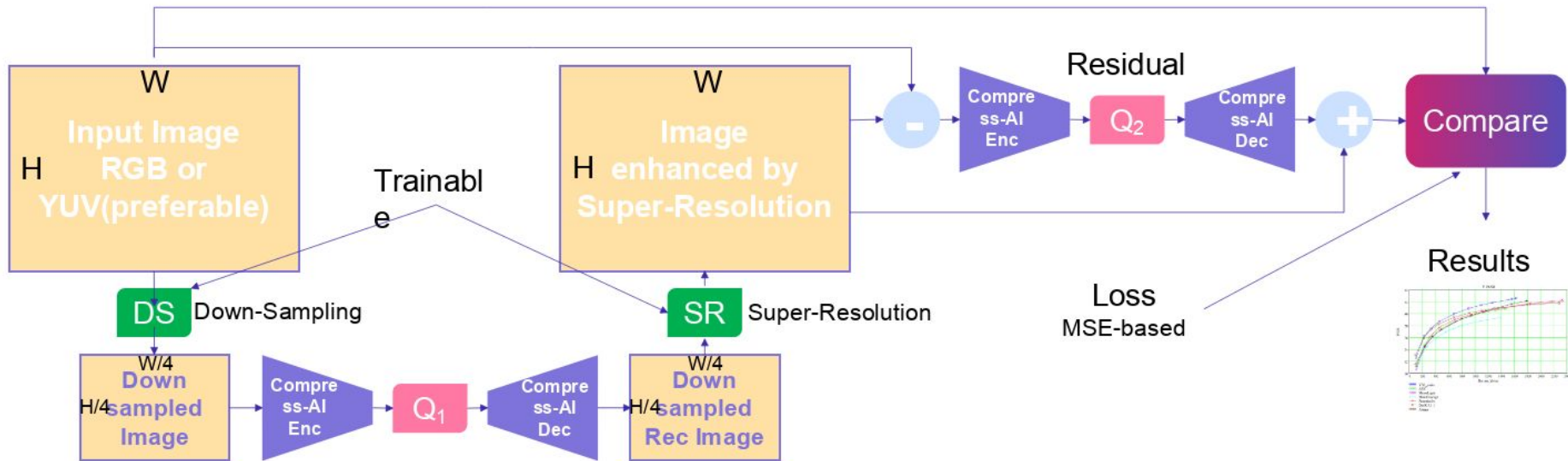


Task for Mark

Mark Mirgaleev

Схема модели



Сравнение методов

Значения PSNR
и bpp - среднее на всем
датасете kodak

Zoom in for best view



Условные обозначения

down=False - обучаемое понижение размерности (см. код)

cod1 - qr кодеков (первая цифра - первый кодек, вторая - второй кодек)

