

Факультет компьютерных наук  
Департамент программной инженерии  
Выпускная квалификационная работа

# Повышение разрешения изображений с использованием конкурирующих нейронных сетей

## Image Super-Resolution Using Generative Adversarial Networks

---

Выполнил студент группы БПИ131  
образовательной программы 09.03.04 «Программная инженерия»

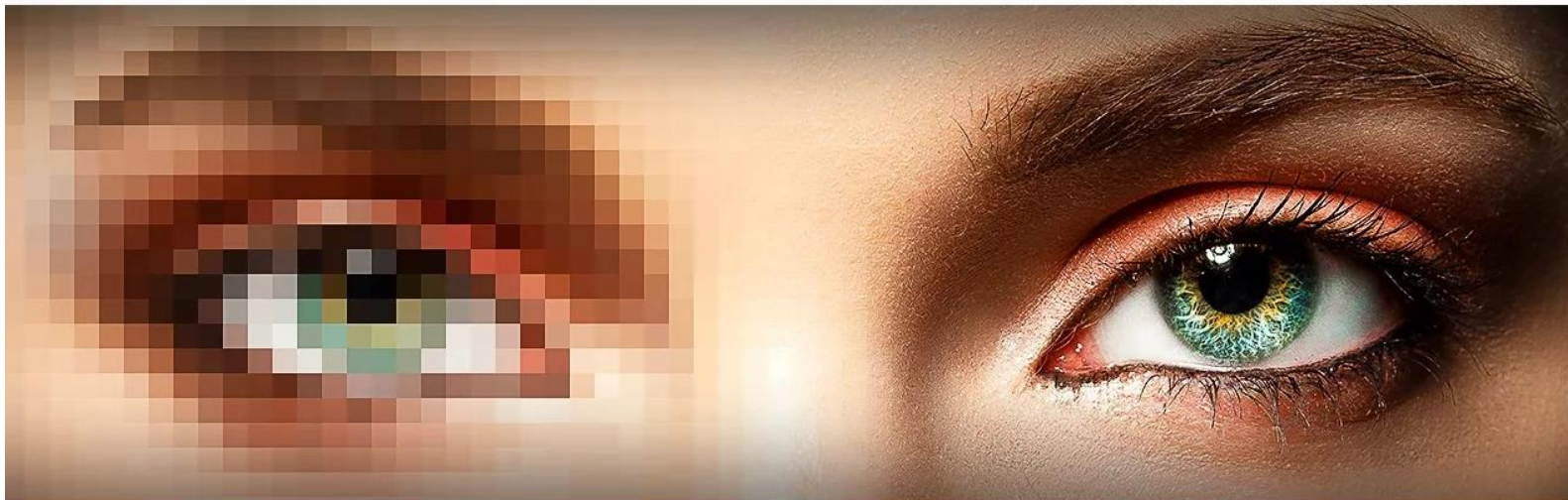
**Ложков Антон Геннадиевич**

Научный руководитель:  
доцент департамента программной инженерии, к.т.н.  
**Ахметсафина Римма Закиевна**

# Постановка задачи

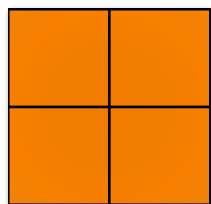
---

**LOW-RES** >> **HIGH-RES**

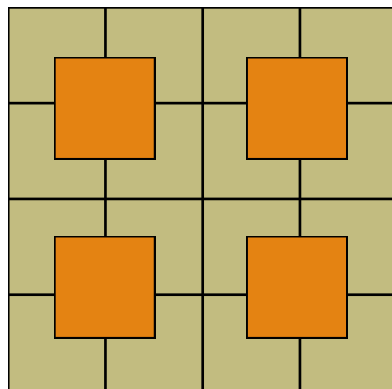


- Увеличение размера растрового изображения
- Сохранение естественного вида и четкости
- 4-кратное увеличение с использованием только одного исходного изображения

