CycleGAN

Goal: Explore style transfer using the CycleGAN architecture for images featuring Gzhel and Khokhloma patterns.

Links: GitHub, Google Collab.

Team Members: Olga Kolomyttseva and Iuliia Liutova

Architecture

We used this paper: CycleGAN architecture .

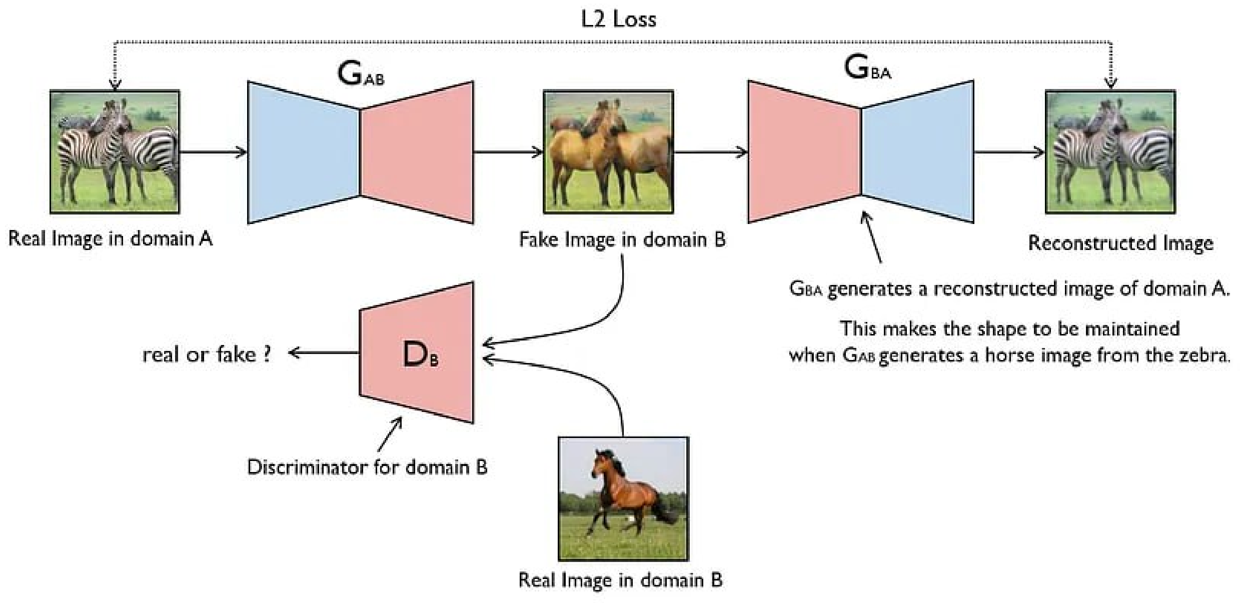
It describes a method for transforming images from one domain to another **without** the need for paired images for training.

The main idea involves the use of Generative Adversarial Networks (GANs) and the introduction of "cycle consistency" to ensure that the transformed image, when transformed back, returns to the original:

1. **Problem**: Traditional methods require pairs of images (e.g., a photo and its Monet-style version), which can be difficult to obtain for many tasks.
2. **Goal**: Develop an algorithm capable of learning to translate images from one domain (e.g., photos) to another (e.g., Monet style) without paired data.
3. **Method**:

- **Generator** G_{AB} translates images from domain A to domain B.
- **Inverse generator** G_{BA} translates images from fake domain B to domain A.
- **Discriminators** D_A and D_A distinguish real images from translated ones.

- **Cycle consistency**: Losses include cycles $(A \to G_{AB}(A) \to G_{BA}(G_{AB}(A)) \approx A)$ and $(B \to G_{BA}(B) \to G_{AB}(G_{BA}(B)) \approx B)$, ensuring that translated images can be returned to the originals.


L2 Loss
G_AB
G_BA
Real Image in domain A
Fake Image in domain B
Reconstructed Image
G_BA generates a reconstructed image of domain A.
This makes the shape to be maintained when G_AB generates a horse image from the zebra.
real or fake ?
D_B
Discriminator for domain B
Real Image in domain B

4. **Applications**:

- Style transfer (e.g., photos to Monet style) - we use for it.
- Object transformation (e.g., horses to zebras).
- Season transfer (e.g., summer photos to winter).