res = True - используется схема с двумя кодеками из презентации,

res = False - после SR-а кодек отсутствует

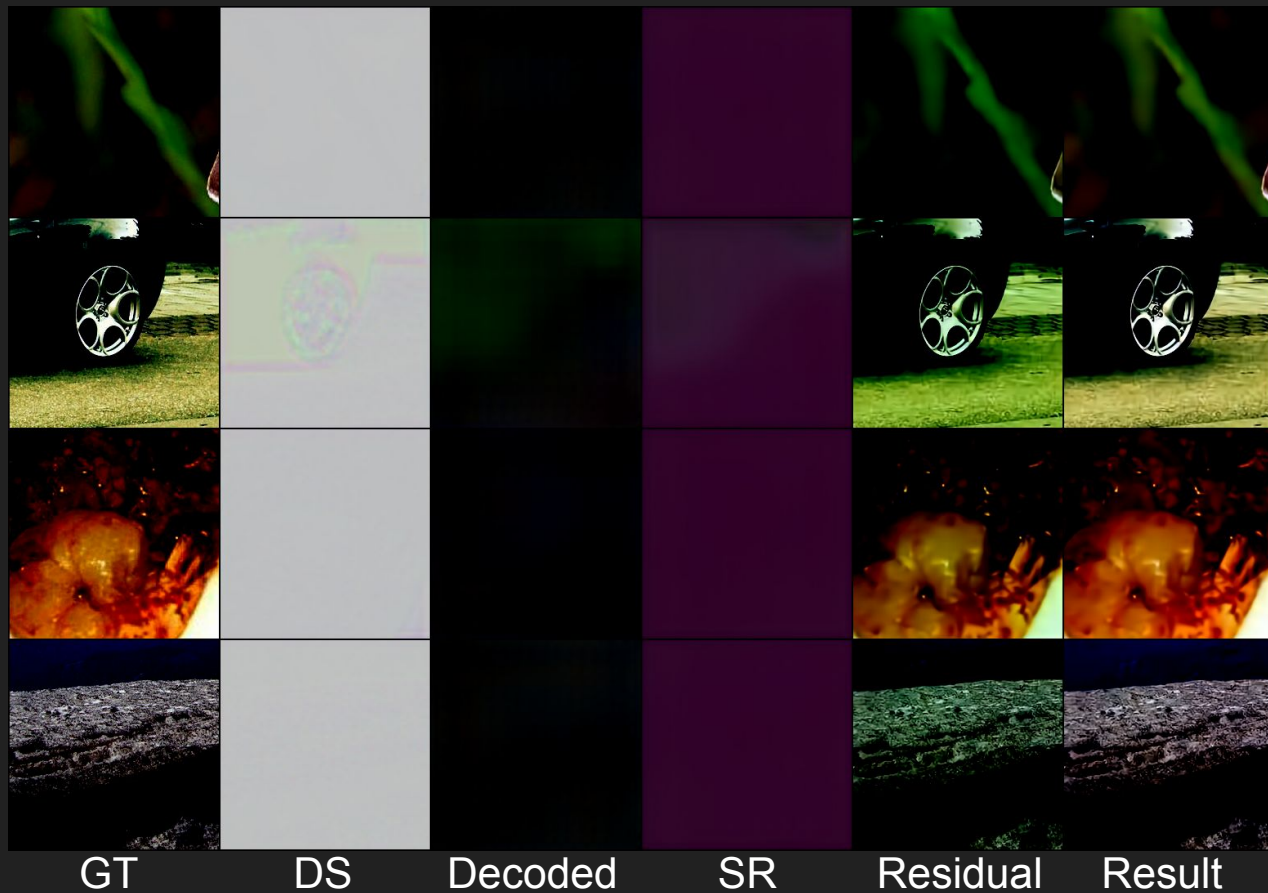
gan=True - использовался gan для обучения SR

vgg=True - использовалась разница VGG фичи в качестве лосса

Значения PSNR и bprp указаны в среднем на датасете KODAK

DS(learned) -> codec(qp=1) -> ... -> codec(qp=5)

