

ПРОЕКТ ПО ООП «РАСКРАСКА»

РЕГАНОВА ЕЛИЗАВЕТА

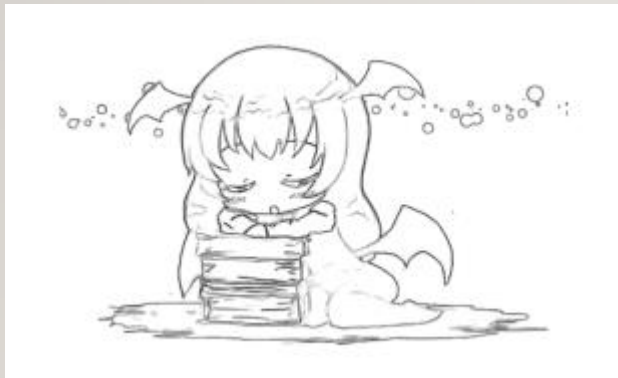
ФОМИЧЁВА СВЕТЛАНА

СИГОВА ЕЛИЗАВЕТА

Б06-806

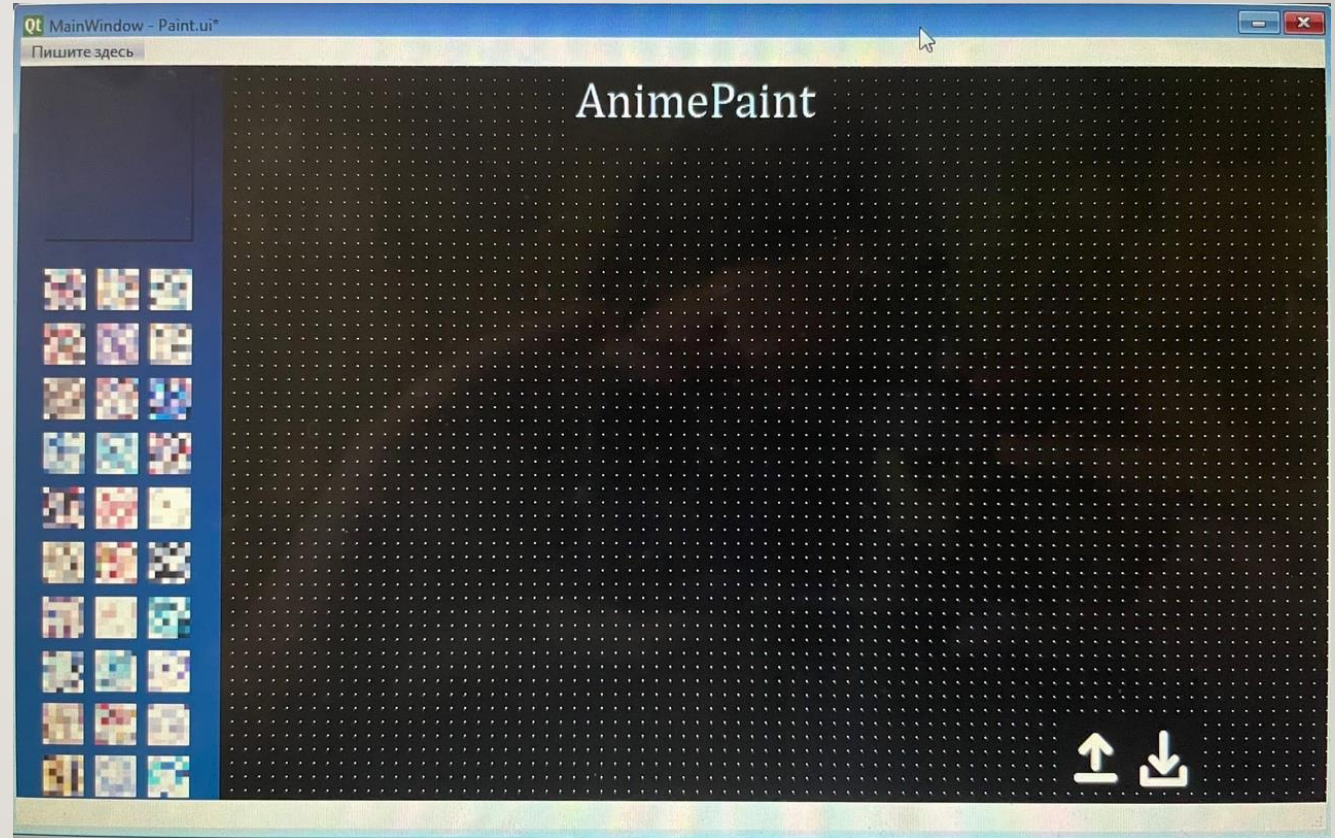
ПОСТАНОВКА ПРОБЛЕМЫ

Задача состоит в том, чтобы создать совместимое цветное аниме-изображение из заданного черно-белого аниме-эскиза с помощью генеративных состязательных сетей (GAN).