Dataset collection

1. Explore sites with Gzhel and Khokhloma products. I found a website with a large number of pictures for Gzhel: Гжельский Фарфоровый Завод. There is no single website with a lot of images, so I need to collect them from different sites: Хохлома, Семёновские традиции, Хохломская роспись, Арт Сувенир, Luxpodarki.ru.
2. I wrote functions for downloading .jpg images and parsing websites:
    1. download_images.py - searches for *img* tags and downloads images from *src*. You can start downloading pictures from a specified number n if the images before this number are not Gzhel

    or Khokhloma products. For fast downloading, I implemented it in parallel on 10 streams.
    2. delete_images.py - sometimes a website contains some unsuitable images at the end, so this function deletes the last n pictures.
    3. gzhel_parser.py - parses Gzhel websites.
    4. khokhloma_parser.py - parses Khokhloma websites.
3. Some pictures that are not products still end up in the folder, so I deleted them manually.
4. After downloading, there were 800 Khokhloma pictures and more than 1500 Gzhel pictures, so I deleted random images from Gzhel to get 800. random_delete.py
5. Not all images are square, so I made them square by adding white pixels. After that, I resized the pictures to 128x128.
    1. squaring.py - makes squares.
    2. resize.py - resizes to 128x128.
6. For training, I split the images into 80% for training and 20% for validation: test-train-create.py

Development

Model:

- **Generator:** ResNet
- **Discriminator:** PatchGAN

In total: 29.594.504 params
G_{AB}: 7.837.699 params
G_{BA}: 7.837.699 params
G_A: 6.959.553 params
G_B: 6.959.553 params

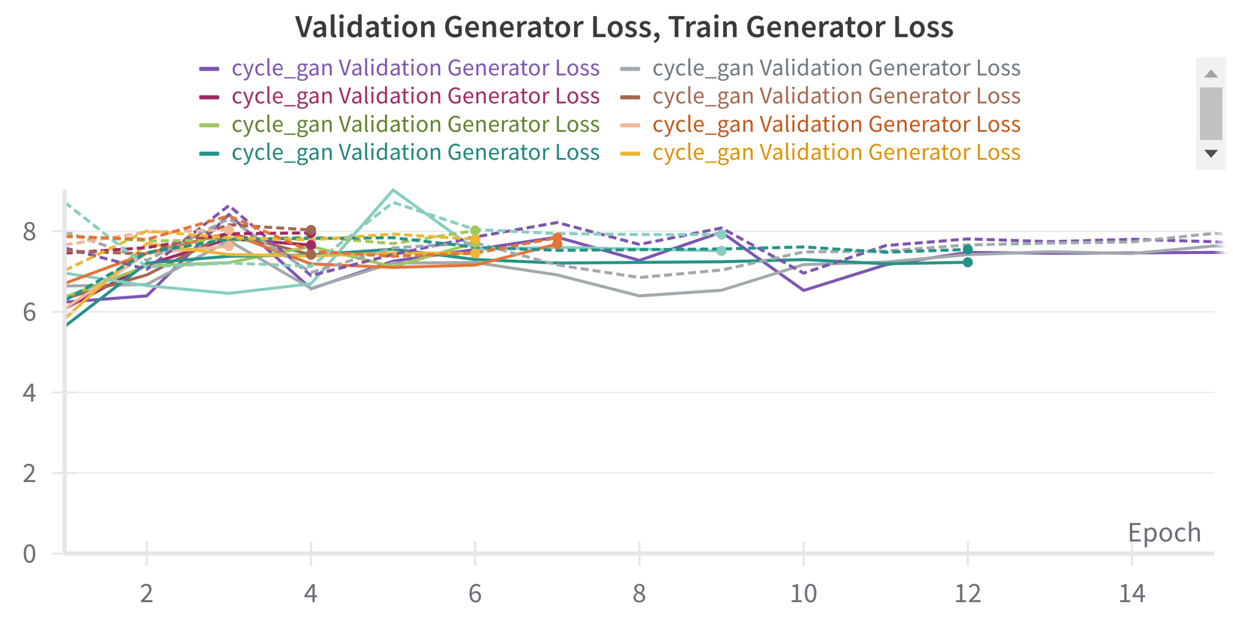
Training details:

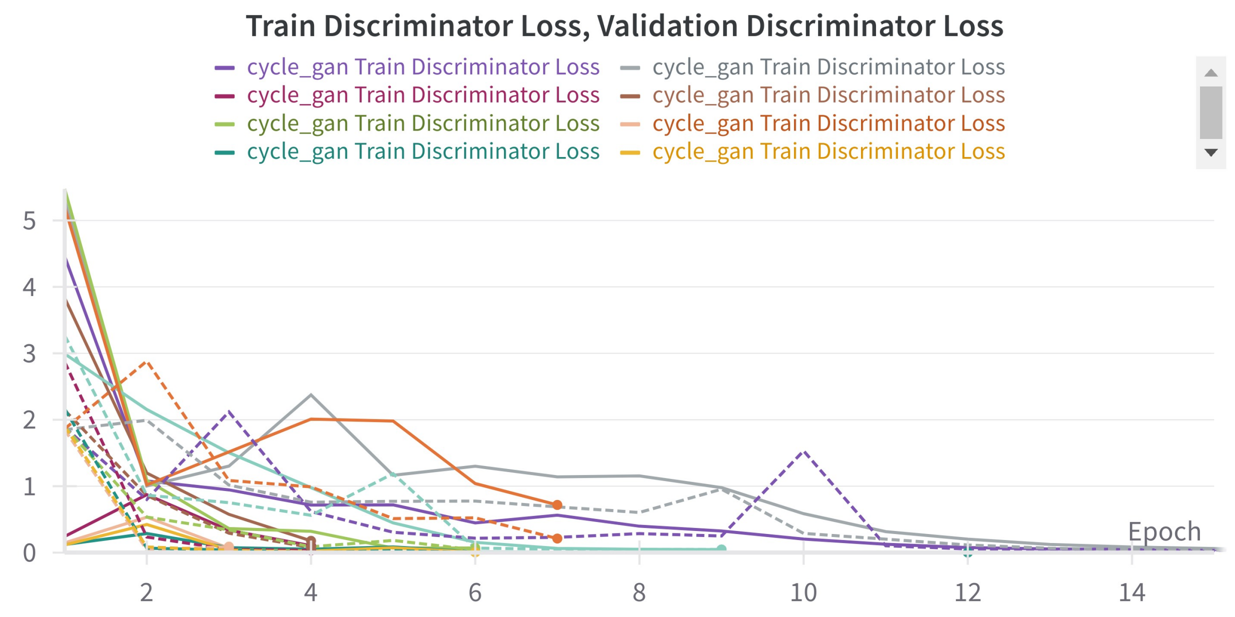
We focused on the parameters suggested in the original article.

Experiments:

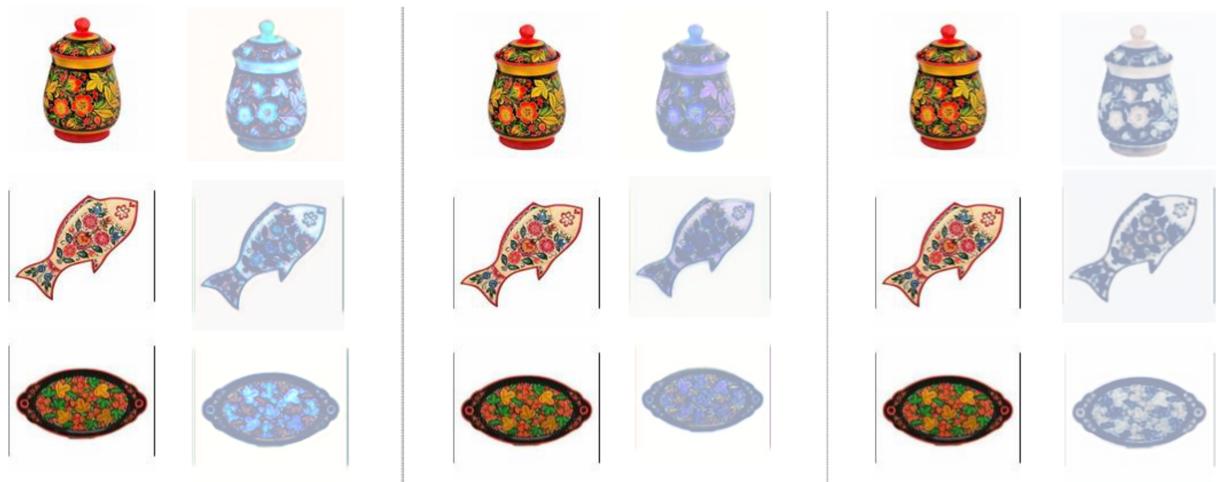
- Tried different sizes: 64x64, 128x128, stuck with the last one
- Initial setup from paper with batch size = 10
- Smaller setup with batch size = 20
- Tried tweaking learning rate; initial 2e-4, 2e-3 worked better
- Tried several schedulers
- Experimented with training the discriminator/generator more than the other one