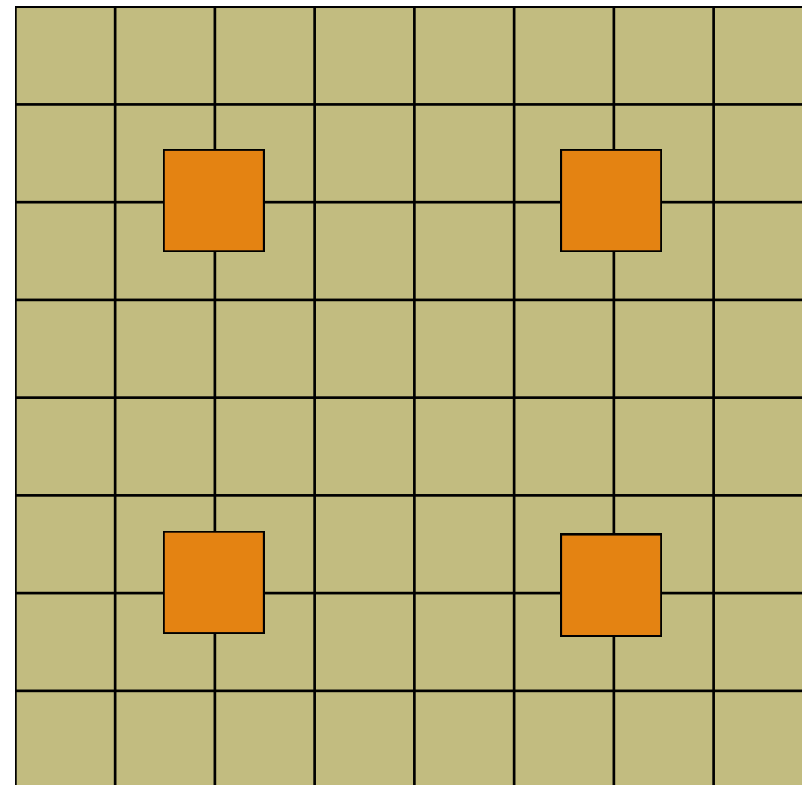
# Постановка задачи



1x



2x



4x

# Актуальность работы

---

- Увеличение разрешения изображений и видео, загруженных пользователями социальных сетей и медиахранилищ;
- Восстановление архивных материалов и трудночитаемого текста, устранение артефактов сжатия;
- Лучшие по качеству результаты достигаются с использованием сверточных нейронных сетей (Deep Convolutional Networks);

# Цель и задачи работы

---

**Цель:** разработка веб-сервиса для повышения разрешения изображений.

**Задачи:**

1. Изучить существующие алгоритмы и подходы к увеличению разрешения изображений;
2. Изучить возможность применения конкурирующих сверточных нейронных сетей для решения представленной задачи;
3. Разработать и реализовать архитектуру нейронной сети;
4. Разработать пользовательский веб-интерфейс для демонстрации возможностей подхода;
5. Экспериментально сравнить качество увеличенных изображений, полученных предложенным методом и аналогами. Разработать критерии сравнения качества изображений;
6. Разработать техническую документацию.

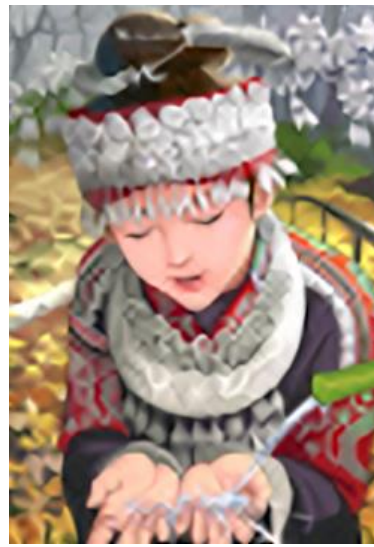
# Сравнение аналогов



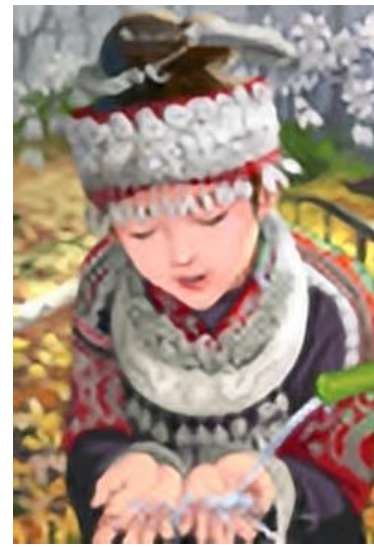
Бикубическая  
интерполяция



Sparse coding  
Gu et al. [1]



Self-exemplars  
Huang et al. [2]



SRCNN  
Dong et al. [3]



ВКР



Оригинал

Преимущества реализации: наиболее реалистичные изображения среди аналогов, простота интерфейса обучения модели, распределенность системы, пользовательский веб-интерфейс.

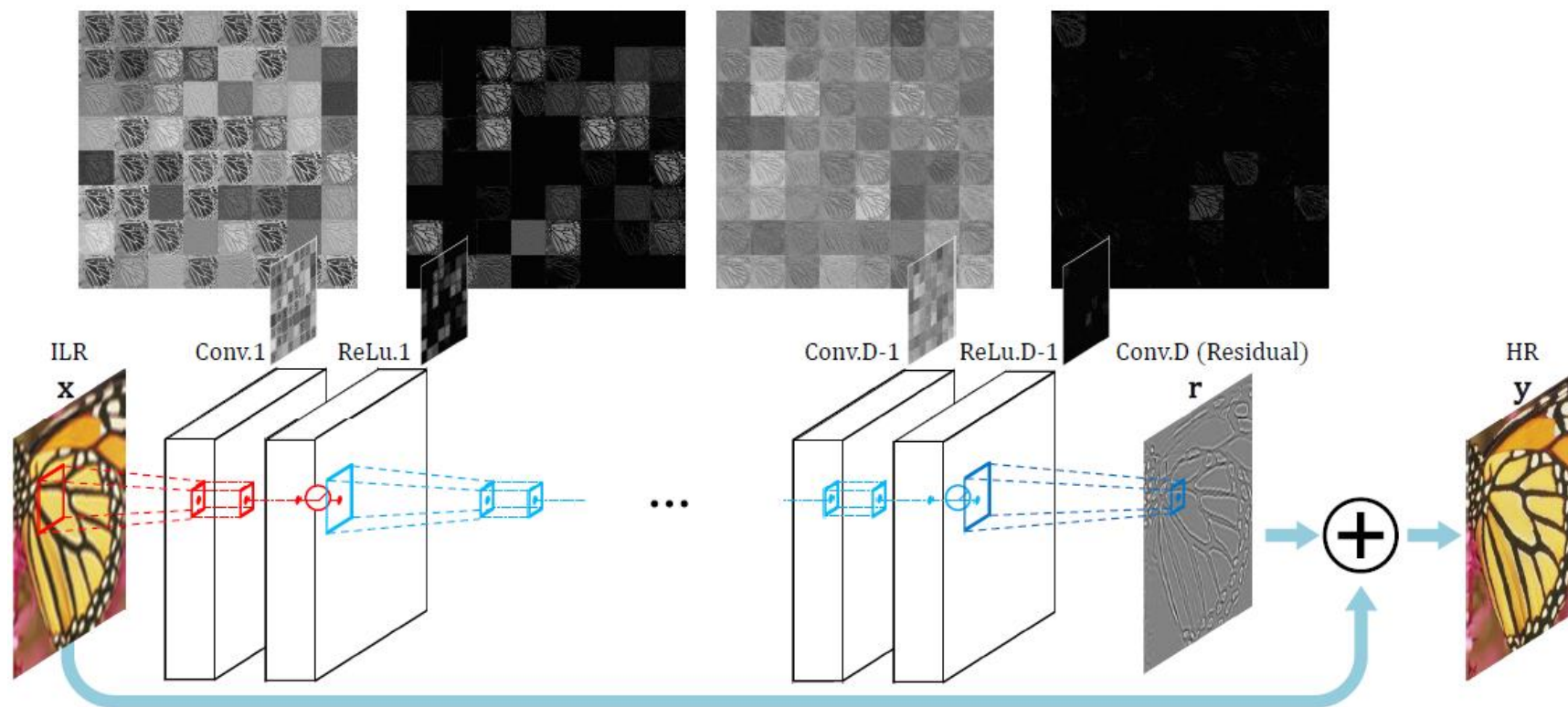


# Сверточные нейронные сети

Серия сверточных слоев и нелинейных функций, преобразующих изображение низкого разрешения.

