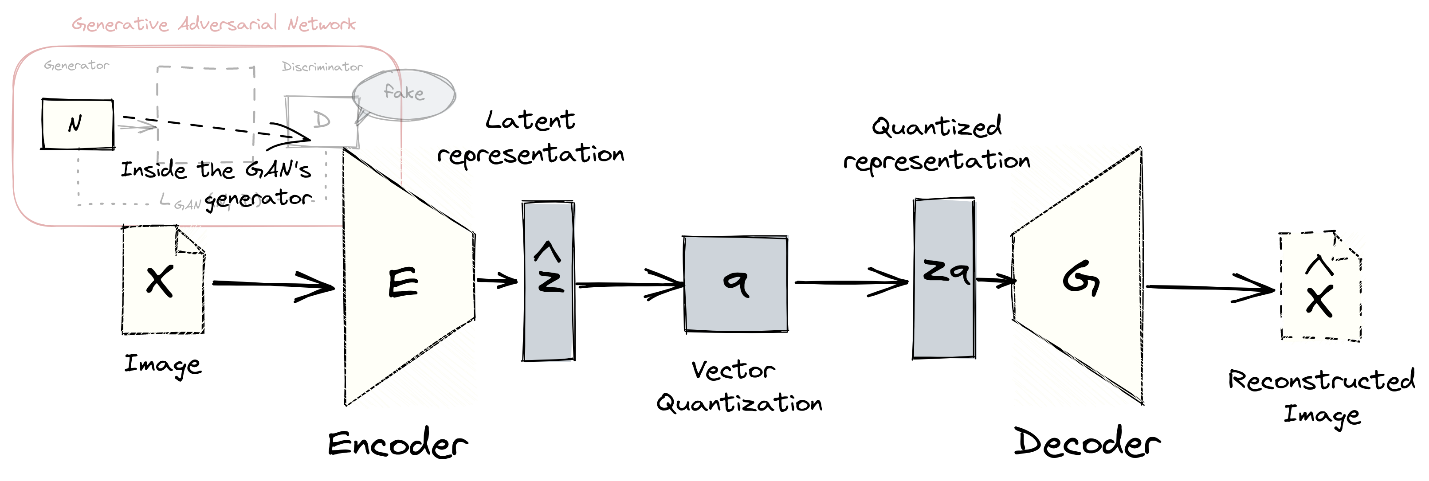


Рис. 4 - Концептуальная схема работы генератора VQGAN

В результате кодирования в (), выполняется поэлементное квантирование q для получения дискретного представления ):

Таким образом, для восстановления изображения используется дискретный вектор вместо непрерывного . Тогда процесс генерации изображения примет следующий вид:

Модель энкодера-декодера и кодовая книга обучаются совместно при помощи комбинированной функции потерь:

,

Где задает потерю реконструкции.

Она показывает, насколько выход соответствует входу при условии использования лишь квантированного представления Функция потери необходима для оптимизации z – пространства. Операция sg отвечает за отключения вычисления градиентов для предотвращения переобучения. Часть называется «функция потери обязательств», которая гарантирует, что энкодер E зафиксирует конкретное представление изображения.

Тогда для получения оптимальной сжатой модели для обучения сети VQGAN необходимо объединить функции потерь и :

Где – адаптивный вес вектора,

предсказание индексов z-пространства Трансформером.

Данный способ оптимизации модели обучения позволяет добиться соответствия контента и стиля для начального и сгенерированного изображений.

Заметим, что итоговая функция ошибки является комбинацией из двух других, что дает больший простор для модернизации процесса генерации. В данном контексте введем еще одно определение понятия модели. Модель –конфигурация обученной нейронной сети. В узком смысле модель представляет собой набор настроенных весов сверточных блоков нейронной сети в результате обучения на конкретном датасете. Напомним, что рассматриваемая комбинация нейронных сетей использует предобученную модель на большом датасете изображений. Для генерации изображения необходимо обучить кодовую книгу, которая изначально содержит гауссовский шум. Введем функцию ошибки целевого изображения , которая показывает, насколько сгенерированное изображение отличается от начального:

,

Где – квантированное z-пространство начального изображения, - квантированное z-пространство сгенерированного изображения. Тогда рассчитывается как квадратичная сферическая ошибка между репрезентациями изображений:

Теперь необходимо добавить возможность генерации стилизованных изображений. VQGAN И CLIP можно комбинировать, благодаря чему появляется возможность использовать начальное изображение и текстовую подсказку, в которой заключен желаемый стиль целевого изображения. Их модель взаимодействия похожа на принцип работы GAN: VQGAN генерирует изображение, CLIP дискриминирует, то есть определяет, насколько полученное изображение соответствует введённой фразе и передает значение функции ошибки от стиля фразы и стиля изображения генератору. Для введения функции ошибки стиля, рассмотрим принцип работы CLIP.

