

Отчет по итоговой работе по курсу «Профессия ML-инженер»

Макаров И.В.

Задача

Тема - Раскрашивание черно-белых изображений

Необходимо:

- Перевести изображение в наиболее подходящее цветовое пространство
- Обучение глубокой модели, например, U-Net, для раскрашивания
- Оценка на стороннем датасете

Обзор существующих решений

[DeOldify](#) - Одна из самых популярных и качественных моделей для колоризации изображений. Онлайн-версии - [DeepAI](#) или [Algorithmia](#).

Особенности :

- Работает на основе GAN (генеративно-состязательных сетей);
- Может работать в режиме "устранения артефактов" и "реалистичной" колоризации;
- Есть предобученные вес.

[DeepAI Colorizer](#) - Веб-сервис, использующий нейросеть для колоризации.

OpenCV + предобученные модели

Выбор датасета

Были выбраны два датасета с платформы Kaggle:

- [theblackmamba31/landscape-image-colorization](#)
- [aayush9753/image-colorization-dataset](#)

Первоначальная гипотеза была в том, что нужны датасеты, в которых представлены пары черно-белых и цветных изображений. Однако, в дальнейшем это оказалось неважным, т.к. при обучении используются только цветные изображения.

Два датасета выбраны в связи с тем, что в landscape-image-colorization представлены преимущественно пейзажи, а в image-colorization-dataset – люди, животные, техника, предметы и т.д.

В результате для обучения собирается объединенный датасет – как показали эксперименты это дает более лучший результат.

Выбор метода решения

После ряда экспериментов был выбран следующий подход:

- Изображения переводятся из RGB в цветовое пространство LAB
- Канал яркости (L) используется в качестве источника для предсказания цветовых каналов (ab)
- Для предсказания используется предобученная модель InceptionResNetV2 в качестве энкодера и сверточного декодера. Энкодер извлекает высокоуровневые признаки из черно-белого изображения, а декодер восстанавливает цветовые компоненты (a и b) с использованием upsampling-операций и сверточных слоёв.
- Итоговое цветное изображение формируется путём объединения предсказанного цвета (ab) с исходной яркостью (L) и преобразования из Lab в RGB.
- Применяются пространственные аугментации (случайный кроп, поворот, отражение) на этапе обучения. Цветовые аугментации не применялись, чтобы не нарушить связи яркости и цветов.

Архитектура и оценка сложности модели

Архитектура модели:

- Энкодер : InceptionResNetV2 (предобученная на ImageNet, извлекает признаки на разных уровнях детализации)
- Декодер : последовательность из сверточных слоёв и upsampling-операций:
 - 5 блоков: Conv2d + ReLU + Upsample
 - Последний слой выдает 2 канала (a и b)
 - Финальный upsampling до размера 400x400

Сложность модели:

- InceptionResNetV2 содержит около 56 миллионов параметров.
- Декодер добавляет порядка 8.5 миллионов параметров.

Общая сложность модели — около 64.5 миллиона обучаемых параметров, что должно быть достаточно для извлечения сложных признаков, но требует значительных вычислительных ресурсов.

Детали обучения

Оптимизатор : Adam (lr=2e-4, betas=(0.5, 0.999))

LR-шедулер : ReduceLROnPlateau (пациенс 3) — снижение скорости обучения при стагнации валидационной ошибки