ИНТЕРФЕЙС

Выполнен с помощью PyQt6



ИСПОЛЬЗУЕМЫЕ МЕТОДЫ

Генеративные состязательные сети (GAN) - это подход к генеративному моделированию с использованием методов глубокого обучения, таких как сверточные нейронные сети (CNN).

- без учителя

- построенный на комбинации из двух нейронных сетей, одна из которых (сеть G) генерирует образцы, а другая (сеть D) старается отличить правильные («подлинные») образцы от неправильных

ГЕНЕРАТИВНЫЕ СОСТЯЗАТЕЛЬНЫЕ СЕТИ (GAN)

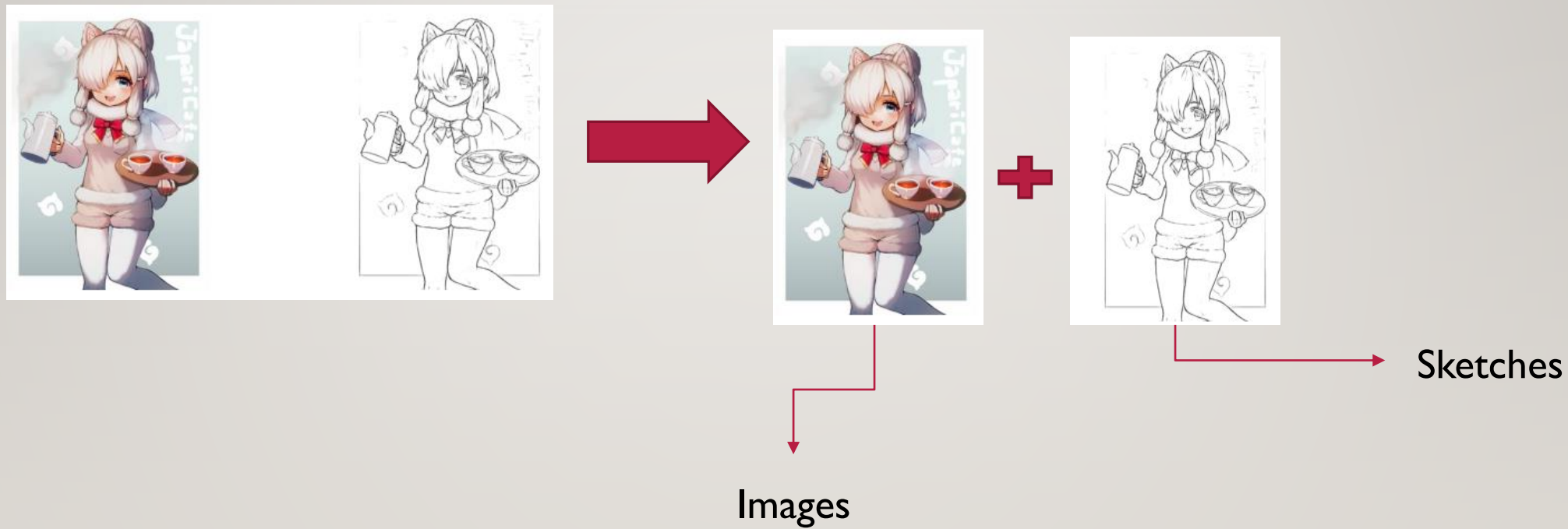
ГЕНЕРАТОР

- Генерирует новые экземпляры данных
- Цель генератора – генерировать такие изображения, которые будут пропущены дискриминатором