* 1. Функция ошибки стиля

CLIP (Contrastive Language-Image Pre-Training) является универсальным классификатором, позволяющим интерпретировать представление изображения на естественном языке.

Классический способ обучения нейронных сетей строится на разделении выборки на обучающую и тестовую, обе содержат одинаковые классы изображений. При таком подходе необходимо переобучать модель для каждого конкретного случая, что ограничивает спектр генерации стилизованных изображений и требует больших временных затрат. CLIP использует подход zero-shot learning [16], или «обучение без обучения». Он заключается в предварительном обучении модели на большой выборке данных (400 миллионов пар изображение-текст) [[7](#Библсписок7)]. Такая модель способна ранжировать широкий спектр визуальных и текстовых представлений без повторного обучения. На [рисунке 5](#Рисунок5) представлена концептуальная схема поиска связи текстового и визуального представления объектов.

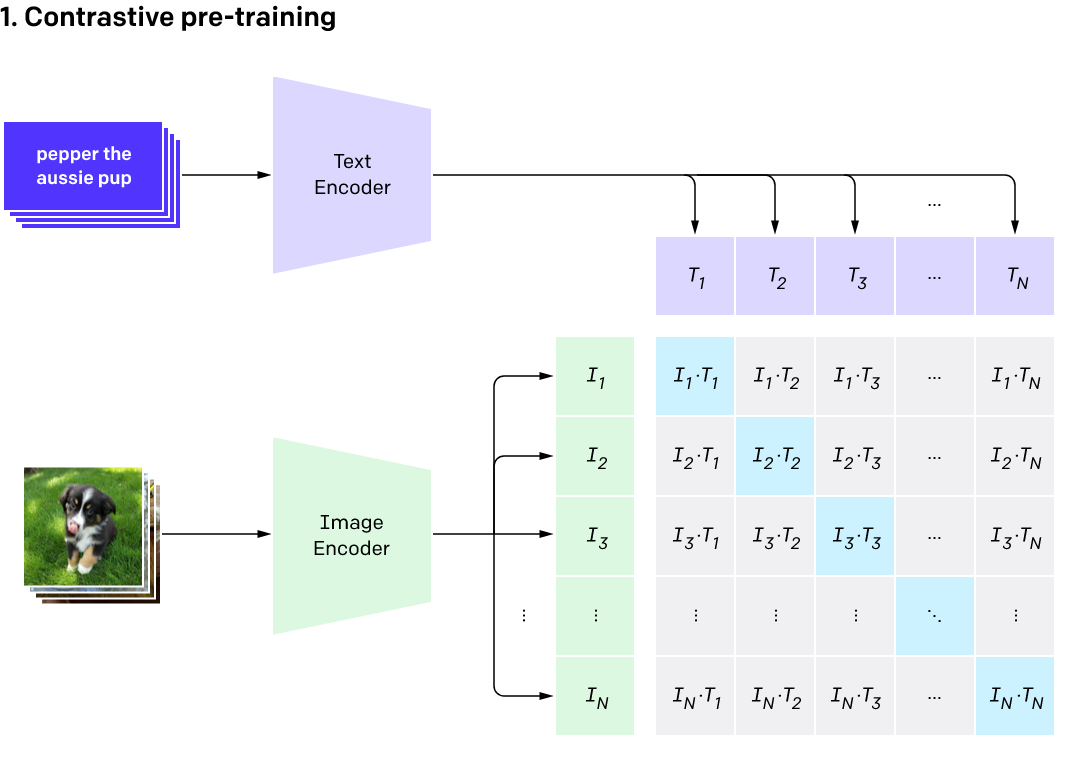


Рис. 5 - Концептуальная схема поиска связи текстового и визуального представления объектов.