Функция потерь : MSE

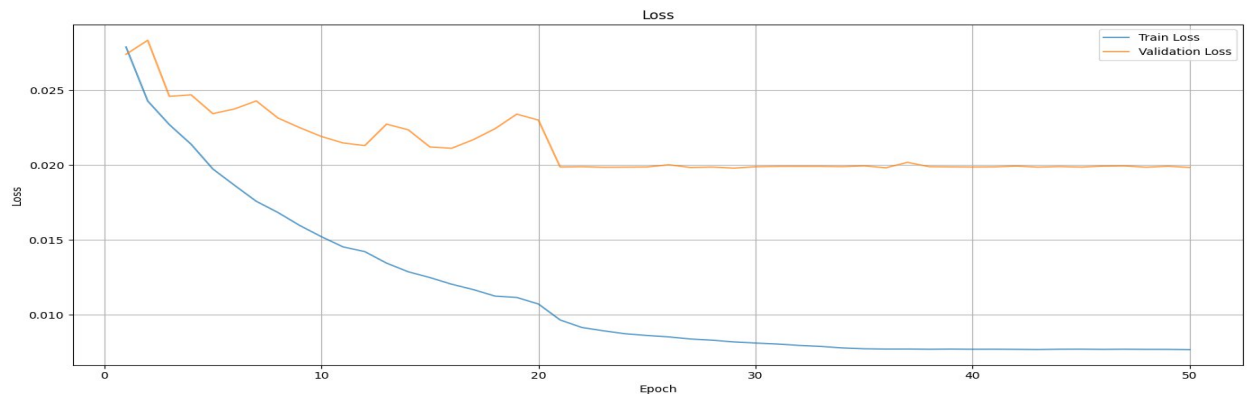
Размер батча : 16

Количество эпох : 50

Обучение : проводилось на GPU GeForce RTX 4060 Ti, 16Gb

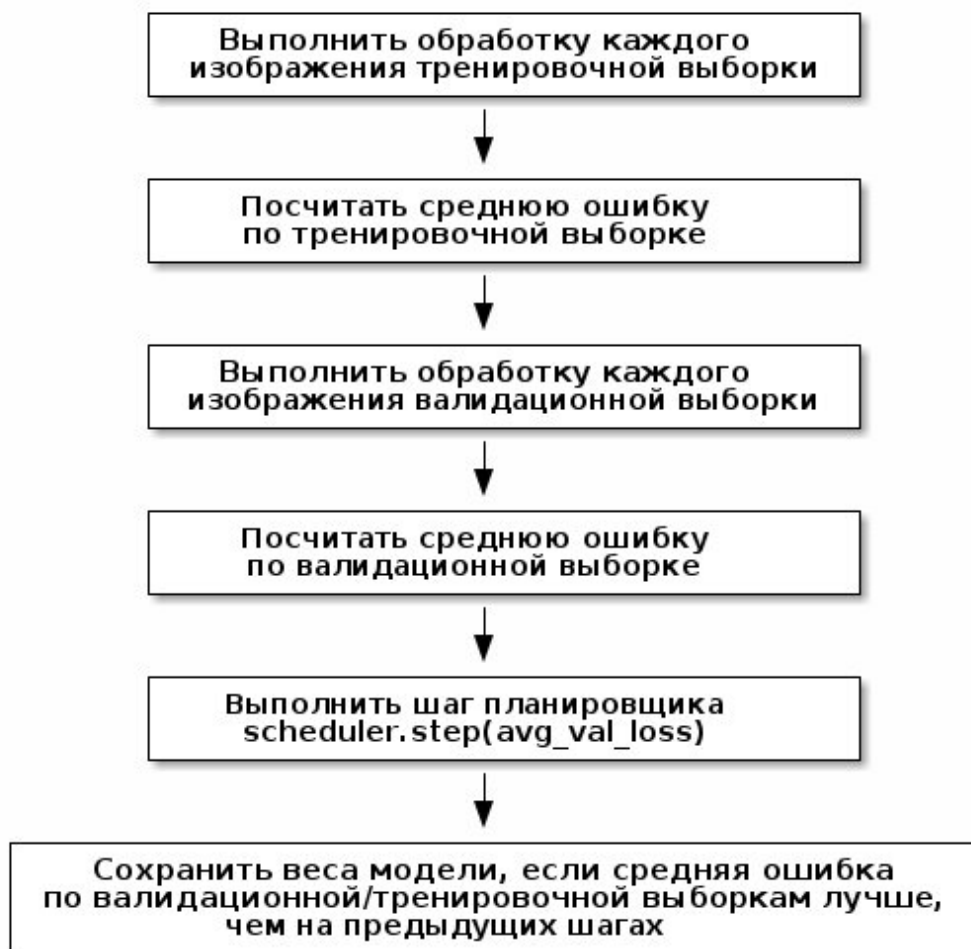
Сохранение модели : лучшие модели сохранялись по метрикам на обучающей и валидационной выборке

