Отчёт о проделанной работе и экспериментах со StyleCLIP.

Выполнен студентом весеннего потока 1-го семестра 2024-го года курса Deep Learning School *Феоктистовым Станиславом*

Что реализовано:

- Оптимизационный подход
- Инверсия GAN

Эксперименты с оптимизационным подходом

Напомню, что наша Loss-функция выглядит следующим образом:

$$\arg\min_{w \in \mathcal{W}+} D_{\text{CLIP}}(G(w), t) + \lambda_{\text{L2}} \|w - w_s\|_2 + \lambda_{\text{ID}} \mathcal{L}_{\text{ID}}(w), \quad (1)$$

Где первый компонент – **CLIP Loss**

Второй компонент - **L2 Loss**, не позволяющий уйти далеко от стартового вектора оптимизации, данный лосс берётся с весовым коэффициентом l2_lambda.
Третий компонент – **ID Loss** с коэффициентом id_lambda.

Эксперименты в основном будут направлены на то, как изменение *l2_lambda*, *id_lambda*, *learning rate*, *количества шагов*, а также самих редактируемых частей будет влиять на эдитинг изображений.

Зададим начальные параметры (они более-менее оптимальные, я их уже подбирал), которые постепенно будем менять

prompt="A red hair"
lr=0.1
n_steps=100
l2_lambda=0.005

id_lambda=0.005

Посмотрим на получившийся результат:

На первый взгляд очень даже неплохо, но есть куда стремиться: лицо довольно существенно изменилось (помолодело и посветлело).

Эксперименты с L2

Теперь попробуем увеличить L2 коэффициент. По идее, это должно сохранить наше изображение более похожим на начальное, т.к. именно L2 Loss не даёт нам сильно менять изображение

Заметим, что изображение стало чуть более похожим на начальное (присмотритесь к морщинам и цвету лица), но уже цвет волос становится не таким ярким и насыщенным.

Разумеется, если сделать L2 очень большим (например, 1), то изображение останется идеально похожим на начальное, но тогда изменения с промпта не отобразятся.

Давайте для разнообразия возьмём другое изображение и сделаем несколько экспериментов, где будем сильно менять l2_lambda, и посмотрим, к чему это приведёт.

l2_lambda = 0.01

l2_lambda = 0

l2_lambda = 1

Как видим – результаты весьма ожидаемые. При нулевом значении этого коэффициента изображение очень сильно скатывается в промпт, а при большом – вообще не меняется.

Эксперименты с ID

Теперь попробуем занулить коэффициент id_lambda=0, а l2_lambda оставим = 0.01 (это весьма оптимальное значение для данных изображения и промпта). Также сразу прикрепим пару экспериментов с id_lambda=0.5 и id_lambda=1.

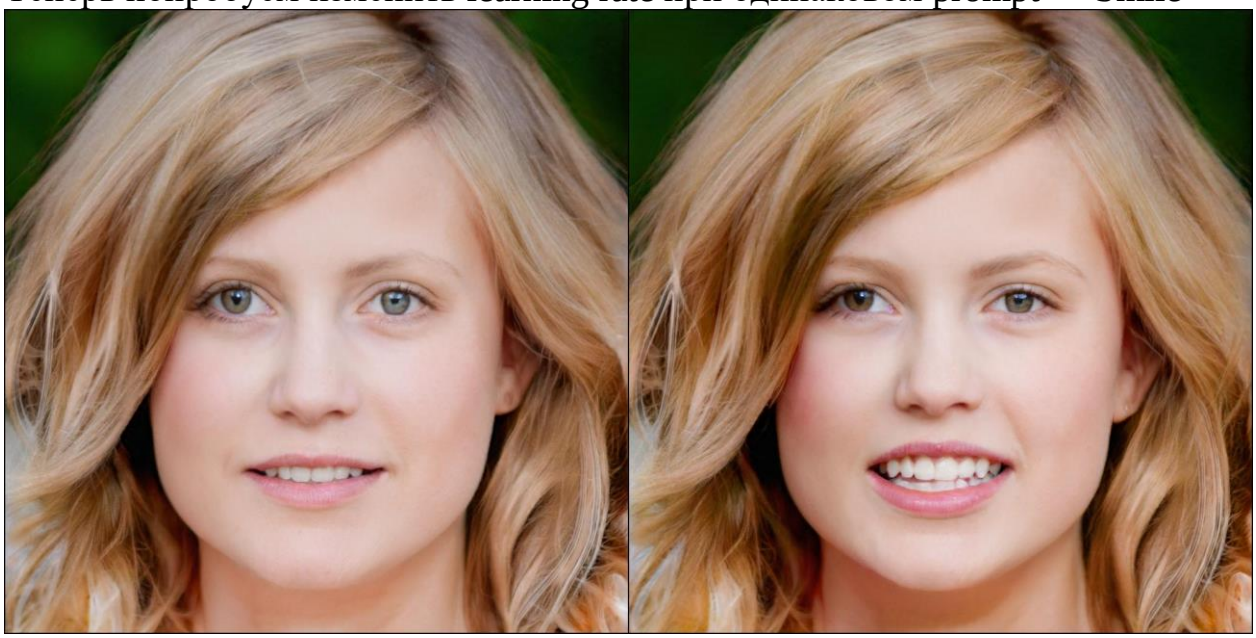
id_lambda=0

id_lambda=0.5

Можно сделать вывод, что id_lambda не особо влияет на изображение (хотя по идее, оно должно делать так, чтобы изображение сильно не скатывалось в другое, возможно у меня какие-то неточности в реализации).