В процессе свертки CLIP использует Трансфомер для поиска долгосрочных связей между представлениями отдельных слов, что дает возможность обрабатывать самые разные способы построения фразы.

Для изображения и текста строятся векторные репрезентации и . Для сравнения представлений изображения и текста необходимо создать единое векторное пространство путем линейного преобразования их латентных векторов.

После линейного преобразования векторов к одинаковой размерности строится матрица подобия , элементами которой являются значения метрики косинусного сходства между представлениями i-го фрагмента изображения и j-го текстового описания:

Отдельно вычисляется перекрестная энтропия (функция ошибки) для текста и изображения через функции кросс-энтропии и :

Общая функция ошибки CLIP выглядит как комбинация энтропий по изображению и тексту:

Теперь при обучении кодовой книги VQGAN необходимо учесть ошибку стиля , общая функции ошибки примет вид:

Далее рассмотрим различные способы оптимизации рассматриваемой комбинации VQGAN+CLIP

1. Создание алгоритма генерации качественных изображения
   1. Создание текстовой подсказки

Качество генерации изображения напрямую зависит от правильности задания текстовой фразы. Переобучение модели для извлечения стилей изображений эффективно для решения узкой задачи. Она будет содержать представления наиболее важных объектов. Главным минусом такого подхода являются большие временные затраты. Для эффективной работы с моделью, обученной на широком спектре изображений (Zero-shot learning) необходимо подбирать наиболее корректные текстовые подсказки для получения желаемого результата.

Под концепцией «Prompt Engineering» понимается подбор модификаций текста для решения задачи, путем задания параметров на естественном языке, что в комбинации с моделью, обученной на широком спектре изображений, позволяет решать большое количество задач. Для составления текстовой подсказки прежде всего нужно определиться с основой подсказки, в нашем случае лицо человека. Это могут быть вариации синонимов: «portrait», «human face», «person», «man» и т.д. Для уточнения, изменения гендера человека, омоложения или старения можно использовать соответствующие подсказки: «woman», «men», «young», «elderly». Для стилизации изображения под рисунки можно использовать такие конструкции, как «the art of… (by …)», «a colorful painting of …» и т.д. При таком подходе можно создавать образы со стилистикой отдельных художников («Hans Holbein», «Ilya Repin»), связанные с направлением живописи («modernism», «cubism») или стилистикой, свойственной сайтам-портфолио современный художников («deviantart», «artstation»).

На [рисунке 6](#Рисунок6) приведено сравнение генерации изображений при различных уточняющих подсказках:



Рис. 6 - Сравнение результатов генерации портретов при различных текстовых подсказках и различных ключевых словах (Art, Sketch).

Ключевые слова позволяют более точно передать желаемый стиль изображения. Однако качество генерации изображений оставляет желать лучшего. Для решения этой проблемы необходим способ выделения контуров лица. В нашем случае это функция ошибки, которая показывает, насколько лицо человека на сгенерированном изображении соответствует лицу на изначальном изображении.

* 1. Оптимизация генерации стилизованных портретов

Принцип наложения стилей JoJoGAN заключался в извлечении признаков стилизованного и изначального изображений. Будем использовать подобный алгоритм для извлечения контента изображения. Добавим новую функцию ошибки, которая покажет, насколько контент сгенерированного изображения расходится с контентом стилизованного. Контентом начального изображения будем считать лицо человека, как образ, обладающий наибольшей семантической сложностью.

Для решения поставленной задачи необходимо использовать нейронную сеть, из которой можно извлекать репрезентации изображения на разных этапах свертки. Примером такой сети является VGG16 [[17](#Библсписок17)], ее концептуальная схема архитектуры приведена на [рисунке 7](#Рисунок7).