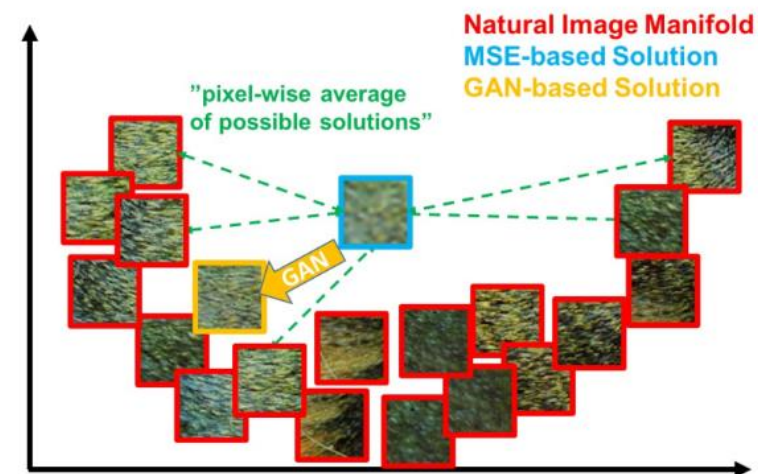
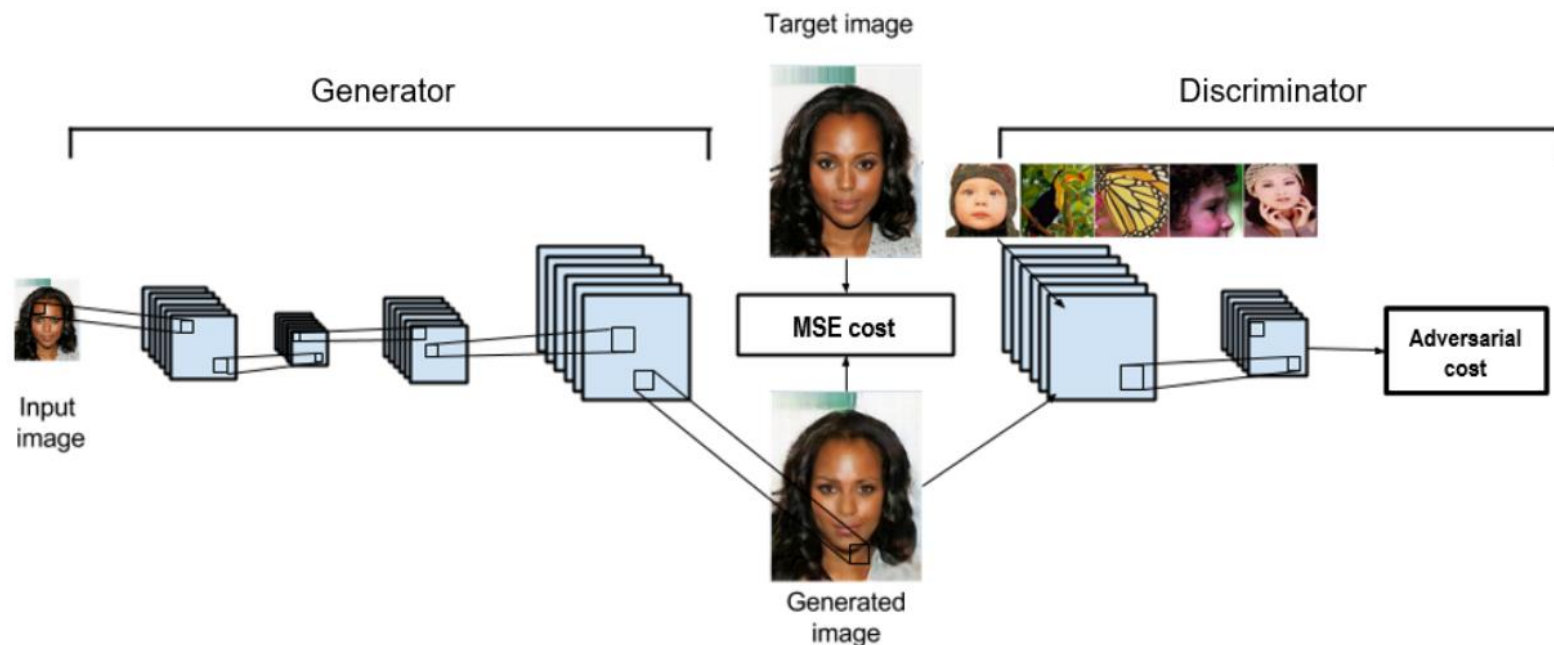
При обучении минимизируется попиксельная средняя квадратическая функция ошибки:

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2$$



[C. Dong et al., "Image super-resolution using deep convolutional networks," 2016]  
[J. Kim et al., "Accurate image super-resolution using very deep convolutional networks," 2016]

# Конкурирующие нейронные сети



При обучении минимизируется комбинированная функция ошибки:

$$I^{SR} = \underbrace{I_X^{SR}}_{\text{ошибка генератора (MSE)}} + 10^{-3} \underbrace{I_{Gen}^{SR}}_{\text{ошибка дискриминатора}}$$

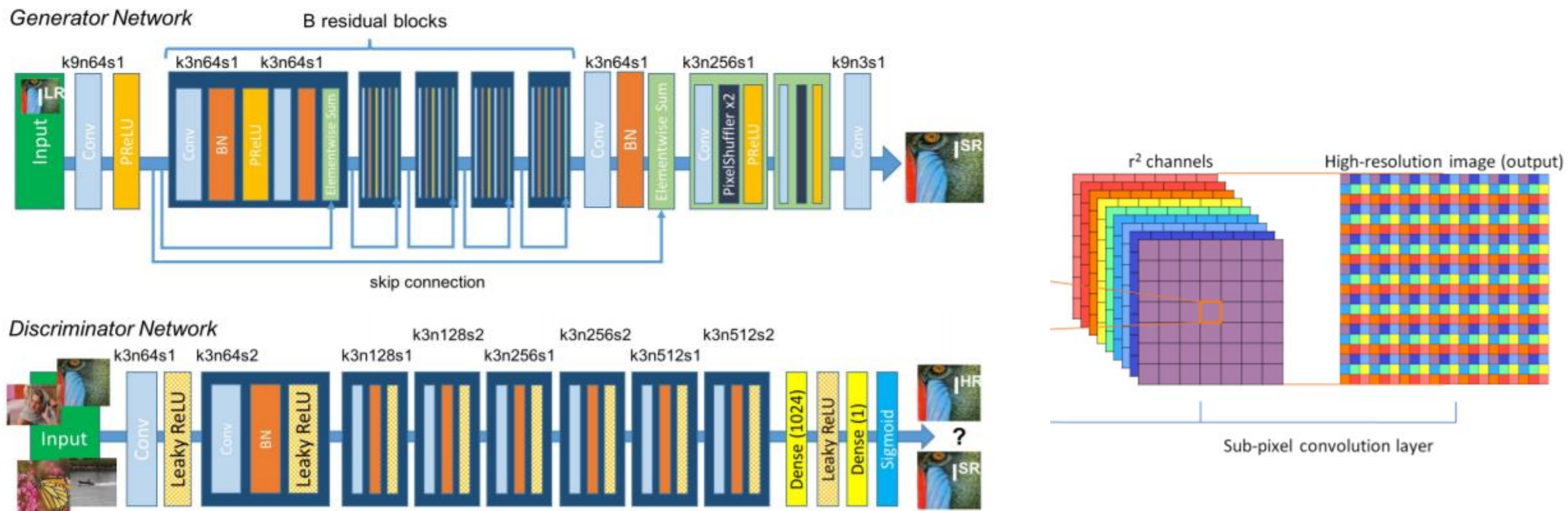
ошибка генератора (MSE)

ошибка дискриминатора

[J. Johnson et al., "Perceptual losses for real-time style transfer and super-resolution," 2016]



