



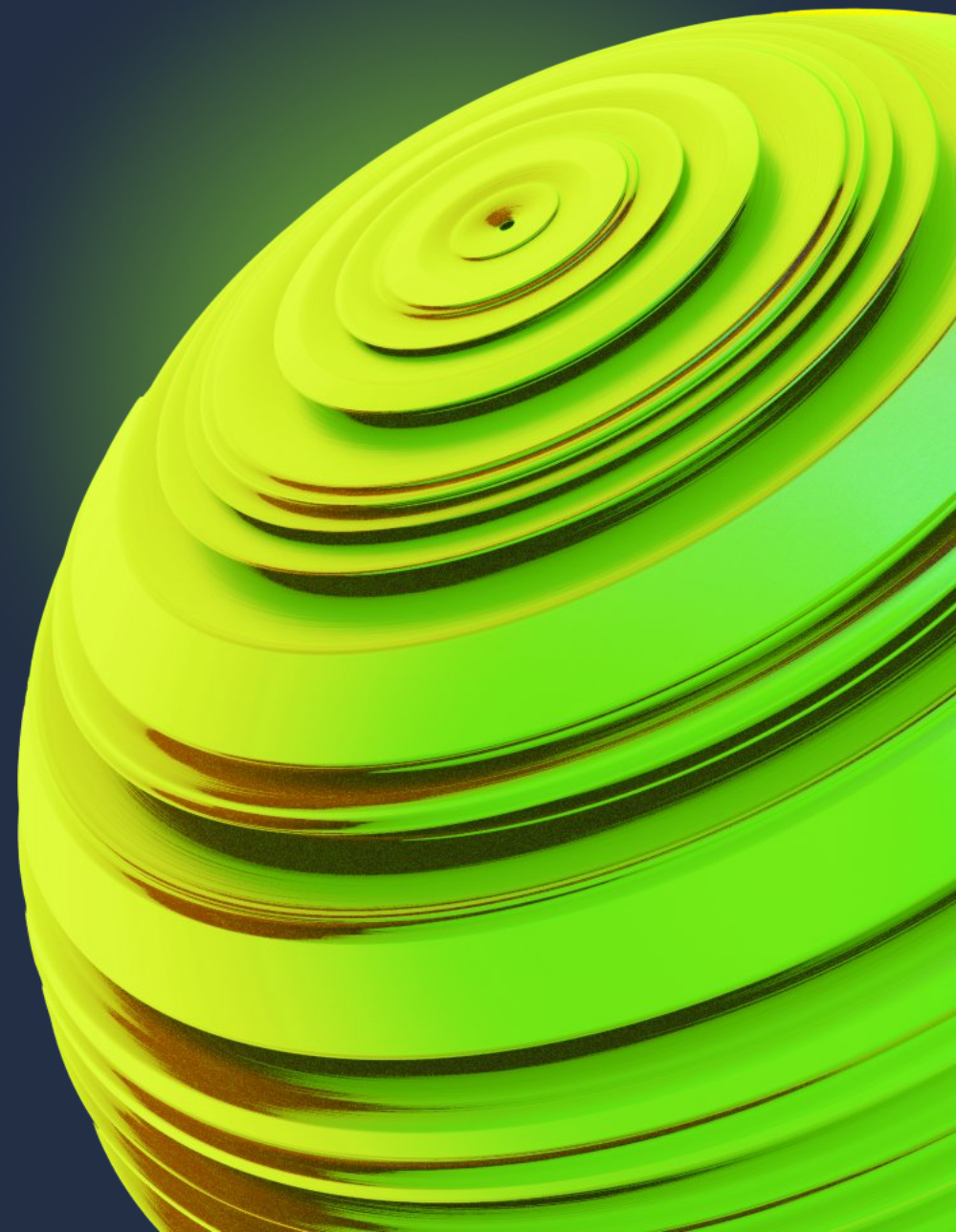
ИНСТИТУТ  
ДОПОЛНИТЕЛЬНОГО  
ОБРАЗОВАНИЯ  
УНИВЕРСИТЕТА ИННОПОЛИС



УНИВЕРСИТЕТ  
ИННОПОЛИС

## Раскрашивание черно-белых изображений

Итоговая аттестационная работа по курсу «Профессия ML-инженер»





## Цель работы

- Обучить глубокую модель раскрашивающую черно-белые изображения
- Создать UI-приложение, позволяющее раскрашивать черно-белые изображения