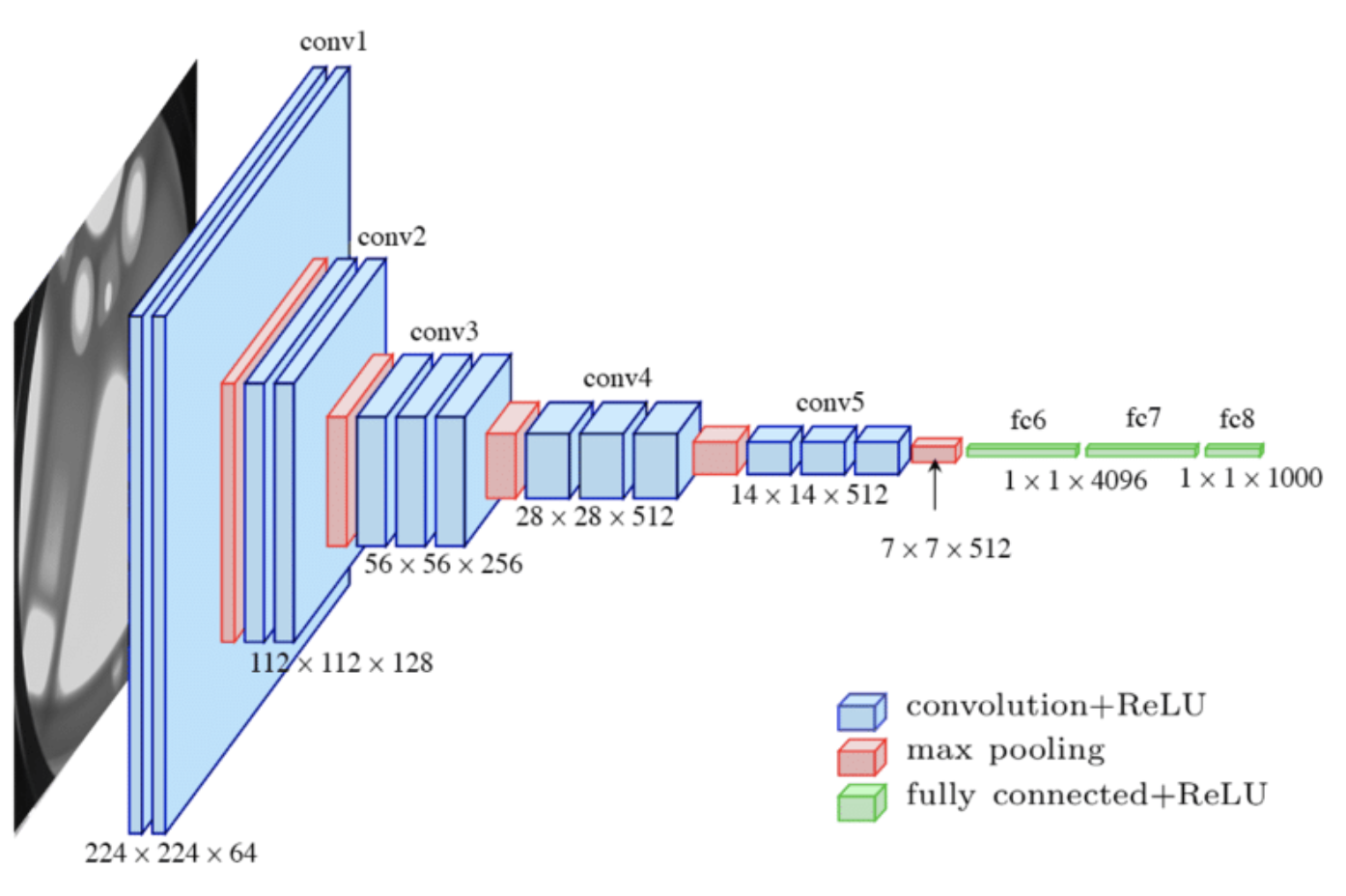


Рис. 7 - Концептуальная схема архитектуры нейронной сети VGG16.

Данная сеть состоит из 5 сверточных блоков, в которой последний 5 представляет собой репрезентацию контента изображения. Чтобы сравнивать контуры лица человека необходимо использовать предобученную модель на лицах людей.

На вход сеть получает изначальное изображение и сгенерированное . Для вычисления функции ошибки по контенту лица необходимо получить репрезентации изображения на 5 сверточном блоке. Для вычисления ошибки используется метрика L1Loss, тогда функция ошибки контента этих изображений примет вид:

С учетом функции ошибки стиля и контента, окончательная функция ошибки для обучения кодовой книги примет вид:

Для построения алгоритма генерации сравним результаты генерации изображений при различных параметрах.

1. Тестирование различных параметров генерации стилизованных портретов людей

Для генерации изображений использовалась предобученная модель для архитектуры VQGAN «VQGAN-f16-16384». При обучении использовался датасет ImageNet, содержащий более 1000 классов изображений и 1 281 167 тренировочных изображений. Также в тестировании использовалась модель, обученная на датасете, состоящем из лиц людей, FacesHQ.