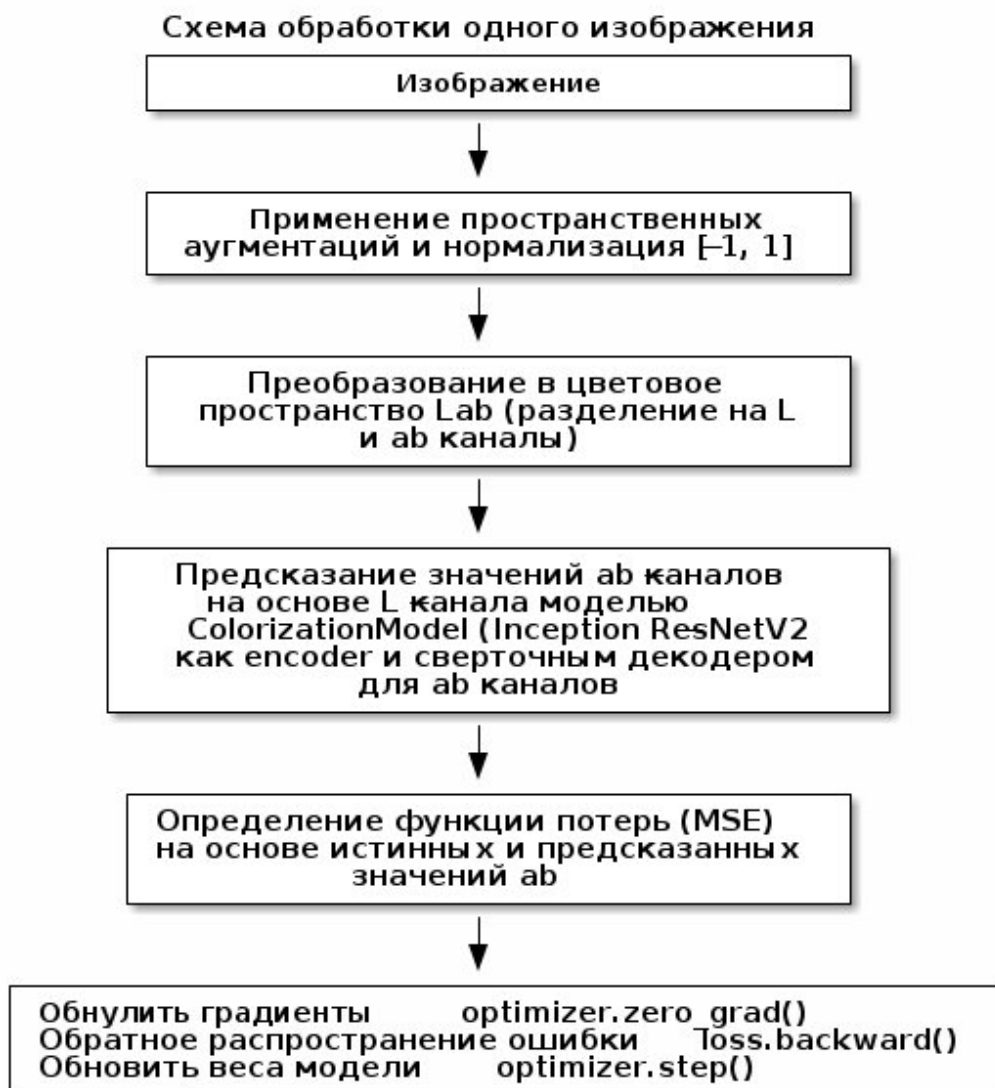
График функции потерь.