  Metrics:



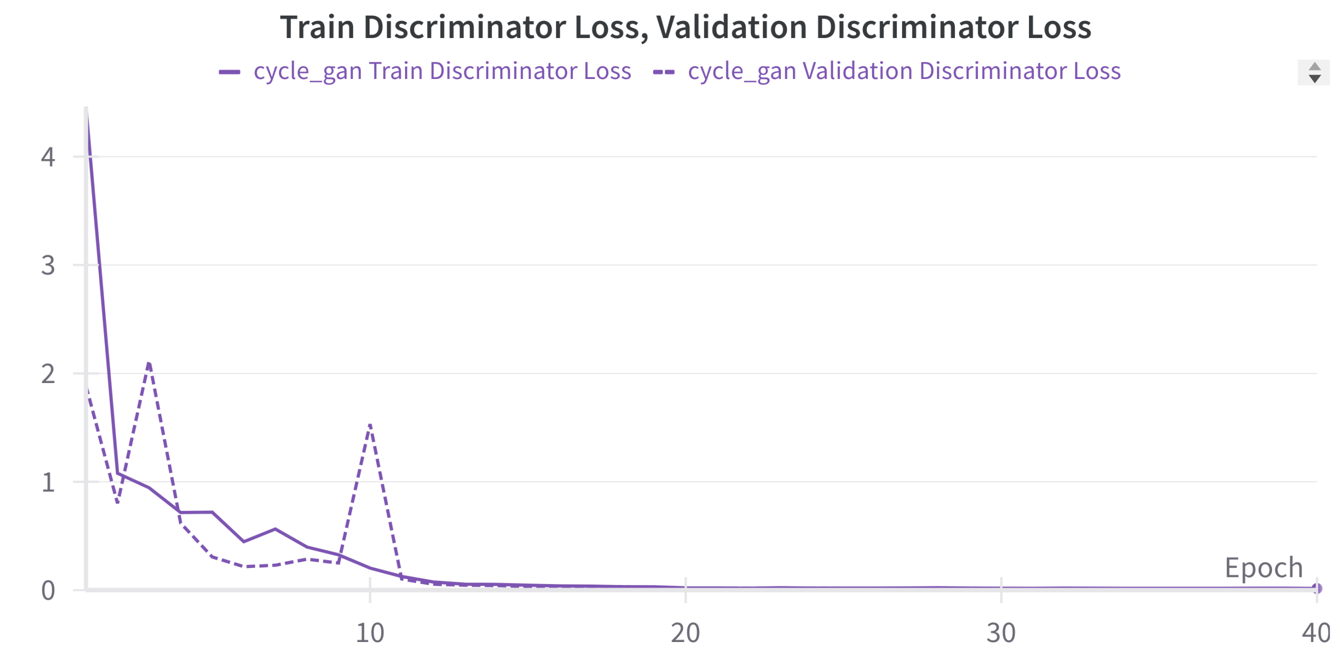


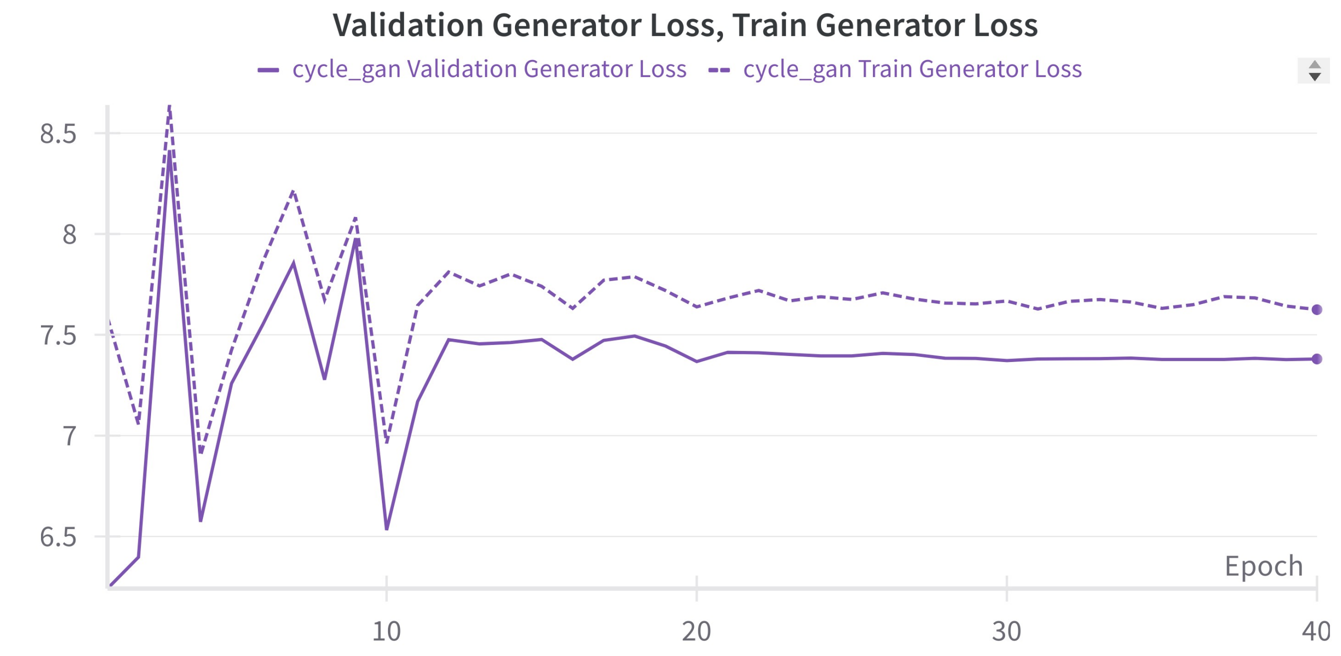