Реализация программной части алгоритма написана на языке Python. Необходимым ресурсом для тестирования алгоритма является видеокарта с CUDA – ядрами. Платформа Google Collab предоставляет видеокарты NVIDIA Tesla K80 и NVIDIA Tesla T4 с видеопамятью в 11ГБ, чего достаточно для загрузки 2 моделей: VGGFace для модели VGG, VQGAN-f16-16384 или FacesHQ для VQGAN. Однако объема видеопамяти недостаточно для загрузки больших моделей, обученных на портретах людей (CelebaHQ, FFHQ).

Для работы с тензорами использовалась библиотека PyTorch, модель VQGAN содержит библиотека taming.models. Алгоритм позволяет настраивать размеры выходного изображения, интервал вывода сгенерированных изображений в процессе итеративного обучения кодовой книги, сид для задания гауссовского шума, набор текстовых фраз, содержащих стиль целевого изображения. Сид – число, входной параметр для генерации гауссовского шума.

* 1. Сравнение текстовых фраз

Для соблюдения равных условий, будем использовать одно входное изображений, которое представлено на [рисунке 8](#Рисунок8), для сравнения результатов генерации при различных параметрах.



Рис. 8 - Начальное изображение для сравнения результатов генерации стилизованных портретов при различных параметрах.

Далее на [рисунке 9](#Рисунок9) приведена сетка результатов генерации изображений при коэффициенте ошибки контента, равном 6.0. Ключевыми словами являются «human face» в 1 строке и «portrait» во 2 строке.

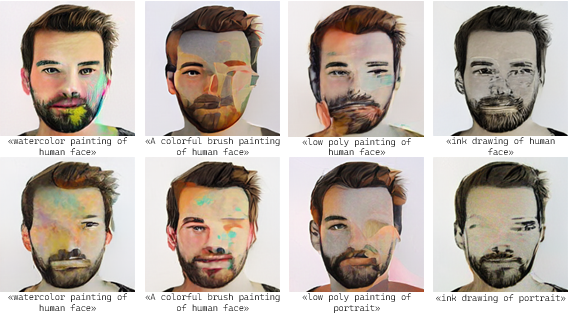


Рис. 9 - Результат сравнения генерации изображений при различных текстовых подсказках с ключевыми словами «human face» и «portrait»

Исходя из результатов, можно сказать, что различные вариации ключевых слов для выделения лица вносят свой уникальный вклад.

* 1. Сравнение обучающих моделей

На [рисунке 10](#Рисунок10) приведена сетка результатов генерации стилизованных изображений при коэффициенте ошибки контента, равном 6.0, на разных датасетах: VQGAN-f16-16384 и FacesHQ:

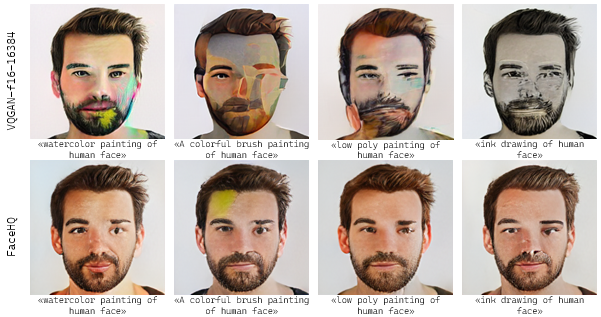


Рис. 10 - Результат сравнения генерации изображений при различных текстовых подсказках на разных обученных моделях.

Исходя из результатов, можно сказать, что модель, обученная лишь на изображениях лиц людей, генерирует менее стилизованные изображения. Черты лица сохранены, однако стиль сгенерированных изображений не соответствует стилю текстовой подсказки.

* 1. Сравнение коэффициентов функции ошибки контента

На [рисунке 11](#Рисунок11) приведена сетка результатов генерации стилизованных изображений с текстовой подсказкой «low poly painting of human face».