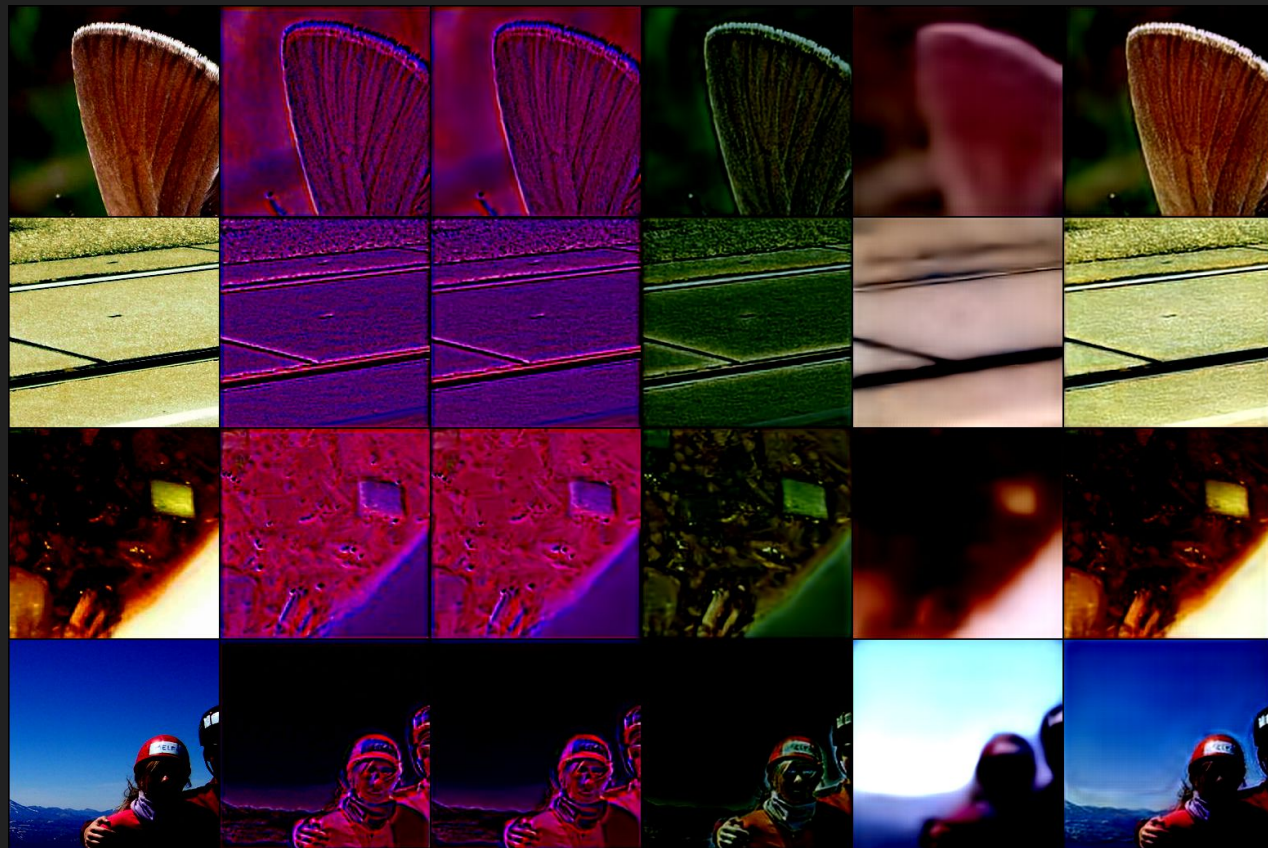
Как видно из-за того что в конце сети кодек лучше, то веса моделей до второго кодека занулились, т.к. модели выгоднее (по MSE) закодировать входное изображение в исходном разрешении



DS(learned) -> codec(qp=5) -> ... -> codec(qp=1)

Теперь “хороший” кодек в начале, видно, что он кодирует высокие частоты, а второй кодирует низкие. Но такая стратегия, судя по результатам, хуже той, что на слайде назад

