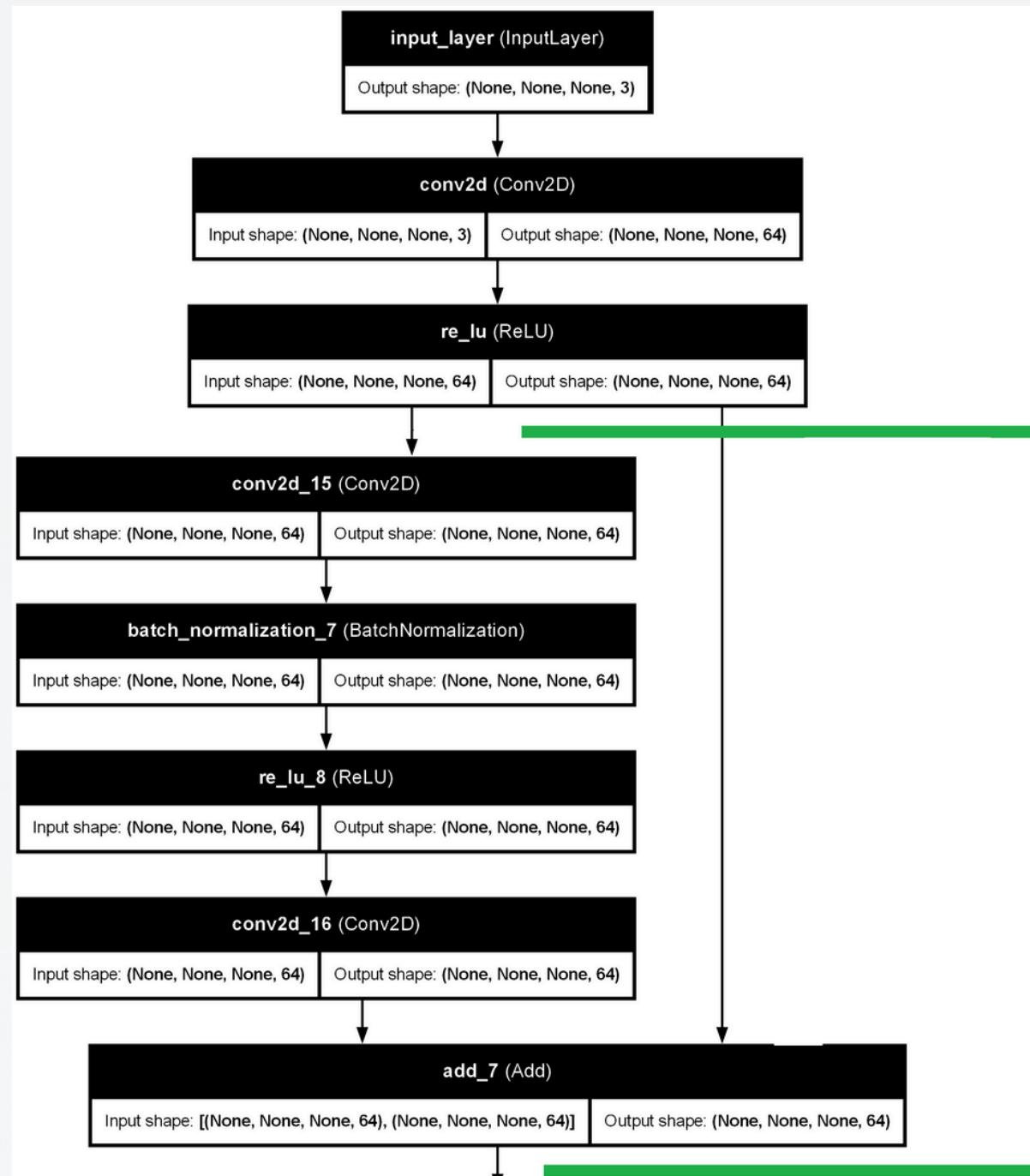


Image Super-Resolution using a Generative Adversarial Network (GAN)