Рис. 11 - Результат сравнения генерации изображений при различных значениях параметра веса контента изображения с текстовой подсказкой «low poly painting of human face»

При значениях параметра в границах от 1.0 до 3.0 наблюдается явное искажение черт лица человека. При значениях в границах от 9.0 до 12.0 стиль, заданный текстовой подсказкой, практически незаметен. Наилучший результат генерации достигается при использовании значений, близких к 6.0. Отсюда делаем вывод, что оптимальное значение параметра веса контента лежит в границах от 5.0 до 7.0.

* 1. Алгоритм генерации изображений

Исходя из проведенных выше экспериментов по генерации изображений при различных параметрах, для решения поставленной задачи стилизации изображений лиц людей предлагается следующий алгоритм:

1. Выбрать исходное изображение, содержащее портрет. Предпочтительнее использовать квадратные изображений, в которых портрет занимает почти все пространство, фон светлый.
2. Определить желаемый стилем целевого изображения. Примерами может быть инструмент рисования (цветной карандаш, кисть), стиль, свойственный сайтам-портфолио современный художников (deviant art, pixiv), особенностям черт человеческого лица (молодой, азиат, веснушки).
3. Создать текстовую подсказку (или набор текстовых подсказок) на естественном языке, наиболее точно описывающую желаемый стиль. Конструкция фразы должна содержать слово или словосочетание, описывающее человеческое лицо («human face», «portrait», «a portrait of» и т.д.).
4. Выбрать предобученную модель VQGAN. Она должна быть обучена не только на лицах людей. Предпочтение отдается большому количеству разнообразных классов объектов. Примером такого датасета может выступать VQGAN-f16-16384.
5. Выбрать коэффициент функции ошибки выделения лица человека. Предпочтительнее использовать значение параметра в границах от 5.0 до 7.0.
6. Итеративно обучить кодовую книгу VQGAN на заданных параметрах путем минимизации общей функции ошибки .

Заключение

В ходе работы были выполнены все поставленные задачи. Определены критерии, которым должна соответствовать комбинированная нейронная сеть для генерации стилизованных портретов:

1. Сгенерированное изображение «похоже» на изначальное.
2. Алгоритм предоставляет возможность наложения широкого спектра стилей.
3. Сгенерированное изображения сохраняет черты лица человека.

Были рассмотрены существующие модели генерации и стилизации изображений, определены их сильные и слабые стороны. На текущий момент область автоматизации генерации широкого спектра стилизованных портретов людей при помощи текстовой подсказки рассмотрена слабо. Существующие методы имеют ряд недостатков для решения поставленной задачи.

Была выявлена перспективная комбинация моделей VQGAN+CLIP для решения поставленной задачи. Были проанализированы принципы работы генеративно-состязательной нейронной сети, в частности особенности работы z-пространства конкретно сети VQGAN для поиска методов оптимизации модели.

Были проанализированы принципы работы сети CLIP, способы внедрения функции ошибки стиля текстовой подсказки. Как результат, был найден и реализован метод выделения контента лица человека на изображении для создания комбинированной функции ошибки, что позволило улучшить качество генерации стилизованных портретов путем сравнения латентных представлений изображений в модели VGG.

Был проведен сравнительный анализ создания текстовых подсказок (Prompt Engineering) для достижения желаемого результата генерации изображений при различных структурах фраз и ключевых слов.

Были проведены эксперименты генерации портретов людей при различных текстовых фразах, различных коэффициентах функции ошибки контента изображения, различных обученных моделях. Как итог, был реализован алгоритм для генерации широкого спектра стилизованных изображений лиц людей. Программная часть алгоритма написана на языке Python.

