

Отчет по проекту “CLIP-Guided Domain Adaptation of Image Generators”

Постановка проблемы

Генерация изображений является сложной задачей, особенно если речь идет об условной генерации – получении изображения с заданными характеристиками. StyleGan2 – архитектура, которая хорошо справляется с задачей генерации лиц с разными атрибутами, благодаря работе с латентным пространством атрибутов. С помощью манипуляций с этим пространством можно задавать разные условия и создавать изображения с нужными признаками. Чтобы задавать нужные условия с помощью текста, используются мультимодальные модели, такие как CLIP, которые умеют работать с текстом и изображениями одновременно, отображая их в общее пространство. Одним из подходов к редактированию изображений является объединение StyleGan и CLIP, что позволяет описывать нужные изменения естественным языком. К таким подходам относится StyleCLIP. Он основан на минимизации CLIP-расстояния между изображением и текстовым описанием. Однако данный подход страдает от важного недостатка: ограниченности доменом генератора. Он позволяет получить только те изменения, на которых обучался генератор, и не способен генерировать объекты, которые не видел генератор, а также адаптировать изображения к стилям, которых не было в данных для обучения генератора.

Описание решения

В данной работе мы воспроизводим результаты статьи “CLIP-Guided Domain Adaptation of Image Generators”, авторы которой предложили подход к редактированию изображений, основанный на дообучении самого генератора, используя только знания, заложенные в CLIP модели. В основе обучения лежит Directional CLIP loss – функция потерь, которая основана на CLIP loss. Суть этого метода в том, чтобы определить некоторое направление в CLIP пространстве между исходным изображением и редактированным и двигаться только в этом направлении. Таким образом, исходное изображение приближается к целевому и при этом не смещается сильно от первоисточника. Для этого рассчитываются расстояния между CLIP эмбеддингами исходного и целевого описания, а также CLIP эмбеддингами исходного и целевого изображения. Direction loss задается как CLIP loss между двумя этими расстояниями. Для генерации изображений используется StyleGan. Модель учитель генерирует исходное изображение, модель-ученик – измененное, веса этой модели оптимизируются с помощью Direction loss, позволяя генерировать на основе исходного изображения его вариант с нужными модификациями (стиль, макияж и тд.). В этой работе мы попробуем использовать такой подход, чтобы превращать человека в Джокера.

Описание экспериментов

Для экспериментов была реализована архитектура StyleGAN-NADA, а также написан Direction loss. В качестве генератора был использован StyleGan2 (код взят из <https://github.com/roinality/stylegan2-pytorch>), также была использована CLIP модель ViT-B/32.

Параметры:

- Основные параметры оптимизации: оптимизатор Adam с lr = 0.002 (взято из рекомендаций авторов в самой статье), размер батча 2, кол-во картинок 4
- Разморозка слоев: все слои (основано на рекомендации авторов для texture-based изменений, а также худшем визуальном качестве при разморозке меньшего количества слоев)
- Количество итераций оптимизации: 100–200

Также всего было попробовано 5 разных промптов:

- Joker
- Joker face
- Joker smile
- Joker makeup
- Joker cosplay

Описание результатов

Прежде всего, было замечено, что при обучении на 1–2 картинках модель переобучалась, то же самое происходило если обучать больше 100 эпох. В целом, 100 — это адекватное количество итераций для данной задачи, поэтому все финальные эксперименты проводились с таким параметром. Также изменение достаточно сильное, поэтому были разморожены все слои: при разморозке меньшего количества слоев даже при увеличении итераций качество визуально хуже — для такого изменения этого недостаточно. Learning rate также не менялся сильно, так как ничего лучшего найти не удалось (только хуже) — поэтому следуем рекомендациям авторов.

Далее посмотрим на результаты с разными промптами на реальном изображении (много примеров можно найти в ноутбуке с обучением и инференсом в репозитории https://github.com/alexandraizhevskaya/image_editing/tree/main):

“Joker”

Aligned image has shape: (256, 256)
Orig aligned VS edited



“Joker face”

Aligned image has shape: (256, 256)
Orig aligned VS edited



“Joker smile”

Aligned image has shape: (256, 256)
Orig aligned VS edited



“Joker makeup”

Aligned image has shape: (256, 256)
Orig aligned VS edited



“Joker cosplay”

Aligned image has shape: (256, 256)
Orig aligned VS edited



В целом, видно, что везде есть изменения, то есть модель обучается и при этом сохраняется приближенность к оригиналу, хотя и в разной степени. Наиболее удачной версией и самой близкой к Джокеру является наиболее лаконичный промпт “Joker”, видимо, помогает, что не вносим лишних смыслов, на которые модель отвлекается. “Joker face” тоже похож, но цвет лица остается ближе к первоисходнику, он не совсем соответствует Джокеру. В случае с “Joker makeup” и “Joker cosplay” модель явно еще сдвигается в направлении “makeup” и “cosplay”, соответственно получается более сильный акцент на сам макияж, в частности, на глаза. В косплее еще что-то случилось с волосами. В случае с “Joker smile”, ожидаемо, акцент на улыбку – она получилась большой и само выражение лица более веселое – не совсем про улыбку Джокера, тем не менее улыбка. Во всех примерах, даже без упоминания smile, люди начинают улыбаться. То есть в модели CLIP заложено знание про то, что Джокер славится не только определенным макияжем, но и своей улыбкой, и наша модель оптимизируется под это.

Выводы

Модель способна редактировать изображения и выдавать изображения людей в образе Джокера. Самый удачный промпт “Joker” выдает неплохой вариант и образ, похожий на киногероя и одновременно приближенный к оригиналу. В случае с остальными промптами грим больше отличается от персонажа, но в целом тоже выглядит узнаваемо. Дообучить лучше не удалось, так как при увеличении итерации (с и без уменьшения lr) модель начинала переобучаться и вместо улучшения макияжа терялись черты лица человека на исходной картинке - получались примерно одинаковые лица Джокера.

В целом, StyleGAN-NADA – хороший эффективный подход для редактирования изображений, который действительно работает на практике. Его главное достоинство – это отсутствие необходимости в датасете и больших вычислительных ресурсах для обучения. Модель полагается на знания других предобученных моделей, что позволяет за несколько итераций оптимизации на сгенерированных изображениях быстро получить качественный результат.

К недостаткам, однако, можно отнести то, что такой подход все еще ограничен знаниями, заложенными в другую модель (CLIP). Если бы CLIP не имела знаний о том, как выглядит Джокер, обучить генератор для такой модификации было бы невозможно. Также текстовые описания могут быть неоднозначными, что затрудняет обучение и может привести к неожиданным результатам. Даже на примере рассмотренной задачи можно заметить, что добавление в промпт слов “makeup” и “cosplay” уже меняло направление оптимизации, отвлекая от основного направления “Joker”.