Эксперименты с Learning Rate

Теперь попробуем изменить learning rate при одинаковом prompt = "Smile"



lr = 0.05

lr = 0.1

lr = 0.001

lr = 0.3

Как видим, lr не сильно влияет на редактирования изображения, хотя при маленьком lr = 0.001 можно заметить, что улыбка стала менее явной.

Эксперименты с количеством шагов

Теперь попробуем изменить количество шагов для оптимизации (при этом оставим одинаковый learning rate)

n_steps = 100

n_steps = 300

Как видим, количество шагов не сильно влияют на результат (конечно, если их значения в адекватных рамках)

Эксперименты с prompt

Теперь попробуем поменять деталь, которую будем редактировать. Пусть теперь это будет не цвет волос, а эмоции. Сделаем prompt = "Sad" и оставим все остальные параметры прежними. Посмотрим на получившийся результат

По какой-то причине получился абсолютно неожиданный результат (выражение лица вообще не поменялось на грустное, зато посветлели волосы и лицо). Возможно, это из-за того, что запрос состоит из очень маленького числа букв, поэтому попробуем сделать промпт с тем же смыслом, но с большим объёмом содержания.

prompt = "Very sad man"

prompt = "A face full of sadness and despair"

Не могу понять, почему получились именно такие результаты, вижу здесь только следующую взаимосвязь: в первом случае он сделал лицо более похожее на «man», а во втором – редактировал именно «face».

Давайте поменяем на prompt = "Smile" и посмотрим на результат.

Уже куда лучше! Но лицо поменялось сильнее чем хотелось бы. Проведём пару итераций с изменённым l2_lambda (с уменьшенным и увеличенным).

l2_lambda = 0.005

l2_lambda = 0.02

Ожидаемые результаты, но попробуем подобрать более оптимальное значение.

l2_lambda = 0.015

Теперь попробуем поменять пол человека (сделаем prompt = "Nice woman"), и проведём несколько экспериментов с различными l2_lambda

l2_lambda = 0.01

l2_lambda = 0.015

l2_lambda = 0.02

Как видим, получились весьма ожидаемые результаты.
Теперь проведём эксперимент, где поменяем 2 параметра за раз (возраст и цвет волос)

prompt = "Young boy with red hair", l2_lambda = 0.01

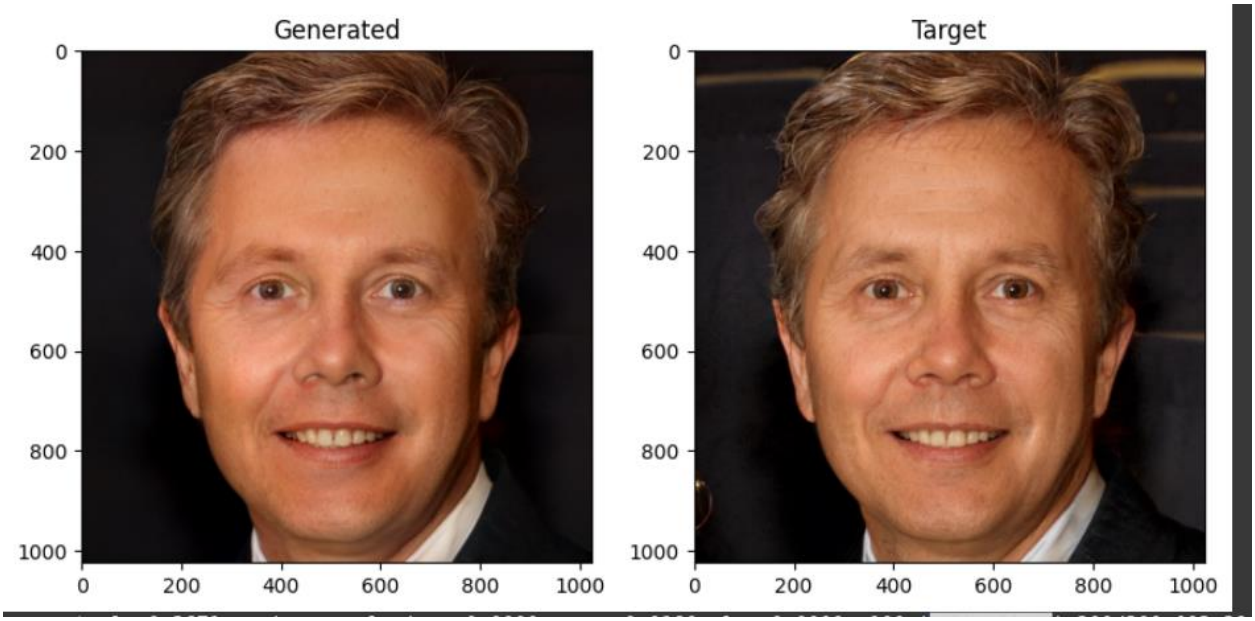
Получилось весьма прилично, при этом даже удалось сохранить неплохую схожесть с начальным лицом, но проведём ещё один эксперимент.

prompt = "Young boy with red hair", l2_lambda = 0.008

Здесь уже лицо сильно посветлело, видимо l2_lambda = 0.01 – было оптимальным.

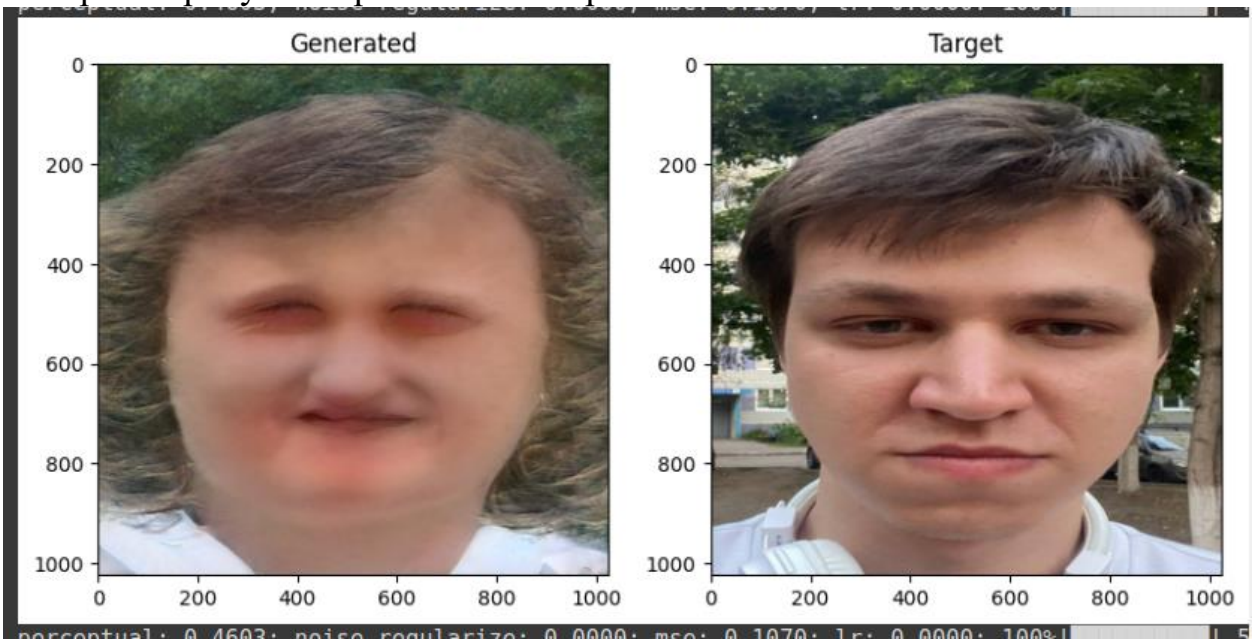
Инверсия

Попробуем сделать инверсию изображения, сгенерированного самим StyleGAN, а потом реального изображения.



Как видим, на сгенерированном – результат весьма неплох (и это на небольшом количестве шагов - 300)

Теперь попробуем на реальном изображении:



Результат получился… Мягко говоря - не очень. В целом – ожидаемо, т.к. картинка не из распределения StyleGAN2.

Выводы

Про **L2 коэффициент** можем сделать следующие выводы – это основной коэффициент, с помощью которого можно регулировать качество редактирования. При большом значении этого коэффициента – изображение может стать очень непохожим на начальное, а при маленьком – не применится промпт. Также можно сказать, что к каждому изображению и промпту нужно подбирать своё значение этого коэффициента (но в целом – оно будет порядка 0.01)

Про **ID коэффициент** можем сказать следующее: он не так сильно влияет, как предыдущий коэффициент (возможно, ошибки в моей собственной реализации), однако всё-таки немного регулирует оптимизацию и не даёт изображению слишком сильно уйти от начального.

Learning Rate – почти не влияет на качество оптимизации

Количество шагов – почти не влияет на качество оптимизации (при учёте что значение в разумных пределах)

Prompt – может сильно повлиять на итоговый результат. На некоторые промпты не получается качественный (или просто ожидаемый) результат – нужно отлаживать, например, заменить описание более подробным. Имеет место редактирование сразу нескольких деталей.

Инверсия (эдитинг реальных изображений) – хорошо работает, если взять какое-то сгенерированное изображение, но на реальных – результат оставляет желать лучшего.