Библиографический список

1. *Goodfellow, Ian J.; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron & Bengio, Yoshua* Generative Adversarial Networks [Электронный ресурс] [Режим доступа: https://arxiv.org/abs/1406.2661] [Проверено: 27.04.2021]
2. *Tero Karras, Samuli Laine, Timo Aila* A Style-Based Generator Architecture for Generative Adversarial Networks [Электронный ресурс] [Режим доступа: https://arxiv.org/abs/1812.04948] [Проверено: 27.04.2021]
3. *Dor Bank, Noam Koenigstein, Raja Giryes* Autoencoders [Электронный ресурс] [Режим доступа: <https://arxiv.org/pdf/2003.05991>] [Проверено: 27.04.2021]
4. towardsdatascience.com [Электронный ресурс]. – 2022. URL:<https://towardsdatascience.com/understanding-latent-space-in-machine-learning-de5a7c687d8d>. Дата обращения: 27.04.22
5. *Xun Huang, Serge Belongie* Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization [Электронный ресурс] [Режим доступа: <https://arxiv.org/abs/1703.06868>] [Проверено: 27.04.2021]
6. *Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, Ilya Sutskever* Zero-Shot Text-to-Image Generation [Электронный ресурс] [Режим доступа: <https://arxiv.org/abs/2102.12092>] [Проверено: 27.04.2021]
7. *Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, Ilya Sutskever* Learning Transferable Visual Models From Natural Language Supervision [Электронный ресурс] [Режим доступа: <https://arxiv.org/abs/2103.00020>] [Проверено: 27.04.2021]
8. *Qihao Zhu, Jianxi Luo* Generative Pre-Trained Transformer for Design Concept Generation: An Exploration [Электронный ресурс] [Режим доступа: <https://arxiv.org/abs/2111.08489>] [Проверено: 27.04.2021]
9. *Patrick Esser, Robin Rombach, Björn Ommer* Taming Transformers for High-Resolution Image Synthesis[Электронный ресурс] [Режим доступа: <https://arxiv.org/abs/2012.09841>] [Проверено: 27.04.2021]
10. *Keiron O'Shea, Ryan Nash* An Introduction to Convolutional Neural Networks [Электронный ресурс] [Режим доступа: <https://arxiv.org/abs/1511.08458>] [Проверено: 27.04.2021]
11. *Yunchao Gong, Liu Liu, Ming Yang, Lubomir Bourdev* Compressing Deep Convolutional Networks using Vector Quantization [Электронный ресурс] [Режим доступа: <https://arxiv.org/abs/1412.6115>] [Проверено: 27.04.2021]
12. *Nora Vogt* CNNs, LSTMs, and Attention Networks for Pathology Detection in Medical Data [Электронный ресурс] [Режим доступа: <https://arxiv.org/abs/1912.00852>] [Проверено: 27.04.2021]
13. *Mingcong Liu, Qiang Li, Zekui Qin, Guoxin Zhang, Pengfei Wan, Wen Zheng* BlendGAN: Implicitly GAN Blending for Arbitrary Stylized Face Generation [Электронный ресурс] [Режим доступа: <https://arxiv.org/abs/2110.11728>] [Проверено: 27.04.2021]
14. *Min Jin Chong, David Forsyth* JoJoGAN: One Shot Face Stylization [Электронный ресурс] [Режим доступа: <https://arxiv.org/abs/2112.11641>] [Проверено: 27.04.2021]
15. Github [Электронный ресурс]. -2022. –URL:<https://ljvmiranda921.github.io/notebook/2021/08/08/clip-vqgan/> Дата обращения: 27.04.2021
16. *Yongqin Xian, Christoph H. Lampert, Bernt Schiele, Zeynep Akata* Zero-Shot Learning -- A Comprehensive Evaluation of the Good, the Bad and the Ugly [Электронный ресурс] [Режим доступа: <https://arxiv.org/abs/1707.00600>] [Проверено: 27.04.2021]
17. *Karen Simonyan, Andrew Zisserman* Very Deep Convolutional Networks for Large-Scale Image Recognition [Электронный ресурс] [Режим доступа: <https://arxiv.org/abs/1409.1556>] [Проверено: 27.04.2021]
18. *Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, Edward Raff* *VQGAN-CLIP* Open Domain Image Generation and Editing with Natural Language Guidance[Электронный ресурс] [Режим доступа:[https://arxiv.org/abs/2204.08583](https://arxiv.org/abs/1409.1556)] [Проверено: 27.04.2021]
19. ru.wikipedia.org [Электронный ресурс]. – 2022. URL: https://ru.wikipedia.org/wiki/Стилизация\_(изобразительное\_искусство). Дата обращения: 28.04.22

Приложение А

Ссылка на исходных код в блокноте Google Collab:

https://colab.research.google.com/drive/15dPpPSTy5\_Nejt2xiVU8B\_zUsxJSVy9m

Приложение Б

Ссылки на предобученные модели VQGAN-f16-16384 и FacesHQ.

VQGAN-f16-16384:

<https://huggingface.co/flax-community/vqgan_f16_16384>

FacesHQ:

https://app.koofr.net/links/a04deec9-0c59-4673-8b37-3d696fe63a5d