# План работы



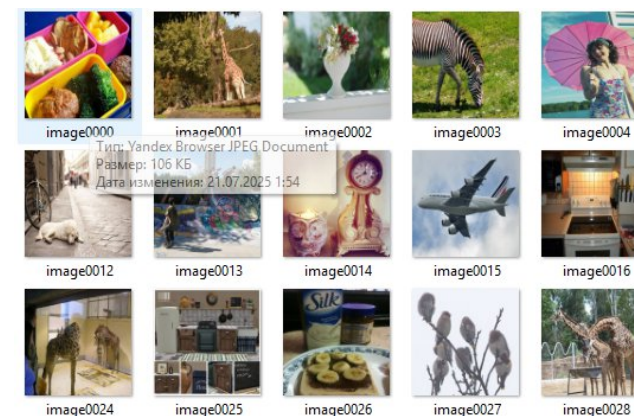
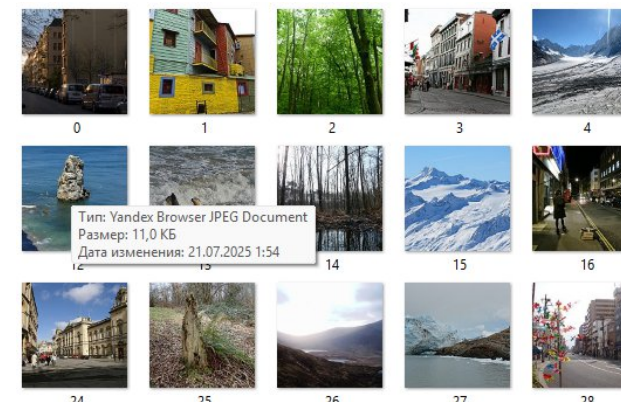
- Выбор подходящего датасета
- Выбор метода раскрашивания
- Разработка и обучение модели, раскрашивающей черно-белые изображения
- Создание UI-приложение, позволяющее раскрашивать черно-белые изображения
- Deploy приложение в виде docker-контейнера

# Подбор датасета

С платформы kaggle было загружено два датасета:

- [Landscape color and grayscale images](#) – содержащий изображений улиц, зданий, гор, ледников, деревьев и т. д.
- [Image Colorization Dataset](#) – содержащий достаточно много изображений людей, животных и предметов

Эксперименты показали, что лучших результатов удалось добиться при сочетании обоих датасетов - весь объем датасета landscape-image-colorization и подмножество train датасета были объединены в общий датасет, который в соотношении 80/20 был разделен на тренировочную (9703 изображения) и валидационную (2426 изображения) выборки.



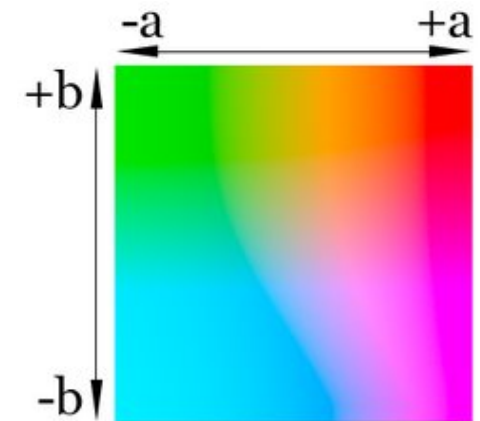
# Выбор метода



Первоначальная идея была использовать черно-белые изображения из датасета в качестве источника, а цветные – в качестве цели.

Но после изучения рекомендуемых подходов был выбран следующий метод:

- переводим цветное изображение в цветовое пространство [LAB](#)
- канал L (яркость) используем как черно-белое изображение в качестве источника
- цветовые каналы ab используем как цель
- после предсказания снова переводим в RGB





# Выбор модели

В ходе экспериментов проверялось несколько вариантов Unet моделей, а также вариант генеративно-сопоставительной сети (GAN, Generative Adversarial Network).

В результате за основу была взята идея из статьи [Раскрашиваем чёрно-белую фотографию с помощью нейросети из 100 строк кода](#) – Unet-модель, у которой:

- в качестве encoder используется предобученная [Inception-ResNet\\_v2](#) в реализации из библиотеки [timm](#).
- В качестве decoder - сверточной сети из серии блоков Conv2d+ReLU+Upsample



# Обучение модели

- стек технологий для обучения модели:
  - Python 3.12.9
  - Pytorch
  - Предобученная сеть InceptionResNetV2 из библиотеки Timm
  - Albumentations для аугментации
  - Библиотеки Opencv, Kornia, PIL для работы с изображениями
  - Matplotlib для визуализации в jupyter-ноутбуке
- Обучение модели проводилось на компьютере:
  - CPU: Core(TM) i7-12700KF, 3600 МГц, ядер: 12, логических процессоров: 20
  - RAM: 64Gb
  - GPU: NVIDIA GeForce RTX 4060 Ti, 16Gb
- Длительность обучения:
  - ~5 мин на эпоху: 4:40 на этап тренировки и 0:20 на валидацию
  - 50 эпох - ~4 часа 10 мин

