

ChromaGAN Project

Евгения Сивец

Проект основан на статье <https://arxiv.org/pdf/1907.09837.pdf>.

В проекте была реализована модель по колоризации черно-белых изображений.

Цель - по L-каналу (Luminance) изображения в пространстве CIE Lab восстановить два канала цветности: каналы a и b.

Архитектура сети

Общая архитектура состоит глобально из двух блоков: генератора и дискриминатора. Генератор принимает на вход 3 склеенных L-канала изображения, и выдает два выхода: первый выход генератора - (a, b) – каналы изображения, второй выход генератора возвращает распределение входных изображений по классам в ImageNet. Дискриминатор берет на вход изображение $H \times W \times 3$ в цветовом пространстве CIE Lab и возвращает валидность сгенерированной раскраски.

Более подробное рассмотрение архитектуры:

Посмотрим на рисунок архитектуры модели:

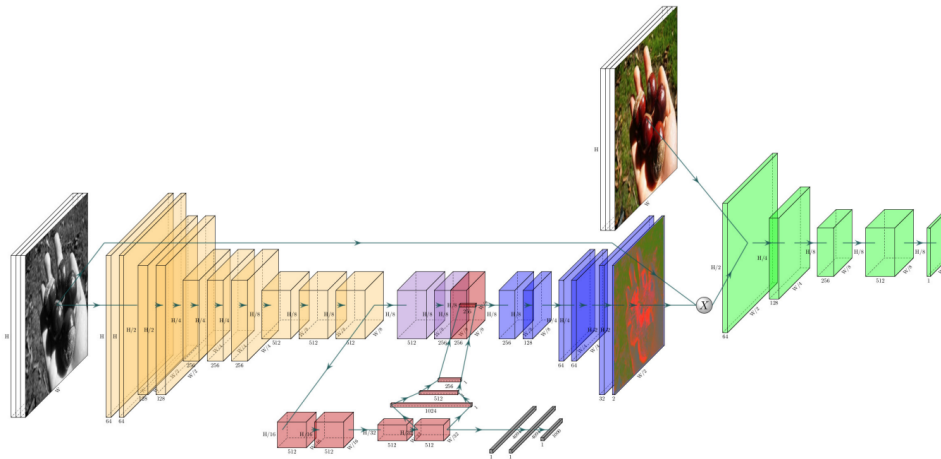


Рис. 1: Архитектура модели

Генератор принимает на вход чб изображение $3 \times L$. Первые слои генератора повторяют архитектуру VGG-16 (это желтые блоки), состоящие из последовательно сверток, relu. После желтых блоков выходы генератора разделяются на два потока. В первом (фиолетовом) потоке - Conv-BatchNorm-Relu, в красном потоке снизу - блоки Conv-BatchNorm-Relu, при этом есть разделение: серая часть возвращает распределение по классам Imagenet, красная часть из последовательности линейных слоев (это доп. фичи из VGG-16) размывается по размеру фиолетового блока и приклеивается к основному потоку генератора. Далее идут апсемплинг блоки и на выходе генератора получаются два цветных канала (a,b).

Зеленая часть на схеме архитектуры модели - это дискриминатор, он основывается на архитектуре PatchGAN-a со свертками размера 4, BatchNorm, LeakyRelu.

Loss functions

Целевая функция обучения - сумма 3 лоссов: color error loss - L_2 —норма между истинными каналами цветности и сгенерированными, adversarial Loss with GP - лосс wasserstein gan-a с gradient penalty, class distribution loss - классификаторный лосс, считается как KL-дивергенция между серыми выходами генератора и предсказаниями предобученного классификатора (используется VGG-16).

Обучение модели

Эксперименты логировались в [neptune.ai](https://ui.neptune.ai) можно посмотреть по ссылке:

<https://ui.neptune.ai/calistro/chromagan/experiments?viewId=standard-view>.

Авторы статьи обучали модель на всем ImageNet, но поскольку у меня нет таких ресурсов по памяти, я обучала модель на датасете TinyImageNet. Чтобы приспособить свой датасет под данную архитектуру и задачу я увеличила картинку до размера 224x224 (на таком размере обучался предобученный классификатор) и сделала соответствие между 200 классами в TinyImageNet и 1000 соответствующими классами в ImageNet. Для получения предсказаний предобученного классификатора на TinyImageNet брались предсказания на Imagenet, после чего из полученного вектора логитов размера 1000 выделялись 200 нужных логитов TinyImageNet-a, после чего применялся Softmax.

Для получения черно-белых картинок из датасета TinyImagenet делалось преобразование из RGB в CIE Lab, чтобы получить (L, a, b)-каналы. Данное преобразование делалось с помощью библиотеки skimage. Это преобразование делалось на вход генератору и на вход дискриминатору соответственно (дискриминатор берет на вход оригинальное изображение в формате CIE Lab и выход генератора (a, b) сконкаченный с L-каналом input-a).

Для визуализации сгенерированной раскраски в процессе обучения делалось обратное преобразование из CIE Lab в RGB также с помощью skimage.

Было проведено несколько экспериментов. Во время первого эксперимента генератор и дискриминатор обучались 3 эпохи с $lr=2e-4$ - эксперимент GHROM-8, второй эксперимент - поставила меньший $lr=2e-5$ и обучила 4 эпохи. Во время второго эксперимента постоянно возникали технические сложности на вычислительном сервере, приходилось дообучать каждую эпоху сохраненную модель - эксперименты GHROM-11 - GHROM-15. Метрика качества PSNR. В первом эксперименте PSNR=23.5, во втором вышло лучше: PSNR=24.5.

Раскрашенные изображения из тестовой выборки логировала каждую эпоху (по два grid-изображения из двух разных батчей на эпоху). Их можно посмотреть в логах в колонке colorization.

Также добавлен эксперимент с раскрашиванием чб фотографии без GT данных, добавлено сравнение с сервисом у Mail <https://9may.mail.ru/restoration/>.