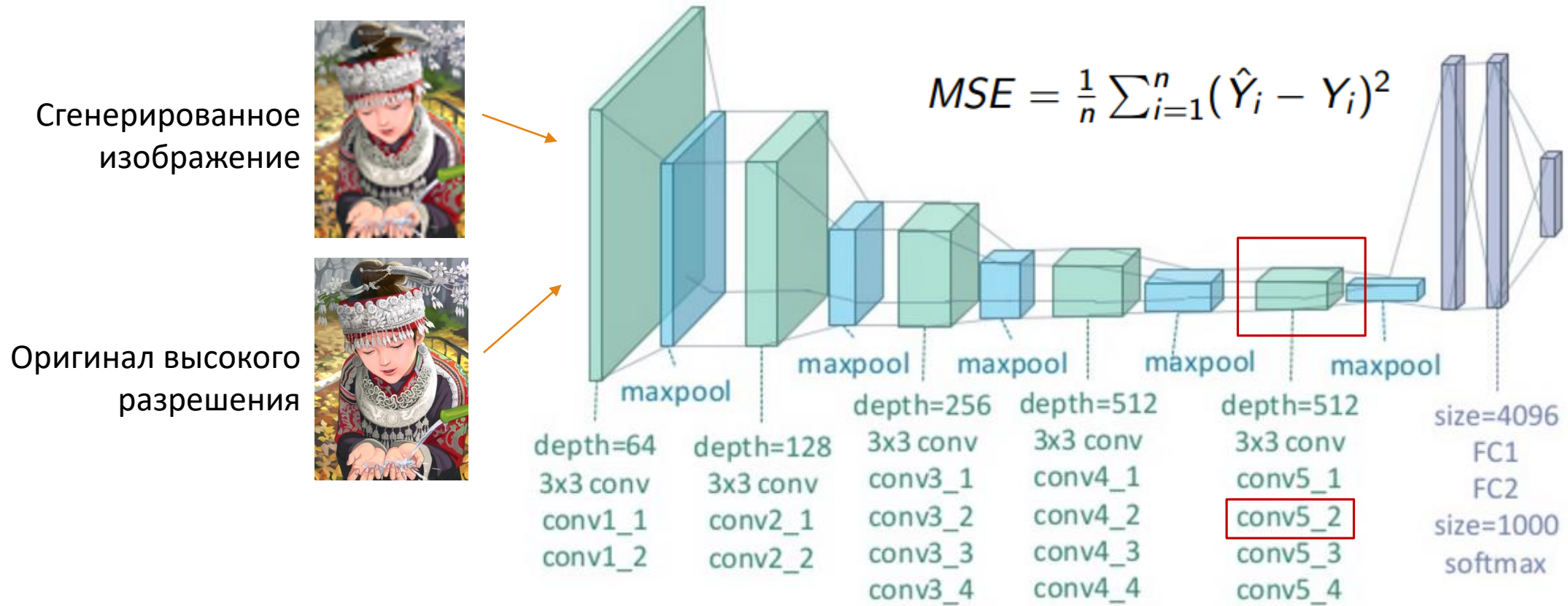
# Особенности реализации



[C. Ledig et al., "Photo-realistic single image super-resolution using a generative adversarial network," 2016]

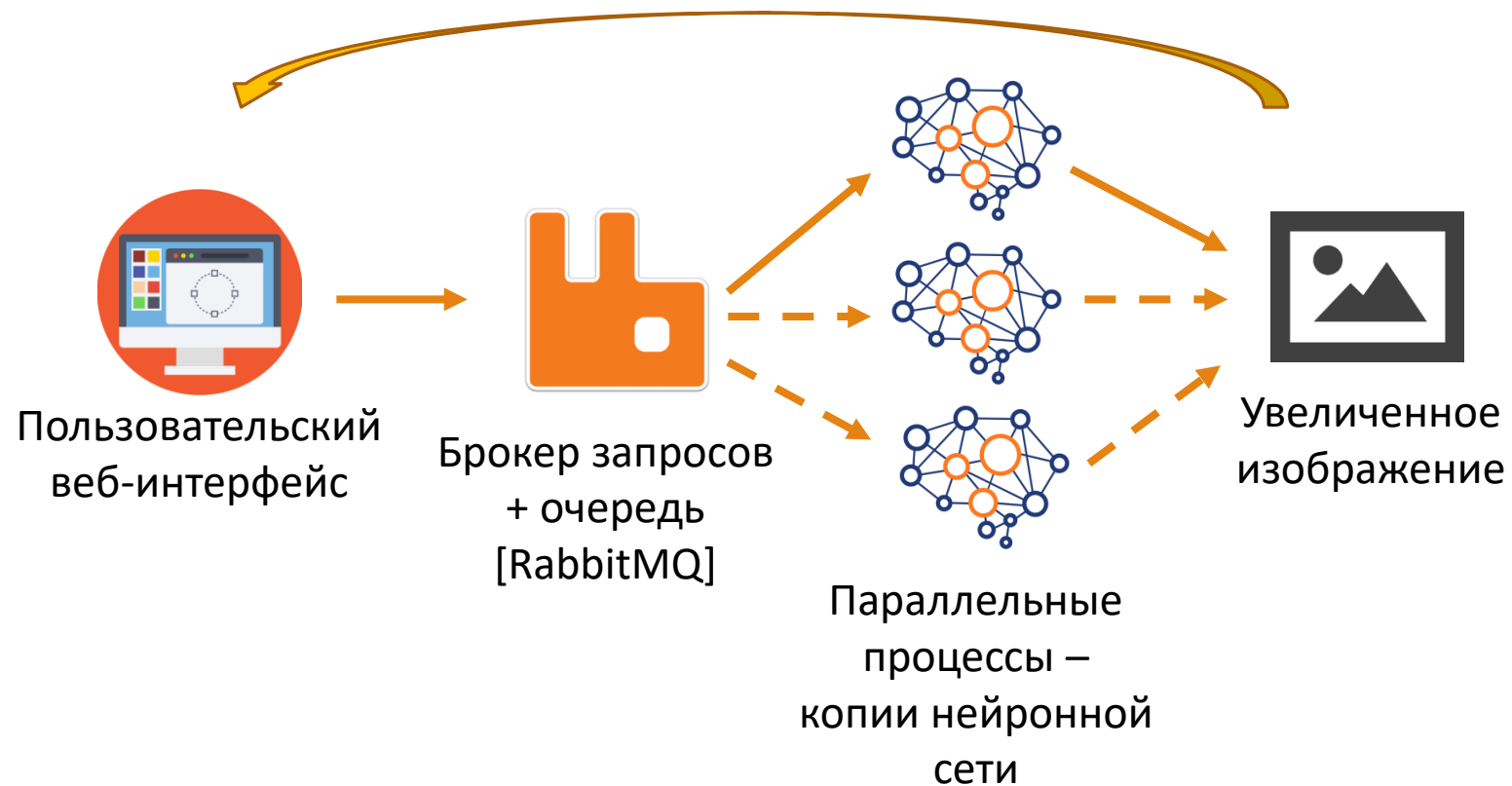
[Shi W. et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," 2016]