Примерно после 21-22 эпохи снижение значения loss почти прекратилось. Особенно на валидационной выборке. Предположительно это говорит о переобучении модели.

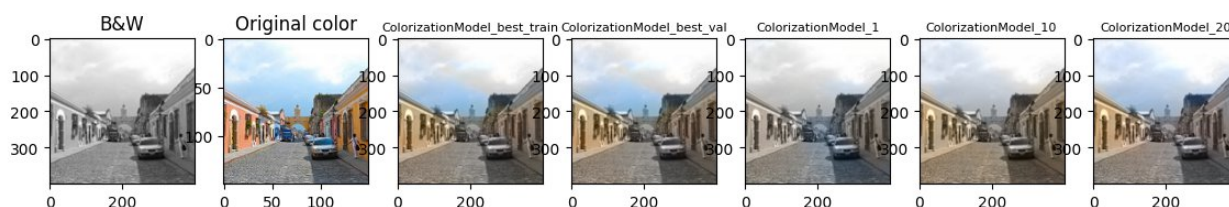
Схема обучения одной эпохи





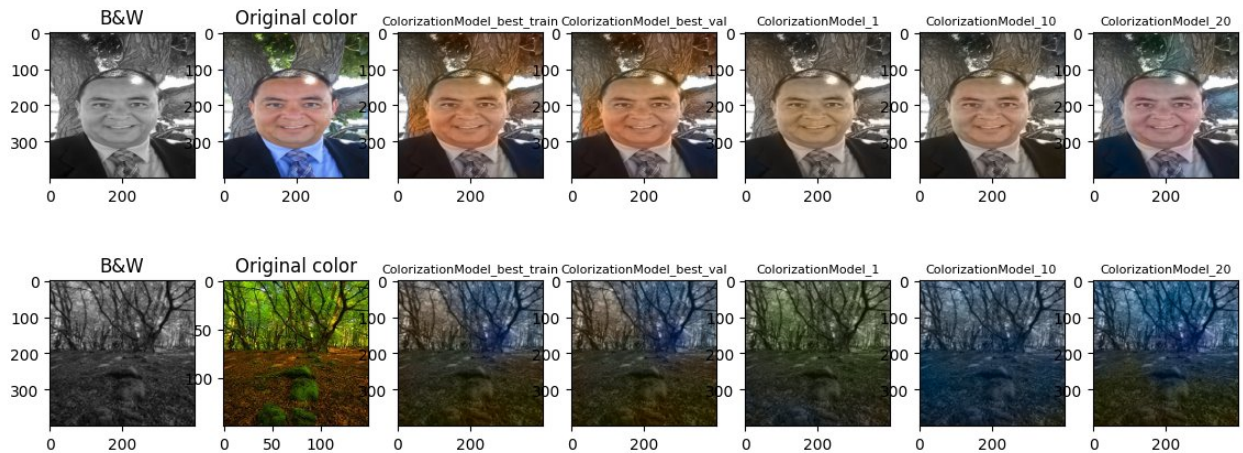
Выводы о пригодности подхода к решению выбранной задачи

Реализованная модель неплохо справляется с раскрашиванием относительно простых изображений





Несколько сложнее – со сложными изображениями, с фотографиями людей.



В целом можно сказать, что выбранный подход пригоден для выполнения поставленной задачи, хотя и не без недостатков:

- Цвета, в основном, получаются неяркие
- Есть склонность уходить в коричневые цвета и слабо раскрашивает в красный цвет

Имеет смысл попробовать доработать. Например:

- Реализовать GAN, в которой текущая модель будет генератором — предположительно сделает цвета более реалистичными и улучшает восприятие.
- Добавить skip connection и/или self-attention в декодер
- Разнообразить датасет
- Попробовать разные функции потерь – L1Loss, SSIM (Structural Similarity Index Measure), Perceptual Loss, Комбинация MSE и L1
- Попробовать разные оптимизаторы и планировщики