Попробуем добавить регуляризацию, на выход SR модели



GT

Down

Decoded

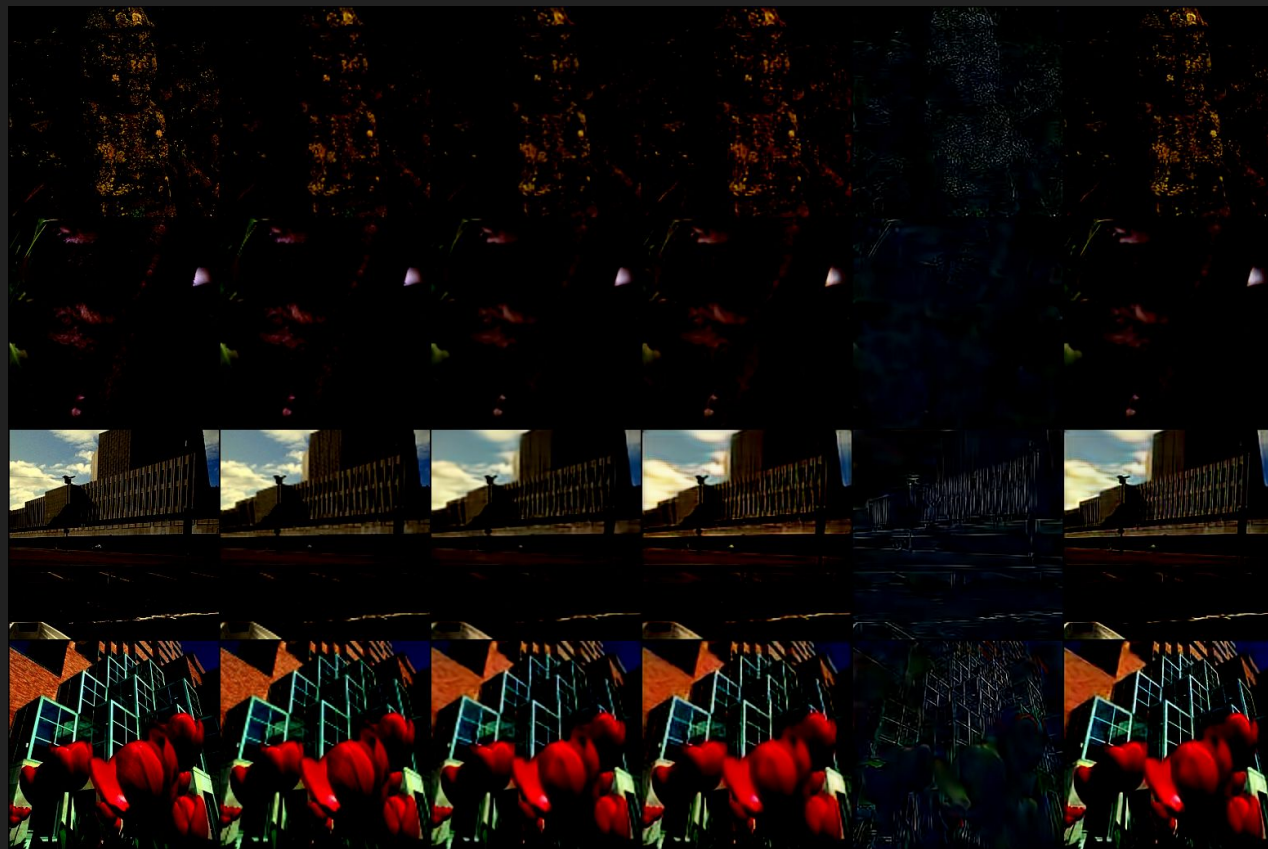
SR

Residual

Result

DS(bicubic) -> codec(qp=4) -> ... -> codec(qp=4)

Добавление
регуляризации в виде
MSE между SR и GT
помогло получить
“ожидаемый”
результат - первый
блок кодирует низкие
частоты, второй
высокие



GT

Down

Decoded

SR

Residual

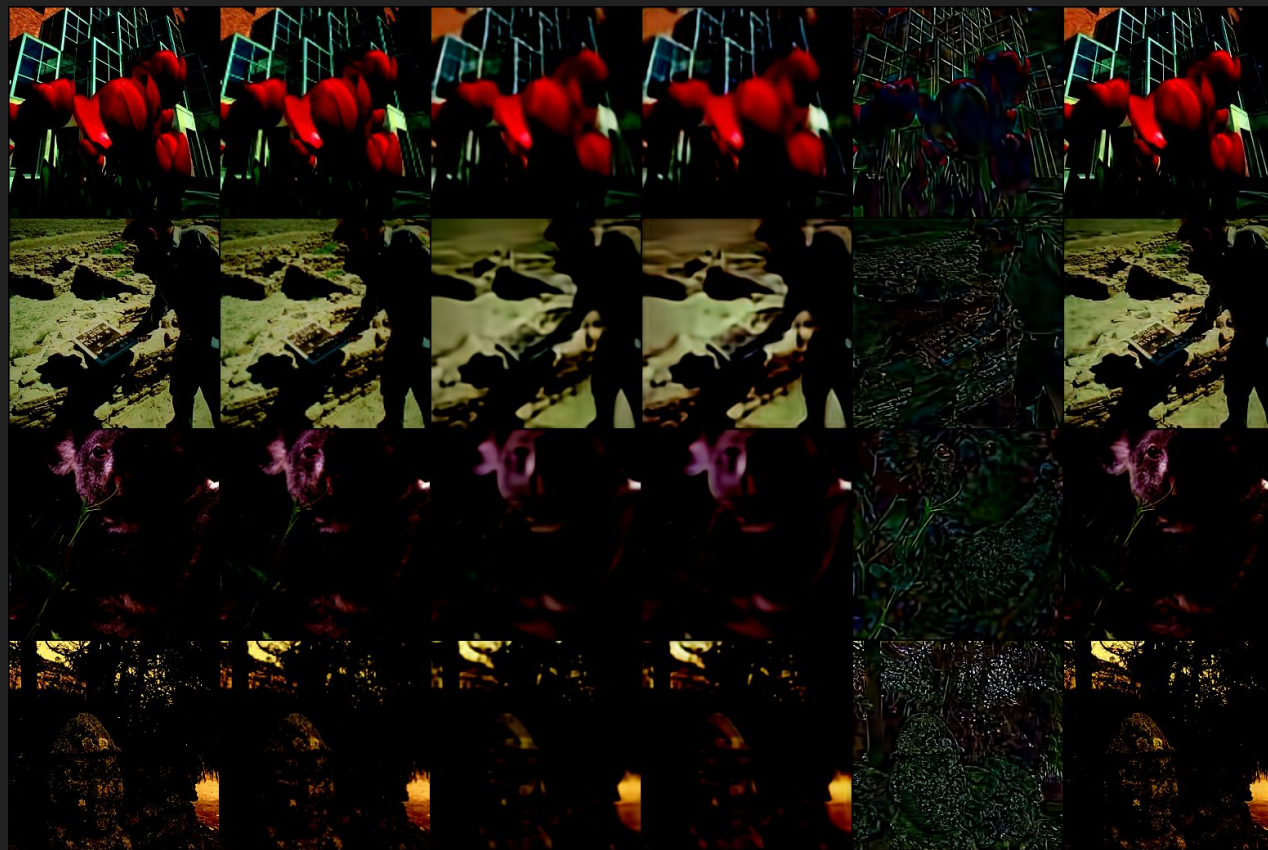
Result

DS(bicubic) -> codec(qp=1) -> ... -> codec(qp=5)

Лучшая модель (по PSNR)

Веса у лоссов для первого и второго кодеков брались в отношении 1:1

$$\mathcal{L} = \frac{1}{2}\mathcal{L}_{sr} + \frac{1}{2}\mathcal{L}_{res}$$



GT

Down

Decoded

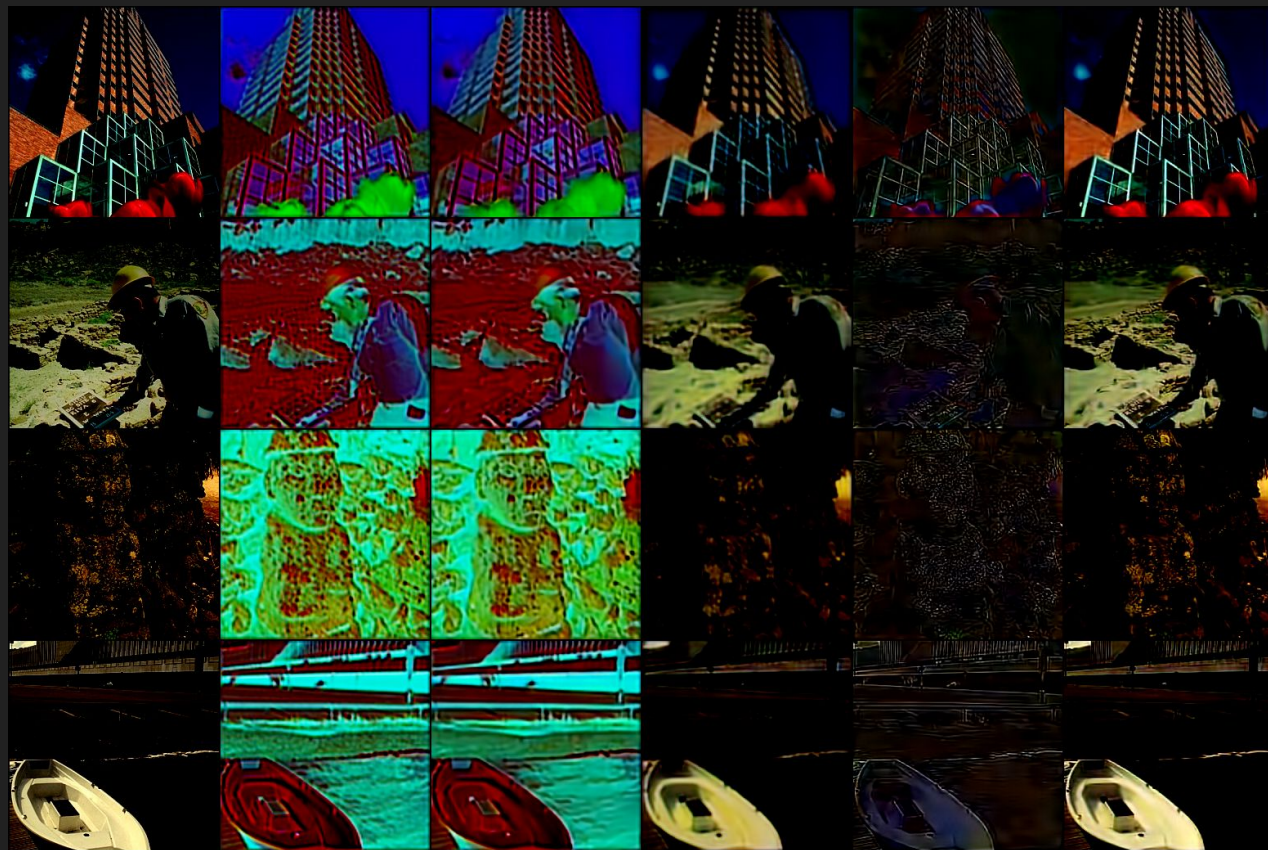
SR

Residual

Result

DS(learned) -> codec(qp=4) -> ... -> codec(qp=4)

Регуляризация не
исправила, что
downsampling
обучается не на
“естественный” домен



GT

Down

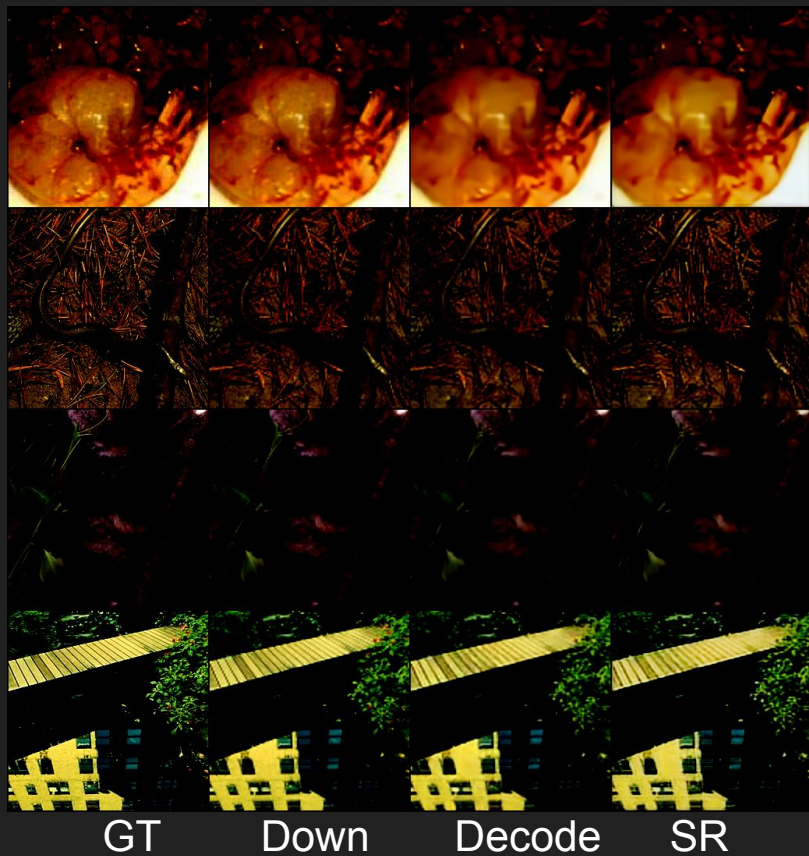
Decoded

SR

Residual

Result

Обучение с GAN(слева) и GAN + VGG (справа). На валидации выглядит неплохо, но на тесте (см. 3 слайд) - проблемы



Что пробовал

- Брать YUV: прироста не было, наверное, т.к. кодек обучался в RGB домене(я посмотрел - в коде модели forward делают в rgb)
- Веса для линейной комбинации лоссов взял из ESRGAN, но с ними ничего хорошего не получилось, возможно, если по перебирать веса, то получится что-то стоящее
- Обучался на кропах 256x256 на BSD100 и выборке из датасете Open Images на повышение разрешения в 2 раза

Выводы

- Видимо такая схема сжатия **не эффективна**, т.к. судя по проведенным экспериментам проще сжимать в исходном разрешении
- Подтвердить эту догадку можно тем, что по данному методу сжатия почти нет статей, видимо весомых результатов не было получено
- Обучаемое понижение размерности вело себя не так как я ожидал - выход такой модели получался в другой цветовой гамме
- Методы без второго кодека имеют проблемы с цветом и в целом хуже методов с двумя кодеками