Results:



Results

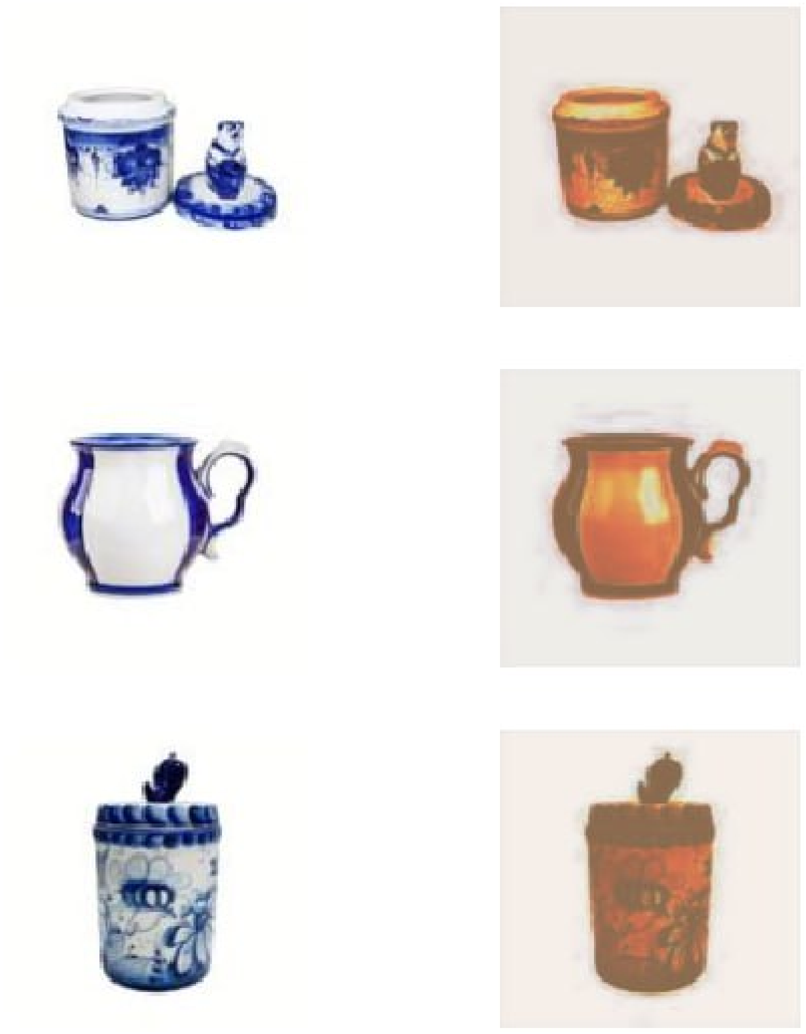
The best metrics:



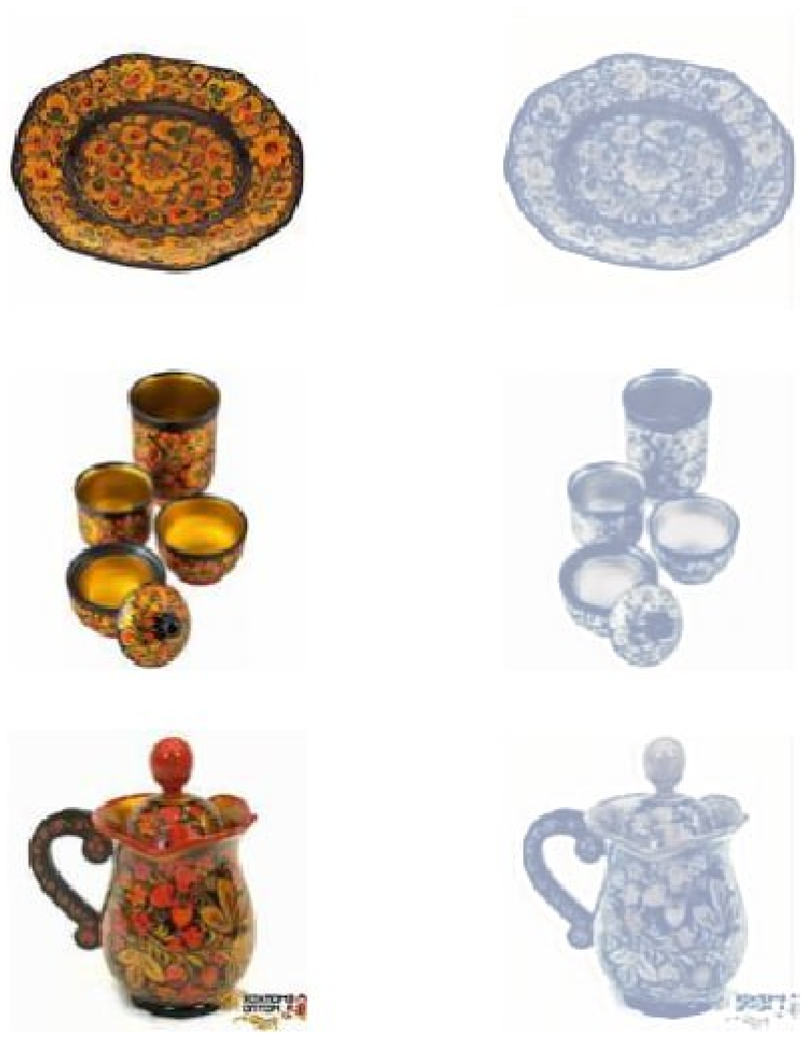


Results:

Gzhel to Khokhloma



Khokhloma to Gzhel



Insights

- Created the Gzhel2Khokhloma dataset from scratch by parsing several websites
- CycleGAN is hard to tailor
- CycleGAN often fails to handle texture/geometric transforms