# Обучение модели

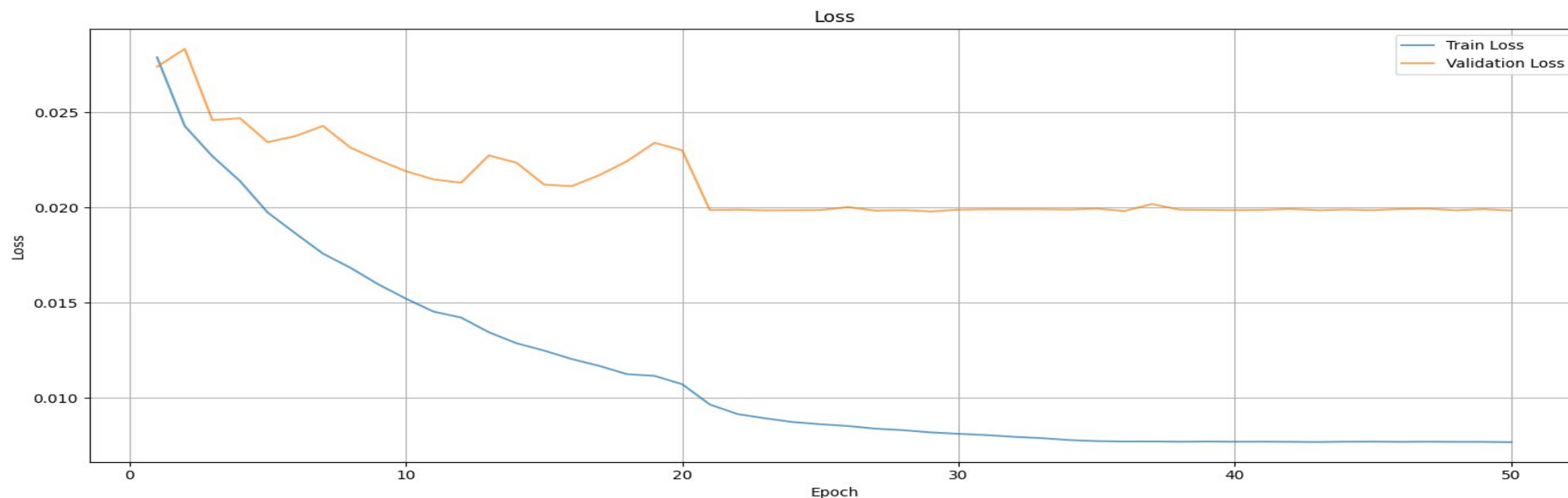


Функция потерь - среднеквадратичная ошибка (`nn.MSELoss()`)

Оптимизатор - Adam

Планировщик - ReduceLROnPlateau

Как видно на графике обучение шло 50 эпох, но при этом примерно после 21-22 эпохи снижение значения loss почти прекратилось. Особенно на валидационной выборке.





# Приложение

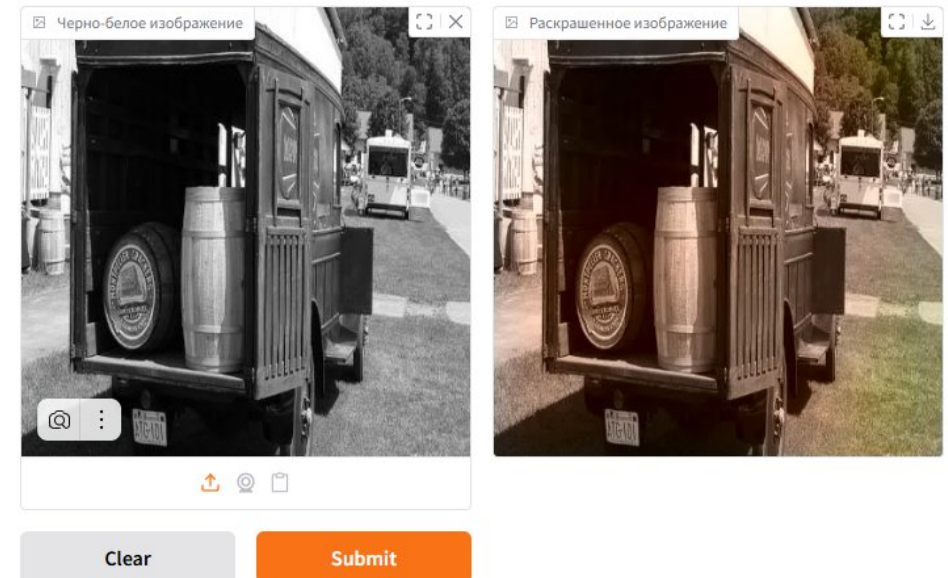


Стек технологий для приложения:

- gradio - для UI приложения
- Docker - как средство деплоя
- Библиотеки OpenCV, Kornia, PIL для работы с изображениями

## Colorization Model 09

Загрузите черно-белое изображение, и приложение раскрасит его с помощью модели ColorizationModel09.





# Возможное развитие

- Доработать модель и приложение так, чтобы могли раскрашивать изображения произвольного размера
- Реализовать GAN, в которой текущая модель будет генератором — предположительно сделае цвета более реалистичными и улучшает восприятие.
- Добавить skip connection и/или self-attention в декодер
- Разнообразить датасет
- Поэкспериментировать с функцией потерь – попробовать L1Loss, SSIM (Structural Similarity Index Measure), Perceptual Loss, Комбинация MSE и L1
- Поэкспериментировать с оптимизатором и планировщиком

# Ссылки



- Репозиторий работы - [https://github.com/igel2000/ml\\_final](https://github.com/igel2000/ml_final)



ИНСТИТУТ  
ДОПОЛНИТЕЛЬНОГО  
ОБРАЗОВАНИЯ  
УНИВЕРСИТЕТА ИННОПОЛИС

Спасибо за внимание!

Контакты