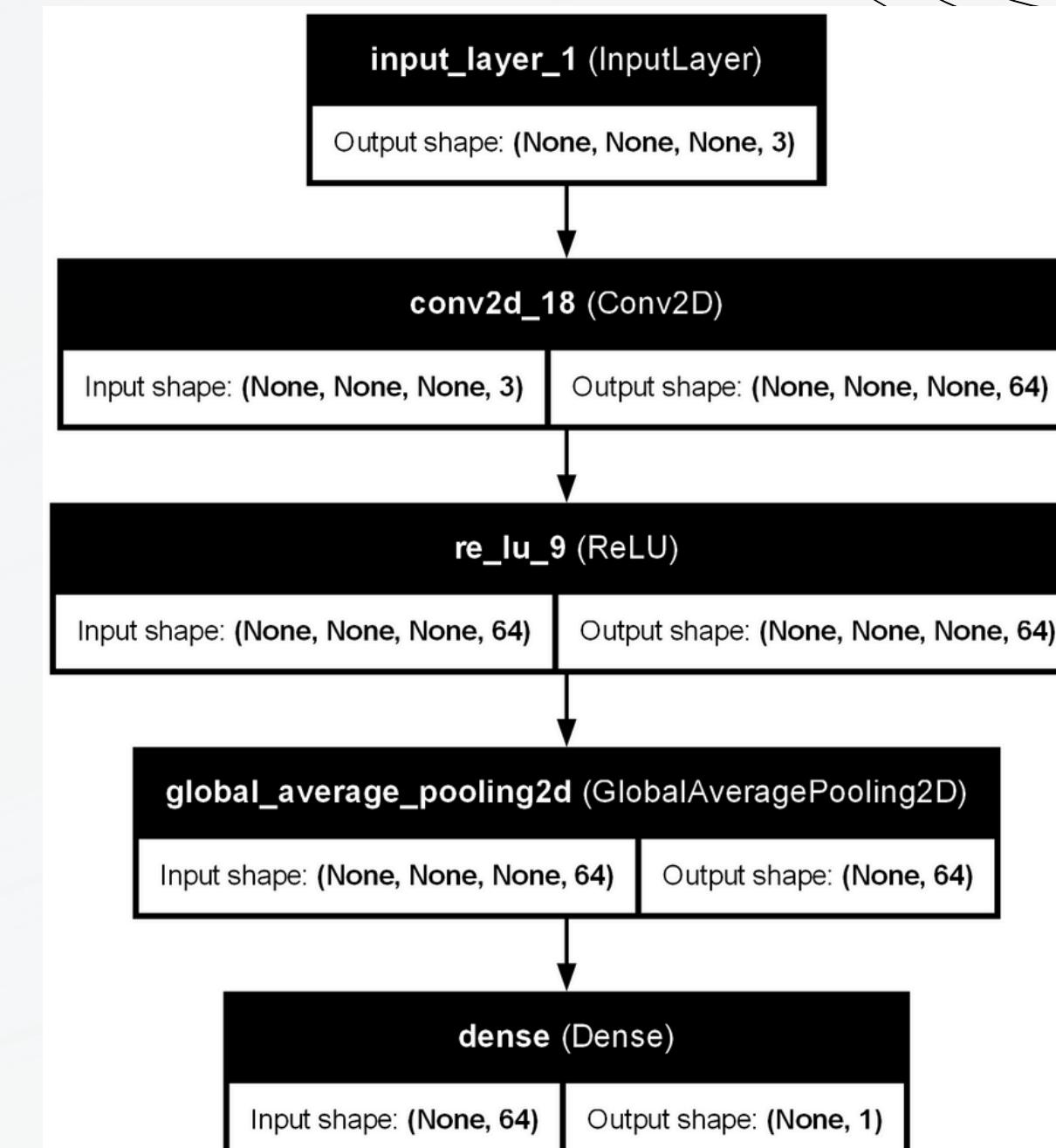
МОДЕЛЬ ГЕНЕРАТОРА



Total params: 596,419 (2.28 MB)
Trainable params: 595,395 (2.27 MB)
Non-trainable params: 1,024 (4.00 KB)

x8

МОДЕЛЬ ДИСКРИМИНАТОРА



Total params: 1,857 (7.25 KB)
Trainable params: 1,857 (7.25 KB)
Non-trainable params: 0 (0.00 B)

LOSS ГЕНЕРАТОРА

Комбінація трьох різних loss

- **Content loss** — це функція втрат, яка обчислює середню абсолютну різницю (MAE) або середню квадратичну різницю (MSE) між реальним зображенням і згенерованим. Вона оцінює базову схожість між пікселями двох зображень.
- **Perceptual loss** — це функція втрат, яка використовує попередньо натреновану модель для оцінки різниці між зображеннями на основі їх високорівневих ознак (features). Вона розраховує втрату семантичної схожості та контексту, враховуючи текстуру, структуру та загальну композицію.
- **Adversarial loss** — це функція втрат, яка визначає вплив дискримінатора на генератор. Результати дискримінатора для згенерованих зображень обчислюються через сигмоїдну крос-ентропію з логітами (`tf.nn.sigmoid_cross_entropy_with_logits`), оцінюючи, наскільки добре генератор зміг "обманути" дискримінатор.

LOSS ДИСКРИМІНАТОРА

Комбінація двох різних loss

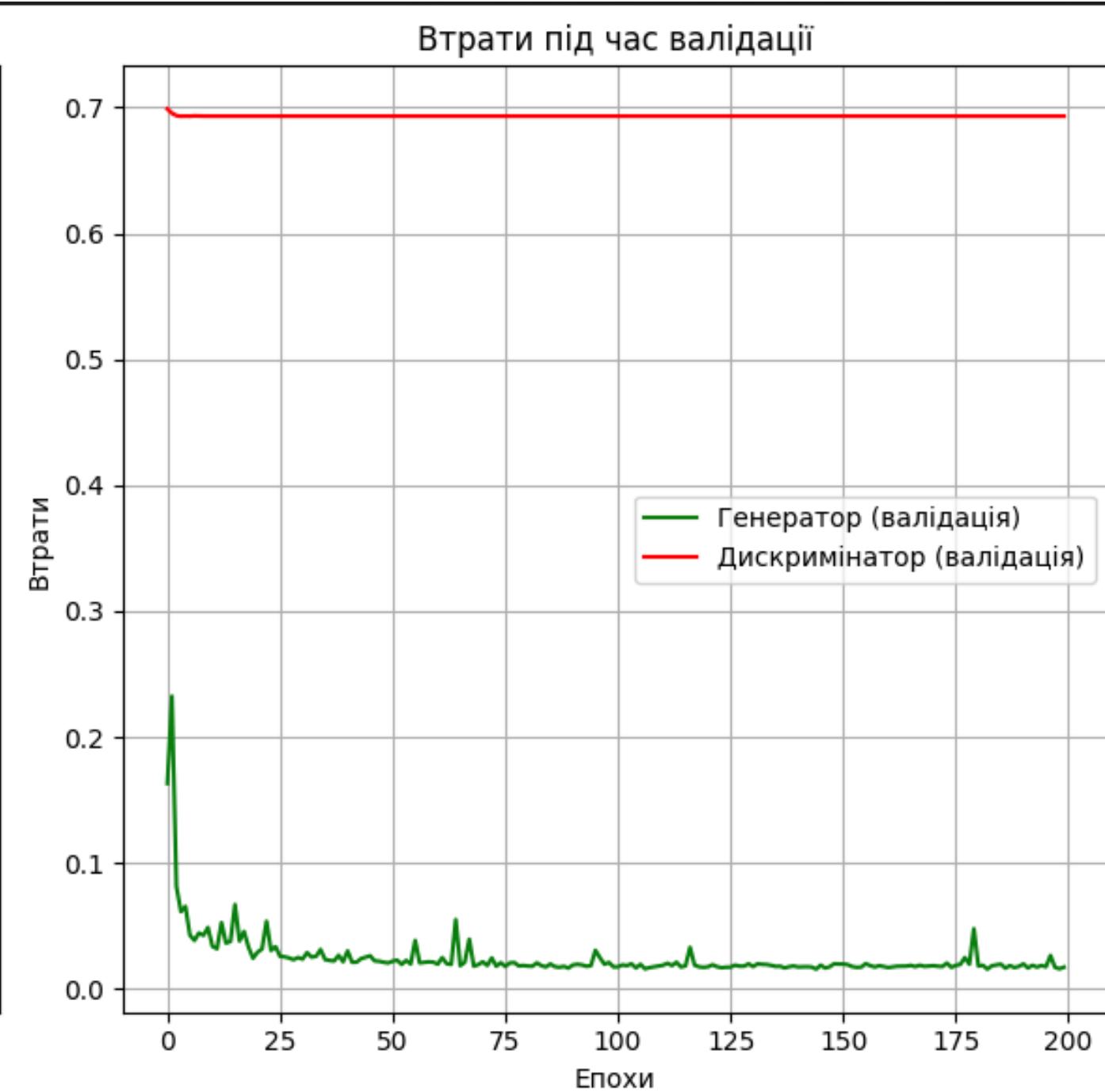
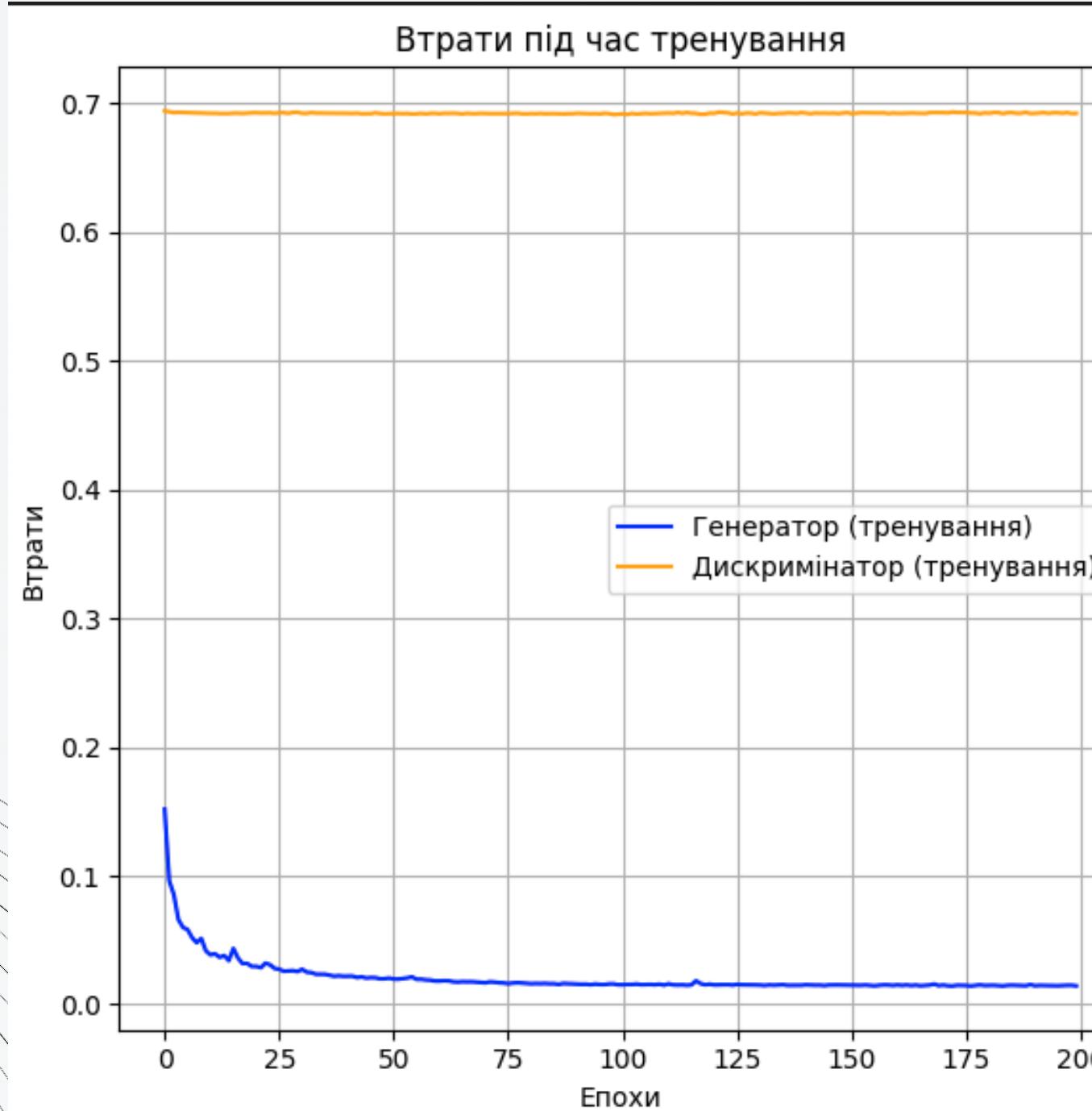
- **Оцінка дискримінатора реального фото:** Вихід дискримінатора для реальних зображень пропускається через функцію сигмоїдної крос-ентропії з логітами, де він порівнюється з матрицею одиниць. Це означає, що дискримінатор навчається максимально наблизити свій вихід до 1 для реальних зображень.
- **Оцінка дискримінатора згенерованого фото:** Вихід дискримінатора для згенерованих зображень пропускається через ту ж функцію сигмоїдної крос-ентропії з логітами, але тепер порівнюється з матрицею нулів. Це означає, що дискримінатор навчається максимально наблизити свій вихід до 0 для підроблених зображень.

ГРАФІКИ LOSS

Дискримінатор

$$\text{loss} = -\log(0.5) \approx 0.693$$

$$\text{sum_loss} = (\text{real_loss} + \text{fake_loss})/2$$



РЕЗУЛЬТАТИ

ORIGINAL



LOW RESOLUTION

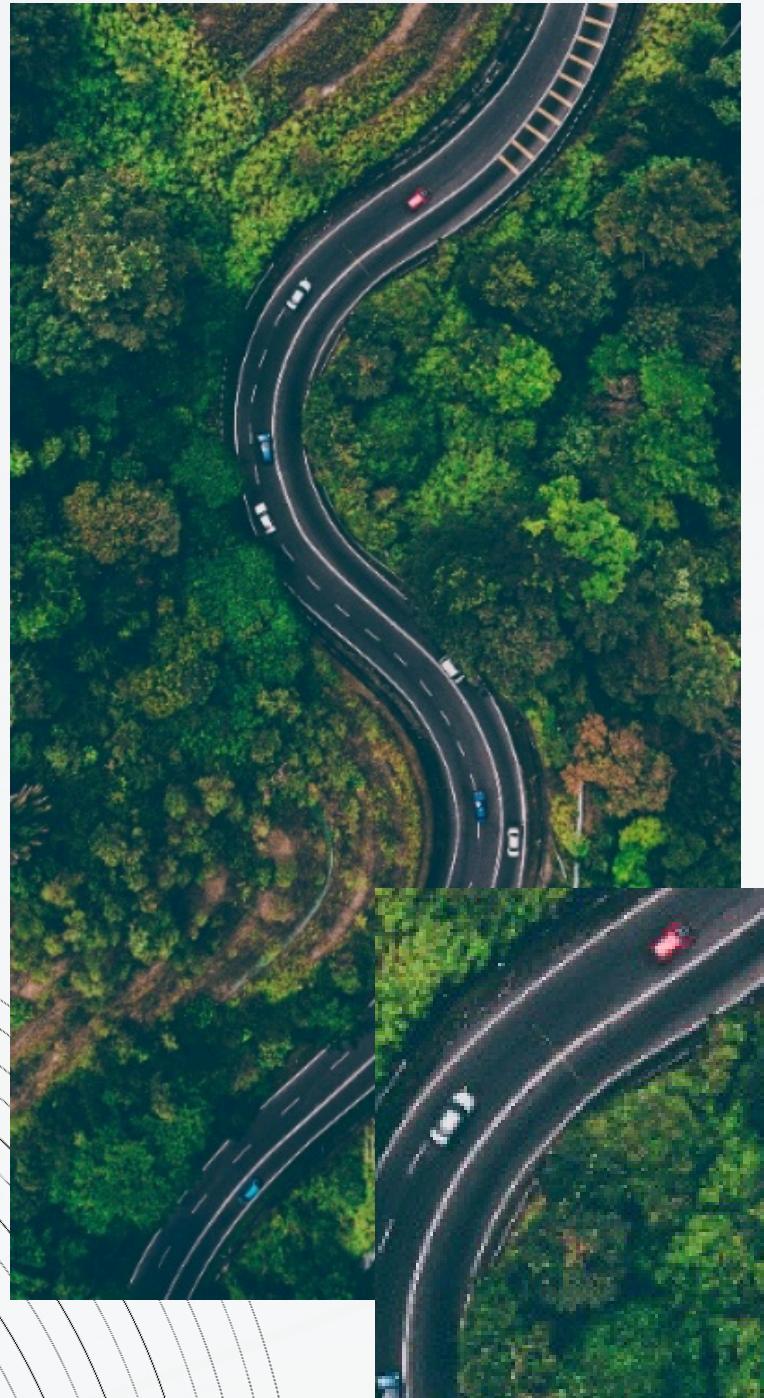


PREDICT MODEL

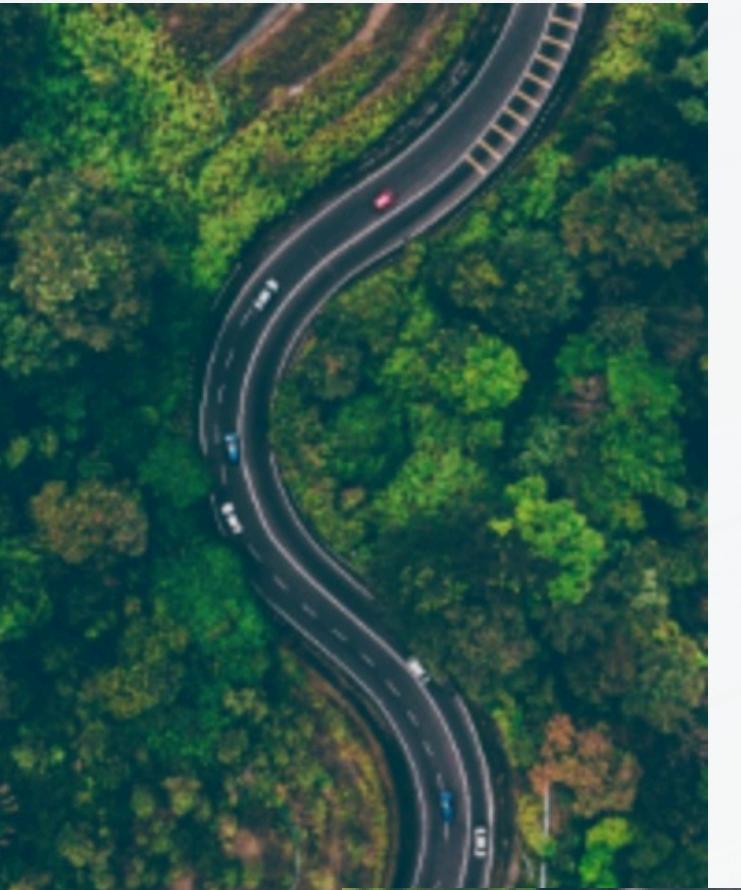


РЕЗУЛЬТАТИ

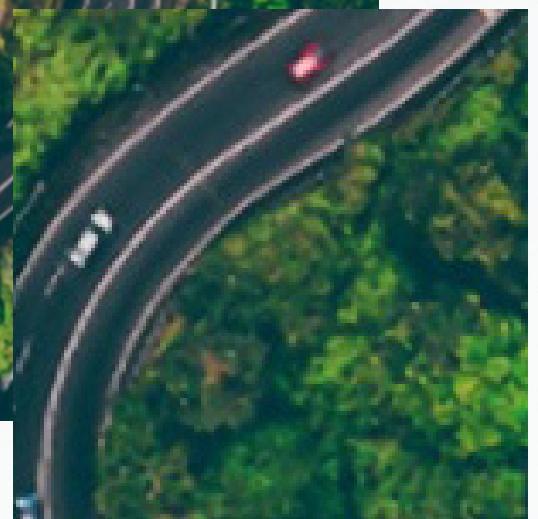
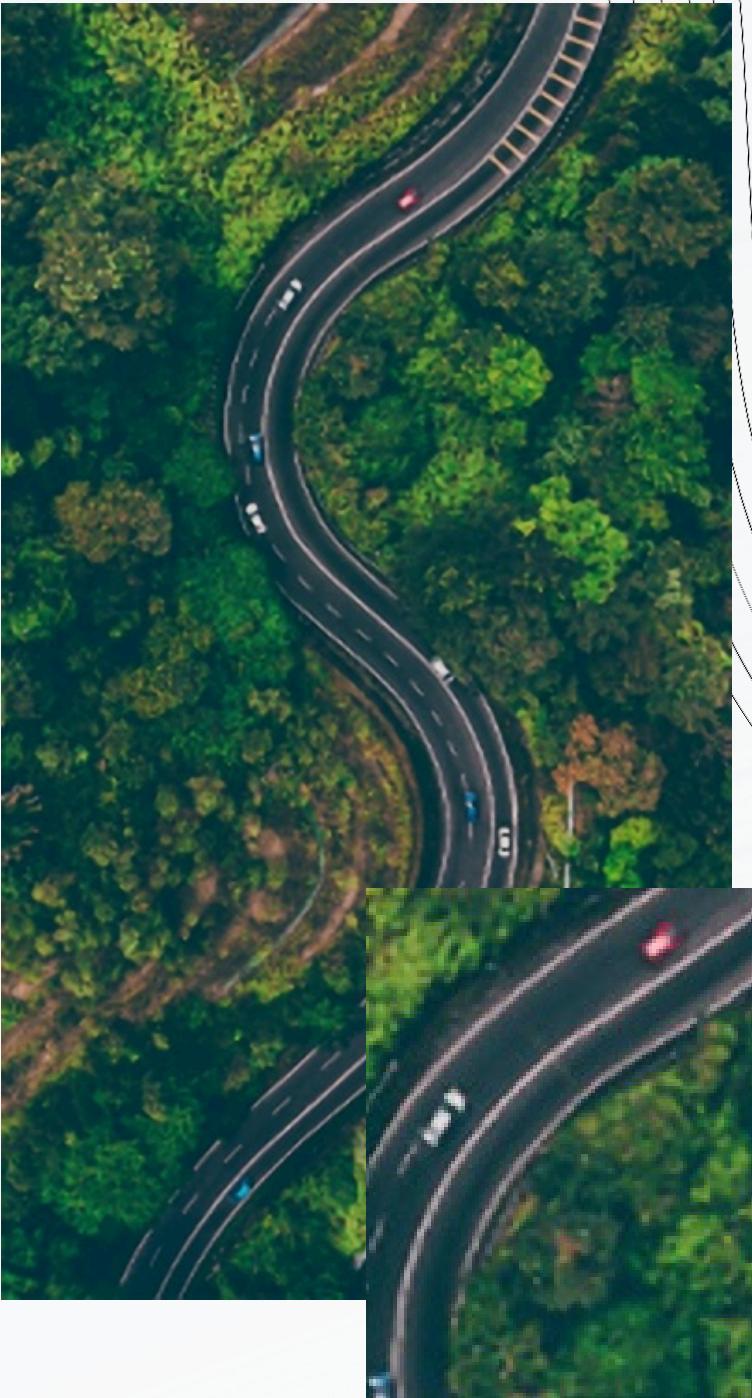
ORIGINAL



LOW RESOLUTION

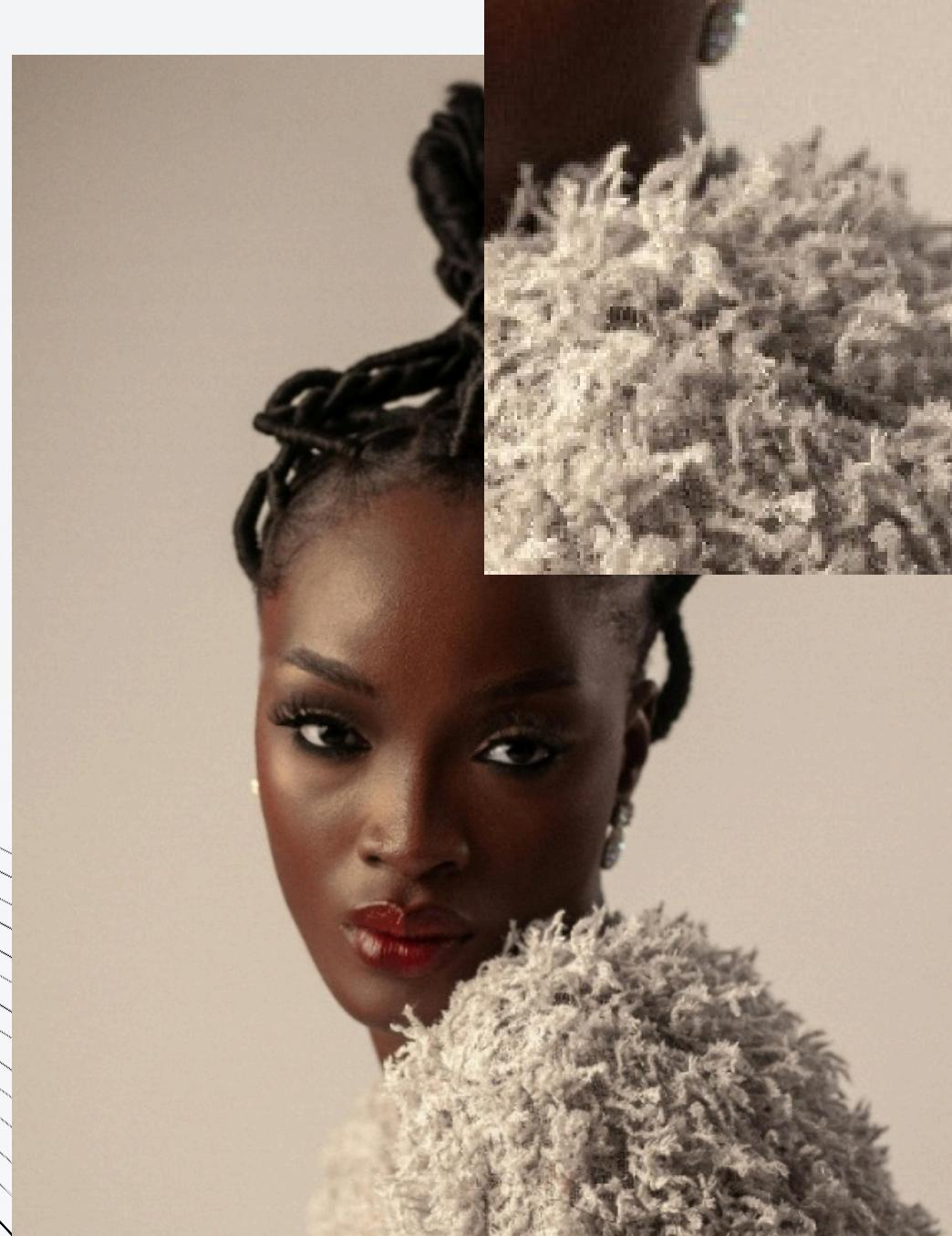


PREDICT MODEL

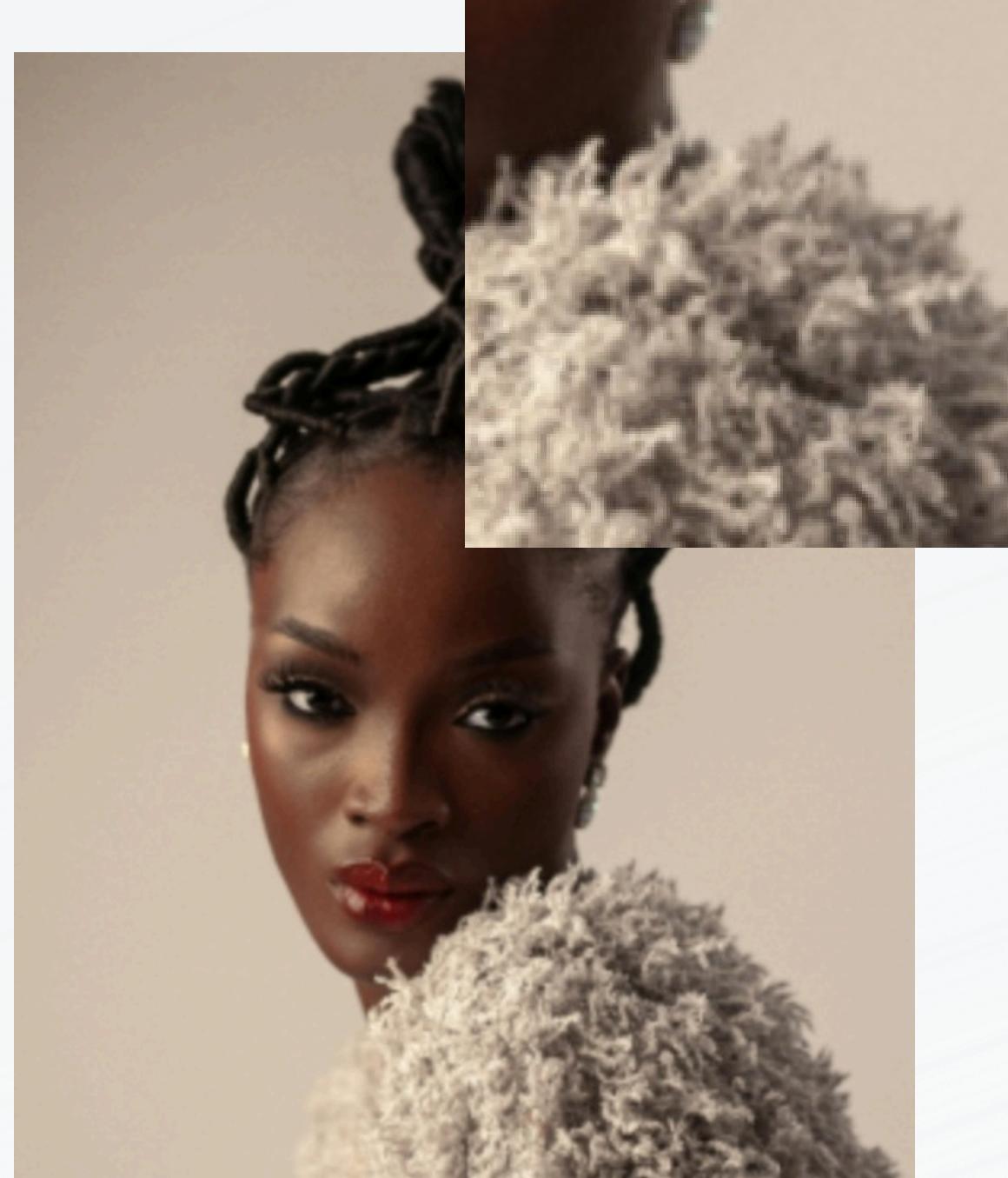


РЕЗУЛЬТАТИ

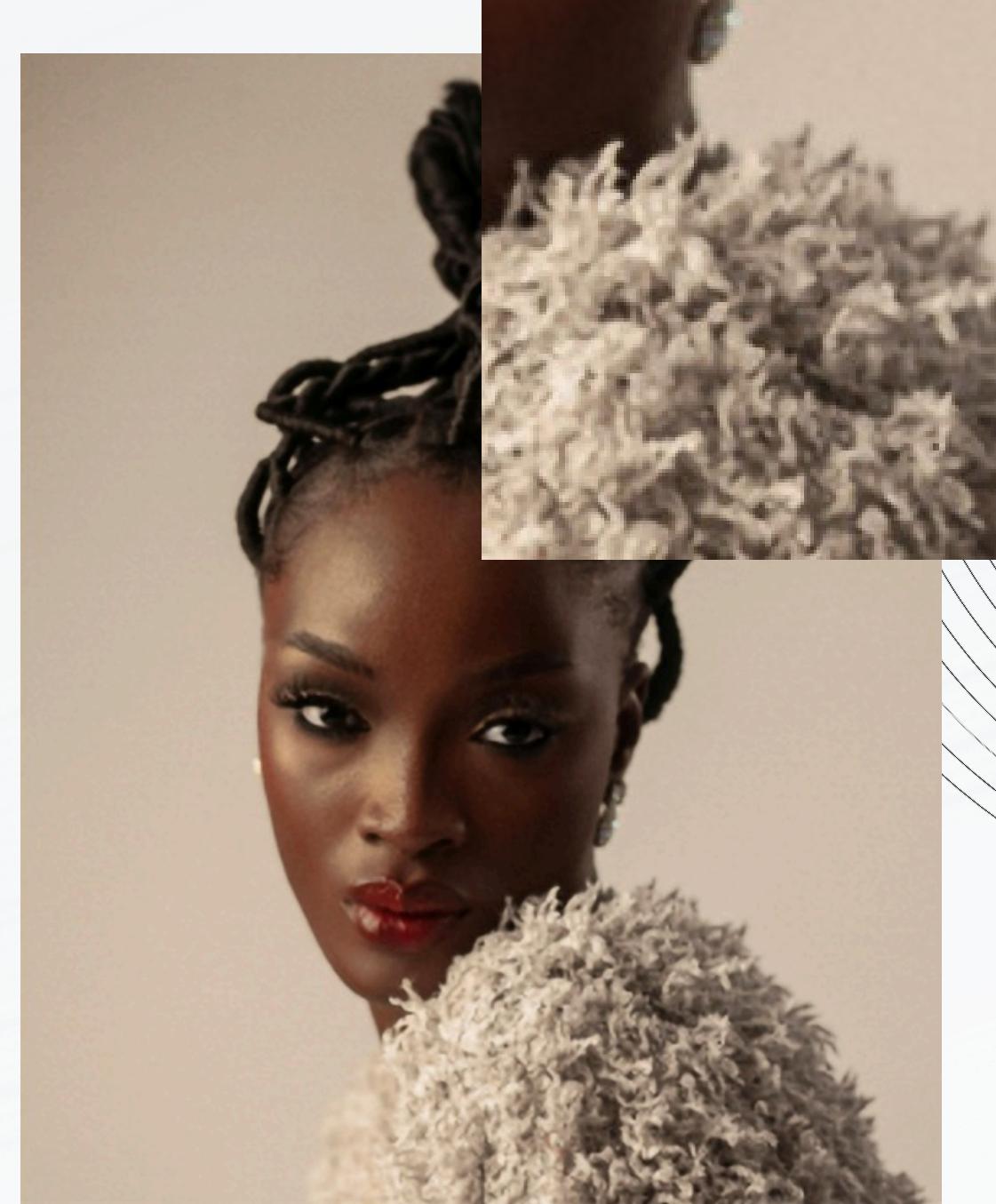
ORIGINAL



LOW RESOLUTION



PREDICT MODEL



РЕЗУЛЬТАТИ ВИКОРИСТАННЯ МОДЕЛІ ЗА ПРИЗНАЧЕННЯМ

ЗБІЛЬШЕНЕ В 4 РАЗИ ФОТО
(W*2, H*2)



PREDICT MODEL



РЕЗУЛЬТАТИ ВИКОРИСТАННЯ МОДЕЛІ ЗА ПРИЗНАЧЕННЯМ



МОЖЛИВІ ПОКРАЩЕННЯ

1. Структура моделей

- **Генератор:** По графікам видно, що модель стабільно навчалась, але не покращувала свої показники після 100 епохи. Тому потрібно переглянути можливість збільшення кількості Residual Blocks
- **Дискримінатор:** Після збільшення кількості Residual Blocks у генератора, важливо спостерігати щоб генератор не домінував над дискримінатором і навпаки. (також це можна вирішити змінюючи `lr` моделей)
- **Оптимізатори/активатори та вхід моделі:** Варто спробувати “погратися” з гіперпараметрами моделей і, можливо, змінити вхід/вихід моделі з діапазону від 0 до 1 до діапазону від -1 до 1

2. Показники Loss

- **Perceptual loss:** варто додати оцінку перцептуальних втрат для генератора
- По моїм спостереженням варто також додати оцінку для втрат кольорового балансу. Чи спостереження за гістограмами зображень

3. Збільшення датасету