# Функция ошибки, основанная на признаках сети VGG19



[J. Johnson et al., "Perceptual losses for real-time style transfer and super-resolution," 2016]

# Схема веб-сервиса



# Обучение нейронных сетей

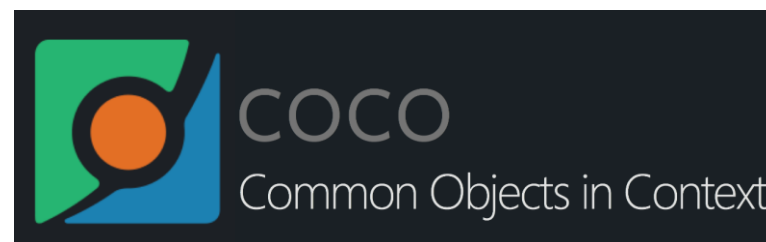
```
C:\Windows\system32\cmd.exe
[input [input ...]]

Super-resolve an image using generative adversarial networks.

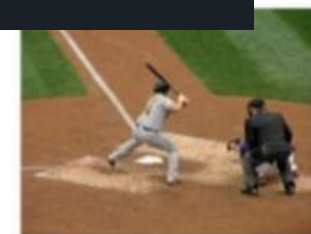
positional arguments:
  input

optional arguments:
  -h, --help            show this help message and exit
  --scale SCALE          Image scaling factor (2 or 4). (default: 4)
  --epochs EPOCHS       Number of training epochs. (default: 10)
  --equalize-colors      Transplant source image color histogram to the output.
                        (default: False)
  --model-name MODEL_NAME
                        Name of the network. (default: default)
  --apply-blur APPLY_BLUR
                        Gaussian blur kernel sigma. (default: None)
  --apply-noise APPLY_NOISE
                        Gaussian noise radius. (default: None)
  --apply-jpeg APPLY_JPEG [APPLY_JPEG ...]
                        JPEG compression level. (default: [])
  --epoch-size EPOCH_SIZE
                        Number of batches in each epoch. (default: 64)
  --batch-size BATCH_SIZE
                        Number of images in each batch. (default: 16)
  --patches PATCHES     Number of fragments to extract from each image.
                        (default: 5)
  --learning-rate LEARNING_RATE
                        Learning rate for Adam. (default: 0.0001)
  --learning-rate-decay LEARNING_RATE_DECAY
                        Adam learning rate decay factor. (default: 0.5)
  --learning-rate-epochs LEARNING_RATE_EPOCHS
                        How many epochs between Adam decays. (default: 75)
  --gen-filters GEN_FILTERS [GEN_FILTERS ...]
                        Number of convolutions in each conv layer. (default:
                        [64])
  --gen-res-blocks GEN_RES_BLOCKS
                        Number of residual blocks in the generator. (default:
                        4)
  --perceptual-weight PERCEPTUAL_WEIGHT
                        Weight of perceptual loss. (default: 1.0)
  --discr-weight DISCR_WEIGHT
                        Weight of adversarial loss. (default: 500.0)
  --discr-filters DISCR_FILTERS
                        Base number of filters in discriminator conv layers.
                        (default: 32)
  --gen-pretrain GEN_PRETRAIN
                        Number of epochs to pre-train the generator. (default:
                        10)
  --platform PLATFORM   Use GPU or CPU. (default: gpu)
```

**Данные:** случайные участки **120000+** изображений высокого разрешения из набора MS COCO.



**Проверка одной конфигурации сети:**  
до **20** часов.



The man at bat readies to swing at the pitch while the umpire looks on.