ДИСКРИМИНАТОР

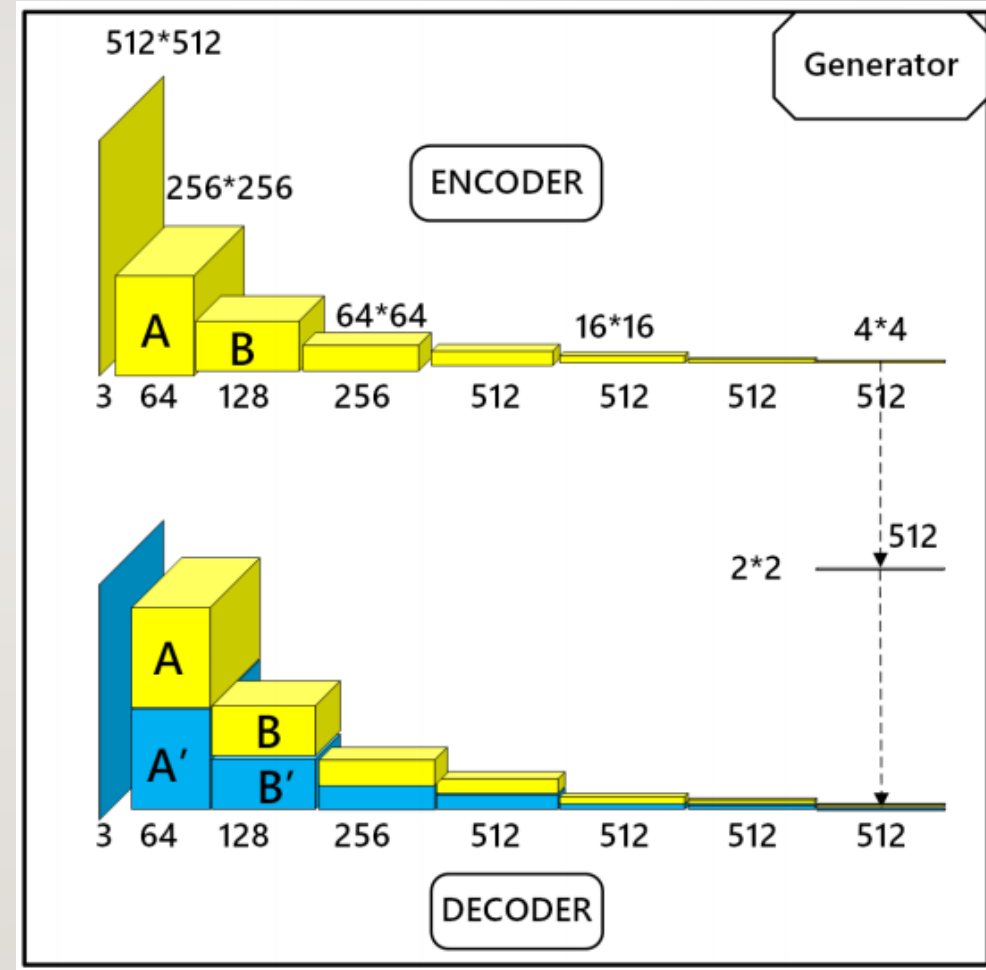
- Оценивает подлинность изображений
- Цель дискриминатора — определить, является ли изображение подлинным.

ПРЕДОБРАБОТКА ДАННЫХ

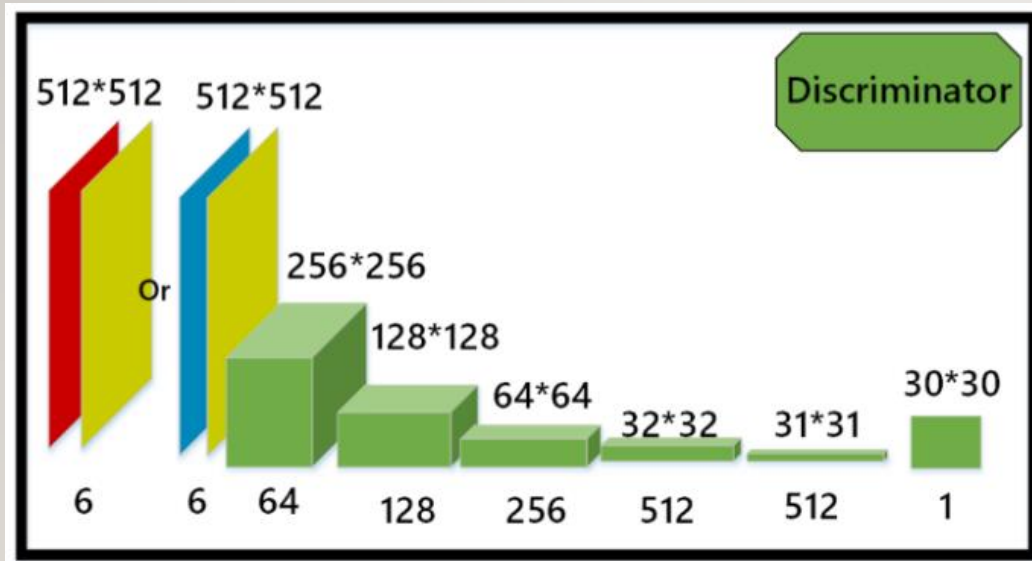


АРХИТЕКТУРА ГЕНЕРАТОРА

Архитектура генератора, который используется для эскиза для цветного перевода аниме, представляет собой своего рода «U-Net» .



АРХИТЕКТУРА ДИСКРИМИНАТОРА



Имеет только блоки кодировщика, и его цель - классифицировать, получено ли цветное изображение из реальных данных или созданное генератором.

Вход дискриминатора - это либо пара эскиза (желтый) и реального целевого изображения (красный), либо пара эскиза (желтый) и сгенерированного изображения (синий).

ПОТЕРИ В ГЕНЕРАТОРЕ И ДИСКРИМИНАТОРЕ

Поскольку наш Sketch2anime GAN является контролируемым обучением, мы будем использовать для этой цели условные GAN .

Функция потерь для общих условных GAN:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x, y \sim p_{data}(x, y)} [\log D(x, y)] + \mathbb{E}_{x \sim p_{data}(x), z \sim p_{data}(z)} [\log(1 - D(x, G(x, z)))]$$

« x » - это входной эскиз, « y » - это цель (цветное мультипликационное изображение), а « $G(x, z)$ » - сгенерированное цветное изображение.

Пока генератор пытается минимизировать потери, одновременно дискриминатор пытается максимизировать их, и в конечном итоге они достигают равновесия.