

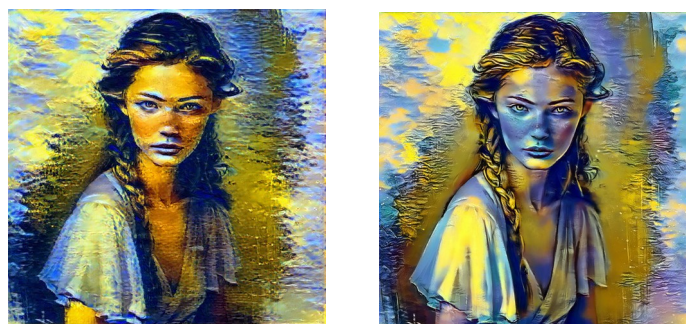
Отчет по проекту telegram-bots

Основные шаги:

1. Изучены предложенные задачи и как более интересная выбрана задача переноса стиля (NST).
2. Изучены предложенные ссылки и статьи. Заинтересовал алгоритм Леона Гатиса, поскольку реализация была достаточно понятной и имелось отличное руководство https://pytorch.org/tutorials/advanced/neural_style_tutorial.html
3. Было проверено базовое решение предложенное в ноутбуке на мощностях Colab.
4. Поскольку работа достаточно старая(2015г), мне стало интересно попробовать использовать в алгоритме более продвинутую CNN нежели VGG19 актуальную на 2015г. Были рассмотрены различные известные CNN, наиболее удачный эксперимент удался с ResNet50.
5. Создание модели на ResNet, отличие архитектуры заставило значительно менять построение модели для оптимизации, поскольку resnet уже нельзя было представить как Sequential. В VGG использовалось лишь 5 первых сверток, поэтому я решил прописать все пути модели в ручную, но к сожалению оказалось что 5ти слоев ResNet слишком мало, в итоге долго экспериментируя и увеличивая кол-во используемых выходов удалось получить хорошие результаты при использовании 13 conv слоев.
6. Тестирование и сравнение моделей. Оказалось что действительно поведение моделей заметно отличается (для чистоты эксперимента были проверены различные конфигурации VGG, что показало что различия связаны не с этим). Модели на VGG меньше затрагивали текстуру объектов сильнее воздействуя на фон, в тоже время модели на ResNet в равной степени изменяли картинку, иногда удавалось получить сходство при очень большом кол-ве шагов оптимизации для VGG, ниже приводиться



наиболее яркий пример, когда эталоном стиля служит просто текстура. Первое изображение создано с помощью ResNet алгоритм пытался в равной степени изменить структуру как фона так и девушки, в тоже время VGG раскрасил фон, но девушка мало изменилась, с точки зрения полноты переноса модель ResNet пыталась сделать более правильно, хотя конечно визуальный результат получился хуже. Еще один пример:



Снова виден тот же эффект VGG мало затрагивает структуру объектов имеющих изначально четкие границы, увеличивая глубину сети можно добиться увеличения

уровня геометрических и структурных изменений с VGG но это отрицательно скажется на производительности, учитывая что ResNet имея более чем в 2 раза больше слоев работает в 2 раза быстрее позволяя использовать больше шагов оптимизации.

7. Выбор модели. Было принято решение оставить обе модели и сделать возможность выбора пользователю.
8. Борьба с нестабильностью. В некоторых случаях возникала сильная расхожимость и скачки лосов, для борьбы с эти был опробован gradient clipping и использование шедулера для уменьшения шага оптимизатора при скачках лоса, второе решение было применено в финальной модели, поскольку меньший шаг оптимизатора действительно упрощал сходимость модели. Так же много сил был потрачено на подбор начальной скорости оптимизации и отношения style и content loss.
9. Также были попытки ускорения модели засчёт квантизации, но они оказались провальны fp16 давала не так много, а int8 не имел градиентов.
10. Была выбрана модель SRGAN для доп части и интегрирована почти в готовом виде <https://github.com/mseitzer/srgan>.
11. Для создания бота был выбран telebot поскольку в прошлом уже работал в ней над простыми ботами, что конечно помогло быстрее написать решение, но вызвало появление большего кол-ва проблем и нюансов которые раньше не встречались из-за простоты проектов, все они были закрыты не самым оптимальным образом, из-за чего код сильно удлинился.
12. Обертка в класс. В представленном решение был создан родительский класс имеющий методы переноса стиля, а также зашитые сценарии бота, что возможно имелось ввиду вынести из класса. Затем были написаны классы наследники для различных конфигураций только vgg / только resnet / оба + super_res.
13. Настройка всех моделей для работы с сри, опробован хотсинг на различных машинах.
14. Поиск нюансов и ошибок, добавление обработки ошибок в сценариях чатов
15. загрузка модели на гитхаб

Основные трудности:

Главной сложностью было написание и отработка проблем в сценариях бота, а также попытки с другими спп, из-за ручного переноса resnet код стала крайне длинным. Много сложностей было связано с подбором гиперпараметров и конфигураций чтобы победить нестабильность обучения.

Удачные примеры:

