



ММОВС23

Пищик Е.С.

Super Resolution в домене видеоигр

Команда состоит из двух человек:

- Пищик Евгений, разработчик сервиса + ML части,
[@Evgenii_Pishchik](#);
- Рогачев Александр, куратор,
[@airogachev](#).

[Ссылка на проект на GitHub](#).



Super Resolution (SR) - задача преобразования исходного изображения низкого разрешения и качества в изображение высокого разрешения и качества.

Low Resolution 270p

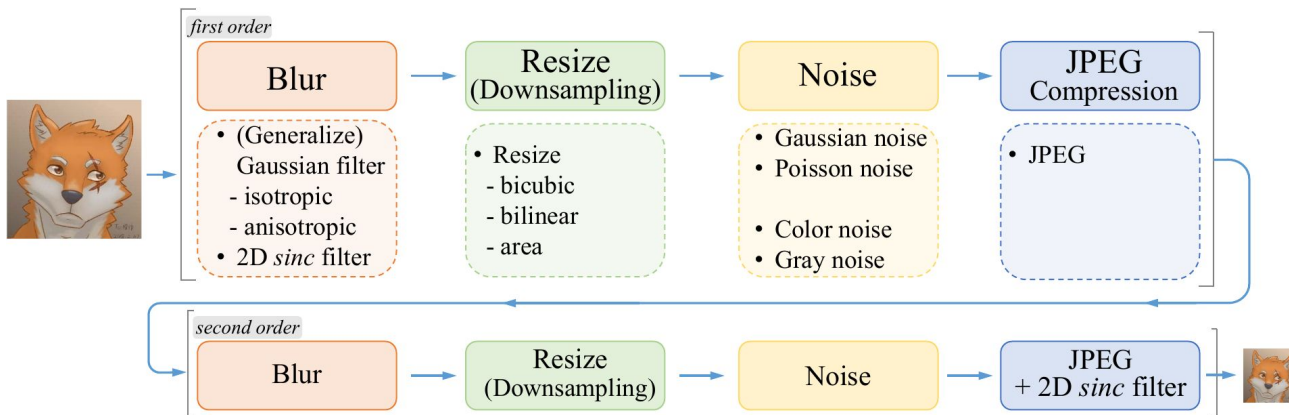
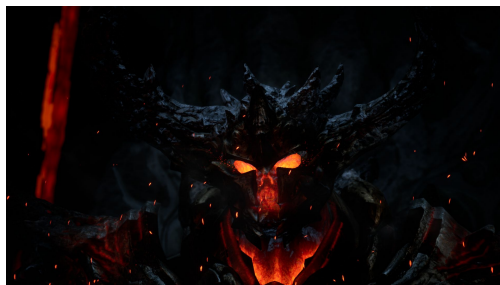


High Resolution 1080p

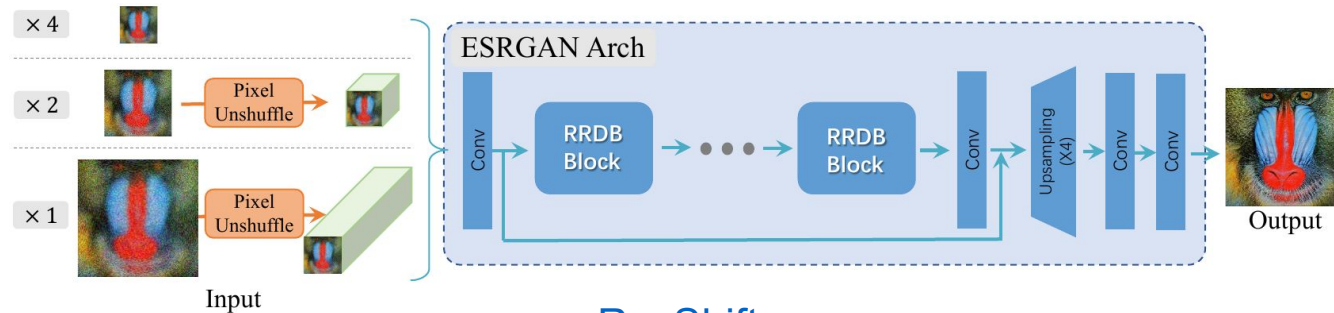


Данные в проекте собирались двумя способами:

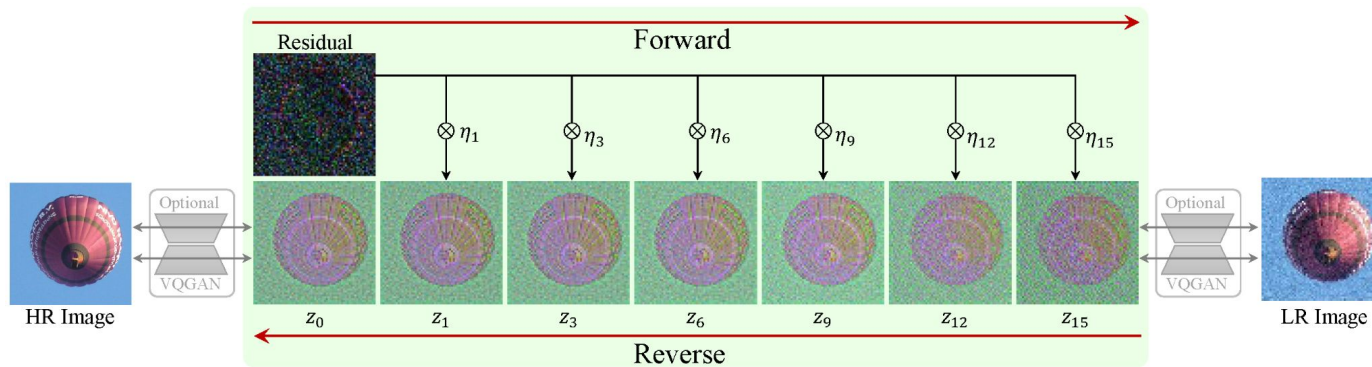
- Рендер на игровом движке [Unreal Engine](#) (UE);
- Рендер на игровых движках (Unreal Engine, [Source2](#)) в 1080p и ухудшение качества со снижением разрешения алгоритмом из статьи [Real-ESRGAN](#) до 270p, 360p, 540p.



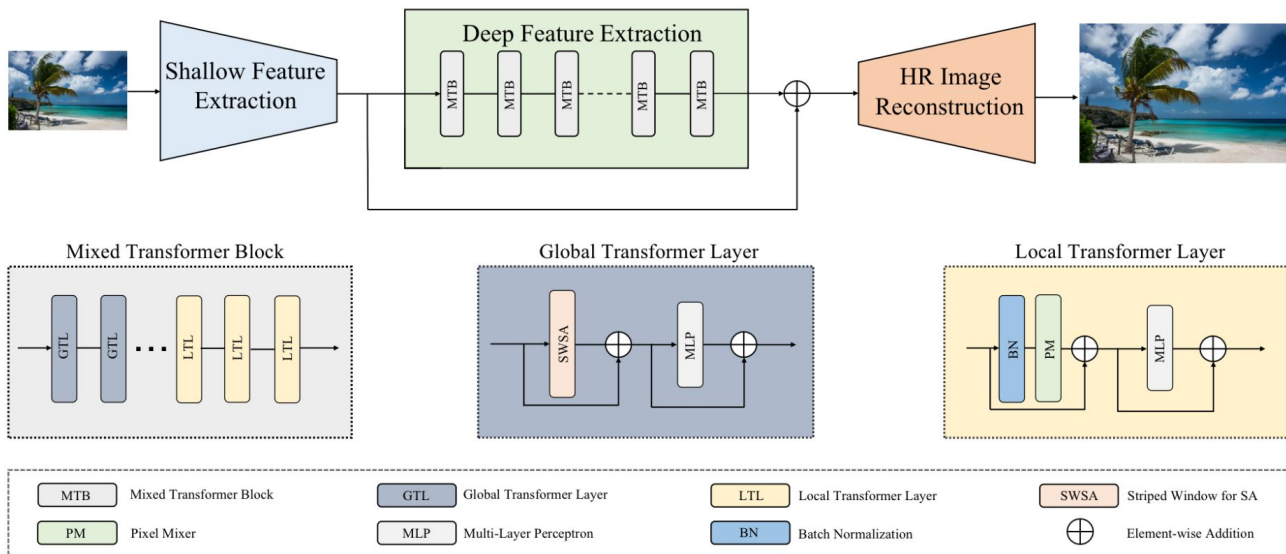
Real-ESRGAN



ResShift



EMT





Models 7



↑↓ Sort: Recently updated

epishchik/realer-animevideov3 private

Feature Extraction • Updated 2 days ago • ↓ 2

epishchik/realer-general-x4v3 private

Feature Extraction • Updated 2 days ago • ↓ 2

epishchik/RealESRGAN_x4plus_anime_6B private

Feature Extraction • Updated 2 days ago • ↓ 1

epishchik/GameEngine_RealESRGAN_x4plus private

Feature Extraction • Updated 2 days ago • ↓ 3

epishchik/RealESRNet_x4plus private

Feature Extraction • Updated 2 days ago • ↓ 1

epishchik/RealESRGAN_x2plus private

Feature Extraction • Updated 2 days ago • ↓ 2

epishchik/RealESRGAN_x4plus private

Feature Extraction • Updated 2 days ago • ↓ 4

Datasets 1



epishchik/srgb private

Updated 2 days ago • ↓ 12



Hugging Face

PSNR is most easily defined via the [mean squared error](#) (MSE). Given a noise-free $m \times n$ monochrome image I and its noisy approximation K , MSE is defined as

$$MSE = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2.$$

The PSNR (in [dB](#)) is defined as

$$\begin{aligned} PSNR &= 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \\ &= 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE). \end{aligned}$$

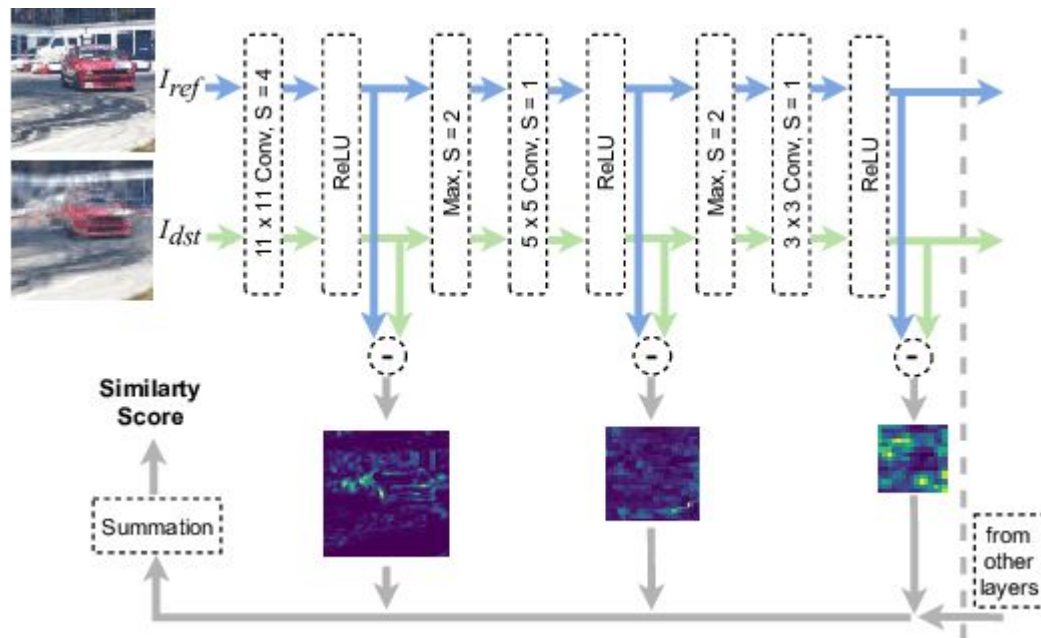
Here, MAX_I is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255. More generally, when samples are represented using linear [PCM](#) with B bits per sample, MAX_I is $2^B - 1$.

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

with:

- μ_x the pixel sample mean of x ;
- μ_y the pixel sample mean of y ;
- σ_x^2 the variance of x ;
- σ_y^2 the variance of y ;
- σ_{xy} the covariance of x and y ;
- $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ two variables to stabilize the division with weak denominator;
- L the dynamic range of the pixel-values (typically this is $2^{\#bits \text{ per pixel}} - 1$);
- $k_1 = 0.01$ and $k_2 = 0.03$ by default.

LPIPS - Learned Perceptual Image Patch Similarity





| RealESRGAN_x4plus | PSNR↑ | SSIM↑ | LPIPS↑ | <u>HaarPSI</u> ↑ | <u>BRISQUE</u> ↓ |
|-------------------|---------|--------|--------|------------------|------------------|
| Pretrained | 23.5409 | 0.7992 | 0.3924 | 0.5158 | 26.5565 |
| Finetuned | 24.3873 | 0.8348 | 0.3009 | 0.5625 | 32.3198 |



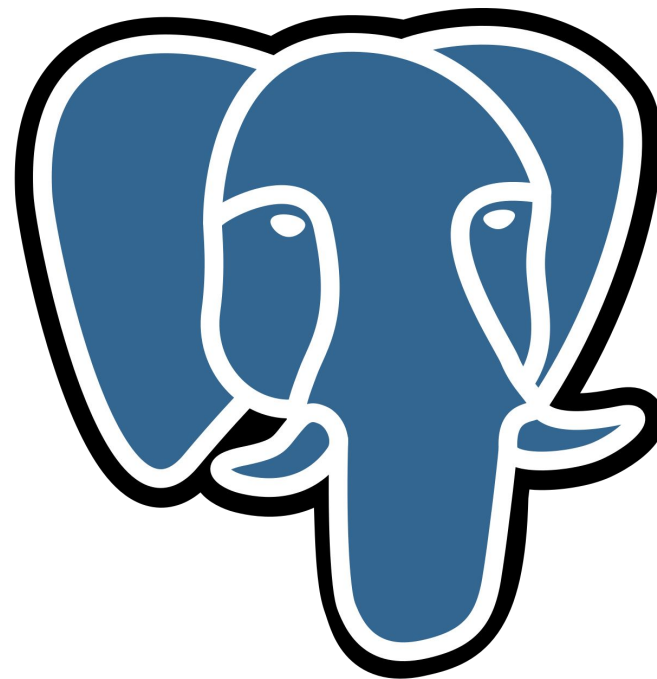
| EMT_x4 | PSNR↑ | SSIM↑ | LPIPS↑ | <u>HaarPSI</u> ↑ | <u>BRISQUE</u> ↓ |
|------------|---------|---------|---------|------------------|------------------|
| Pretrained | 24.5443 | 0.8231 | 0.3889 | 0.5417 | 61.9432 |
| Finetuned | not yet | not yet | not yet | not yet | not yet |

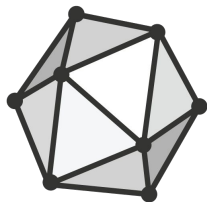


| ResShift_RealSRx4 | PSNR↑ | SSIM↑ | LPIPS↑ | HaarPSI ↑ | BRISQUE ↓ |
|-------------------|---------|---------|---------|---------------------------|---------------------------|
| Pretrained | 23.0368 | 0.7992 | 0.4829 | 0.5225 | 13.0324 |
| Finetuned | not yet | not yet | not yet | not yet | not yet |

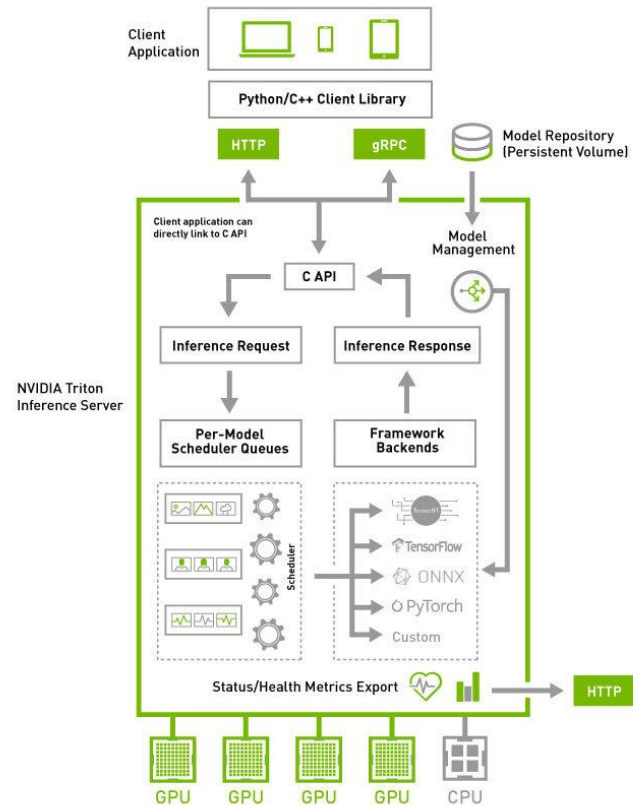
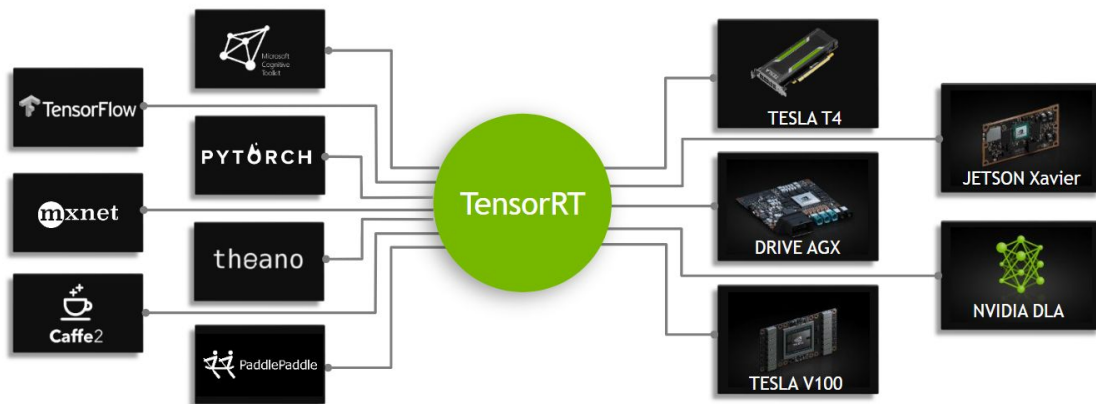


mlflow™





ONNX



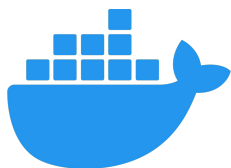


Средние показатели inference на валидации для перевода из 270p в 1080p
покадрово для модели RealESRGAN_x4plus на видеокарте NVIDIA GeForce GTX 1660
Super 6 Gb.

| TYPE | SPEED,sec | INCREASE, % |
|-----------------|-----------|-------------|
| PyTorch CUDA | 1.070 | - |
| ONNX CUDA | 1.072 | 0 |
| ONNX TensorRT | 0.947 | 11 |
| Triton TensorRT | 0.563 | 47 |



Streamlit



docker

FastAPI

0.1.0

OAS 3.1

[/openapi.json](#)

default ^

GET /info Info

POST /configure_model/name Configure Model

POST /configure_model/file Configure Model File

POST /upscale/example Upscale Example

POST /upscale/file Upscale

- Новые модели;
- Обучение моделей на собранном датасете;
- Оптимизации моделей;
- Оформление wiki;
- Публикация датасета на Kaggle и HuggingFace;
- Публикация моделей со ссылками на авторов на HuggingFace;
- Документирование различных частей проекта (например добавление описания датасета и моделей на huggingface, добавление help к аргументам парсеров в скриптах и т.д.).



Doom 1993, RealESRGAN_x4plus, 270p -> 1080p