A large bus sitting next to a very tall building.



A horse carrying a large load of hay and two people sitting on it.



Bunk bed with a narrow shelf sitting underneath it.

# Демонстрация!

---



# Инструменты

---

- Язык Python 3.4
- Фреймворки:
  - Theano 0.8.2 + Lasagne 0.2 для построения нейронных сетей
  - Django 1.11 + AngularJS для веб-сервиса
- Очередь сообщений RabbitMQ 3.6.9 + Celery 4.0.2
- GPU: GeForce GTX 1080 + CUDA 8.0 + CuDNN v3



theano

Lasagne

RabbitMQ



# Сравнение с другими алгоритмами

**Set5**

Фактор увеличения	Бикубическая интерполяция	Kim [9]	Glasner [10]	SelfEx [2]	SRCNN [3]	SRResNet	SRGAN
2x	33.64	36.24	35.43	36.50	36.28	<b>36.87</b>	34.17
4x	28.42	30.07	28.84	30.33	30.08	<b>31.05</b>	29.20

**Set14**

Фактор увеличения	Бикубическая интерполяция	Kim [9]	Glasner [10]	SelfEx [2]	SRCNN [3]	SRResNet	SRGAN
2x	30.22	32.14	31.41	32.23	32.00	<b>32.84</b>	31.25
4x	25.99	27.18	26.43	27.40	27.13	<b>28.12</b>	26.11

**BSD100**

Фактор увеличения	Бикубическая интерполяция	Kim [9]	Glasner [10]	SelfEx [2]	SRCNN [3]	SRResNet	SRGAN
2x	29.55	31.11	30.28	31.18	31.11	<b>31.97</b>	29.89
4x	25.96	26.71	26.17	26.85	26.70	<b>27.15</b>	26.03

$$PSNR = 20 * \log_{10}(MAX_I) - 10 * \log_{10}(MSE); \text{ (больше – лучше)}$$

# Основные результаты работы

---

- Изучены источники по обработке изображений и глубинному обучению, опыт предшественников активно использовался при реализации;
- Реализованы архитектуры остаточной сверточной сети и конкурирующих нейронных сетей для повышения разрешения изображений;
- Разработан веб-сервис для демонстрации подхода, а также инструмент обучения моделей для него;
- Произведено экспериментальное сравнение качества изображений, восстановленных реализованным методом и аналогами;
- Разработана техническая документация.

# Дальнейшая работа

---

- Исследование архитектурных модификаций представленной сети-генератора, а также способов обучения конкурирующих сетей;
- Добавление сети-классификатора для определения типов объектов на изображении;
- Использование рекуррентных слоев (сетей с кратковременной памятью) для восстановления последовательностей кадров видео, а не только отдельных изображений.

# СПИСОК ИСТОЧНИКОВ

---

1. Gu S. et al. Convolutional sparse coding for image super-resolution //Proceedings of the IEEE International Conference on Computer Vision. – 2015. – С. 1823-1831.
2. Huang J. B., Singh A., Ahuja N. Single image super-resolution from transformed self-exemplars //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2015. – С. 5197-5206.
3. Dong C. et al. Image super-resolution using deep convolutional networks //IEEE transactions on pattern analysis and machine intelligence. – 2016. – Т. 38. – №. 2. – С. 295-307.
4. Kim J., Kwon Lee J., Mu Lee K. Deeply-recursive convolutional network for image super-resolution //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2016. – С. 1637-1645.
5. Johnson J., Alahi A., Fei-Fei L. Perceptual losses for real-time style transfer and super-resolution //European Conference on Computer Vision. – Springer International Publishing, 2016. – С. 694-711.
6. Goodfellow I. et al. Generative adversarial nets //Advances in neural information processing systems. – 2014. – С. 2672-2680.
7. Ledig C. et al. Photo-realistic single image super-resolution using a generative adversarial network //arXiv preprint arXiv:1609.04802. – 2016.
8. Shi W. et al. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2016. – С. 1874-1883.
9. Kim K.I., Kwon Y. Single-image super-resolution using sparse regression and natural image prior // IEEE transactions on pattern analysis and machine intelligence, Vol. 32, 2010. pp. 1127-1133.
10. Glasner D., Bagon S., Irani M. Super-resolution from a single image // Computer Vision, 2009 IEEE 12th International Conference on. 2009. pp. 349-356.





# Спасибо за внимание!

---

## Сверточная нейронная сеть и ее составляющие:

