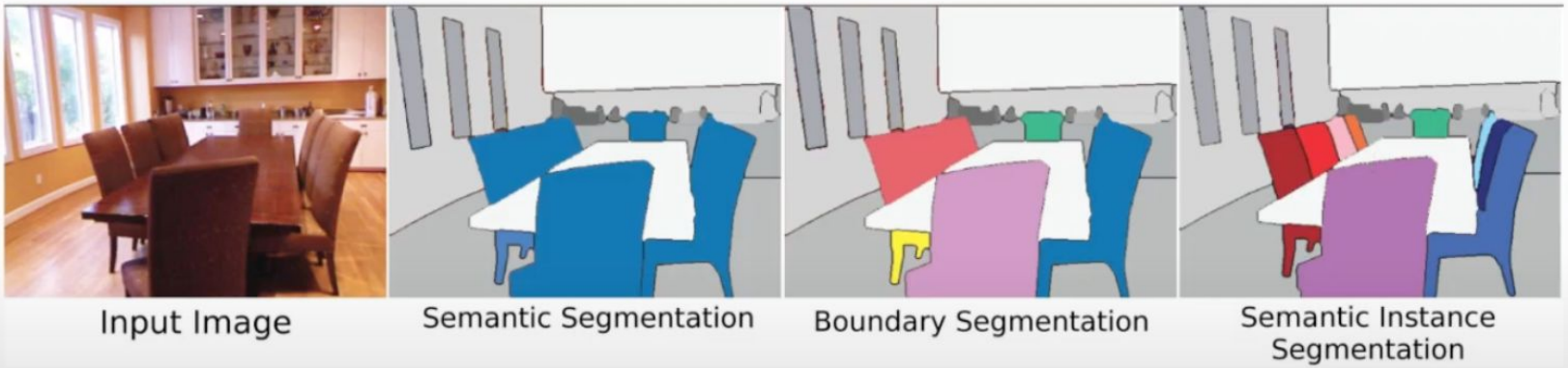
cv and ml

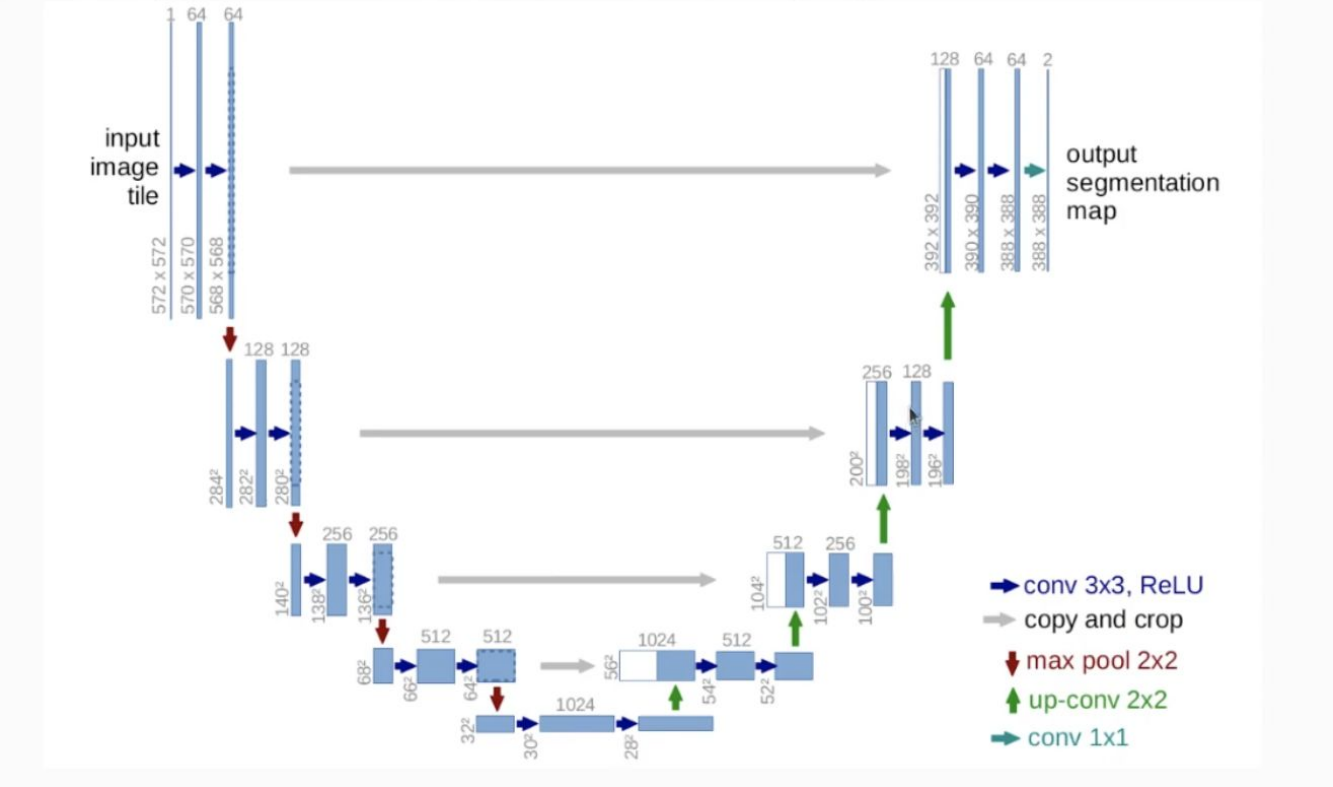
img2img

Владимир Глазачев
cv в rosebud.ai

Решили сегментацию

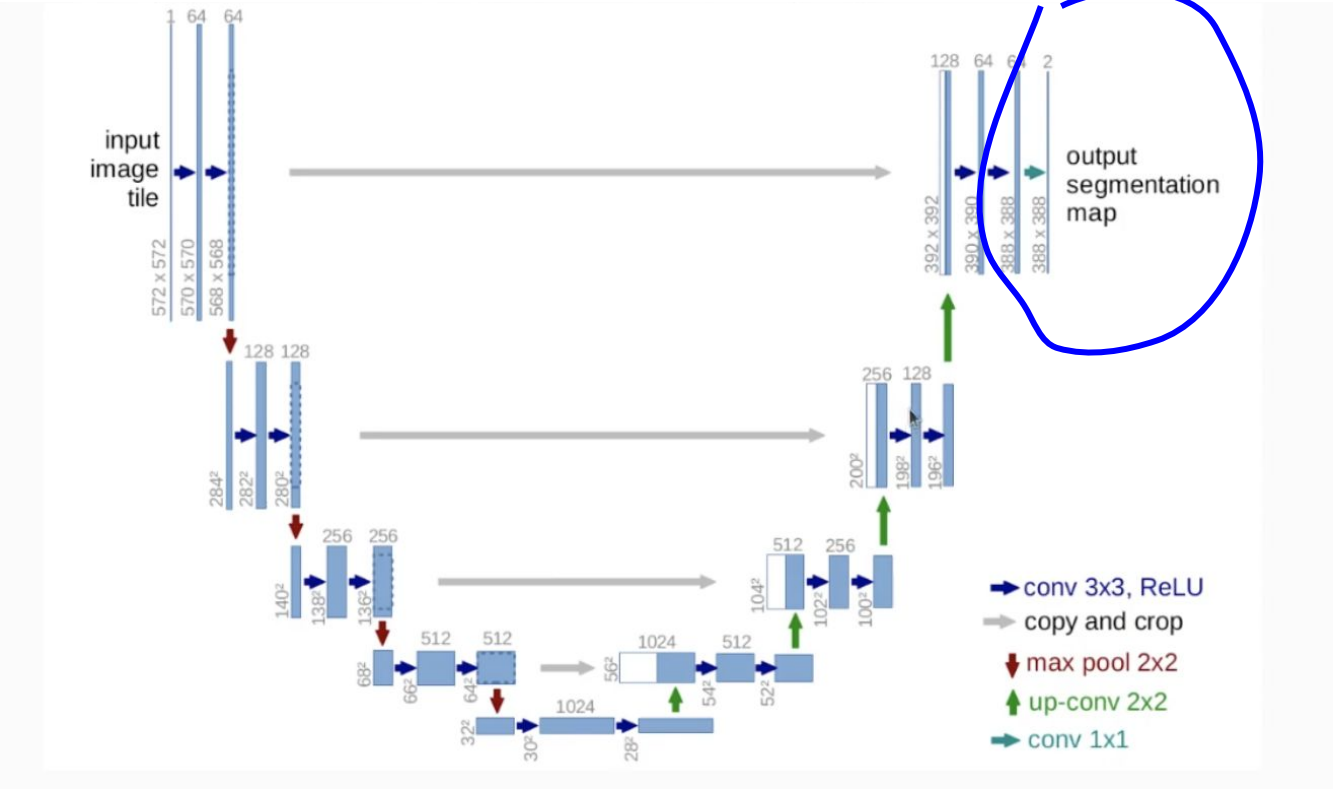


Решили сегментацию



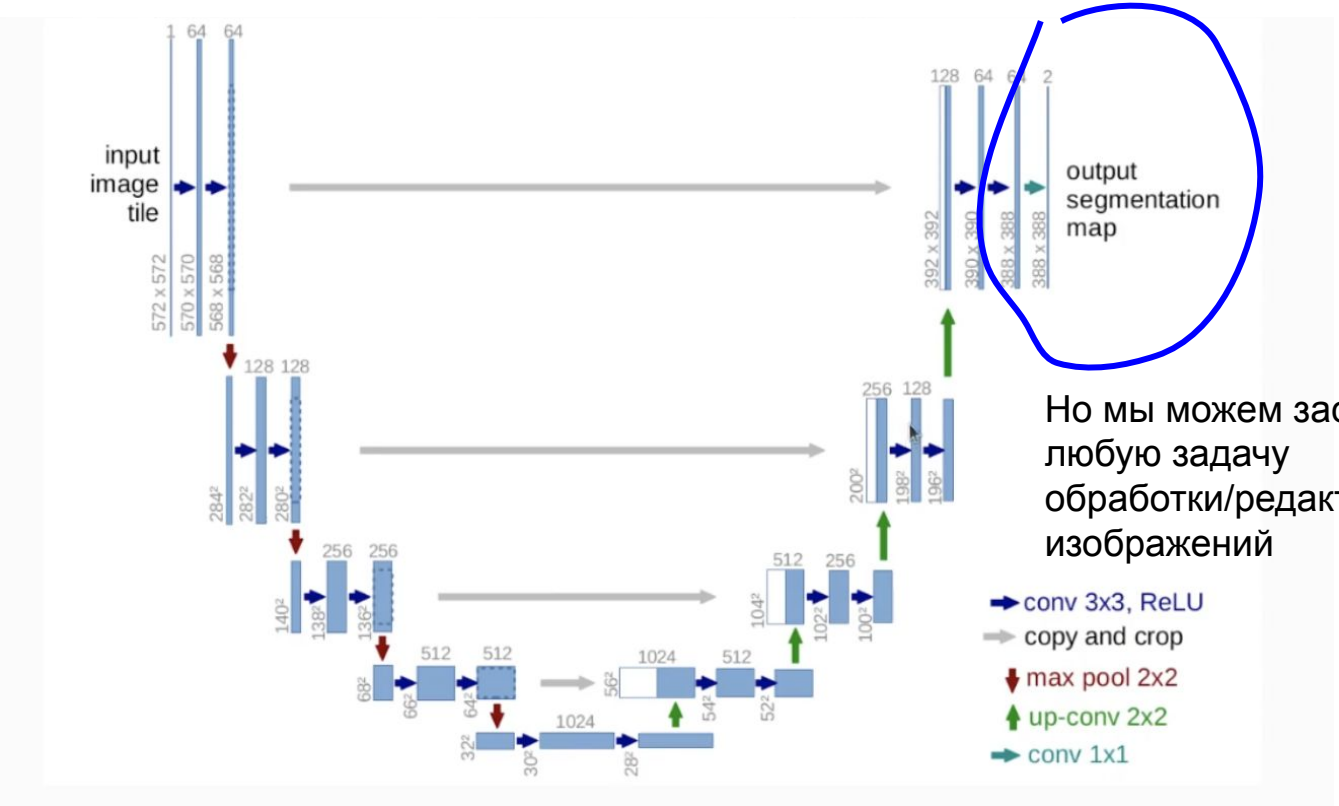
Решили сегментацию

Тут на выходе в каждом пикселе вероятности принадлежности к классу



Решили сегментацию

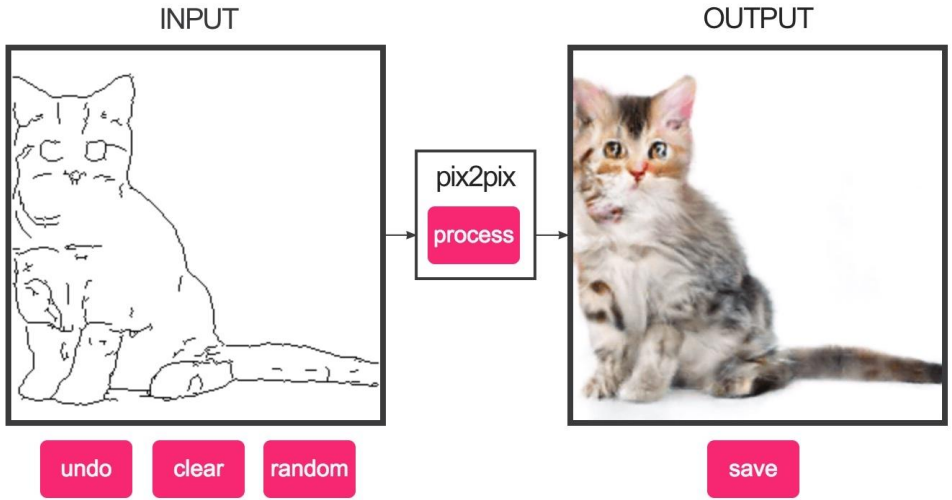
Тут на выходе в каждом пикселе вероятности принадлежности к классу



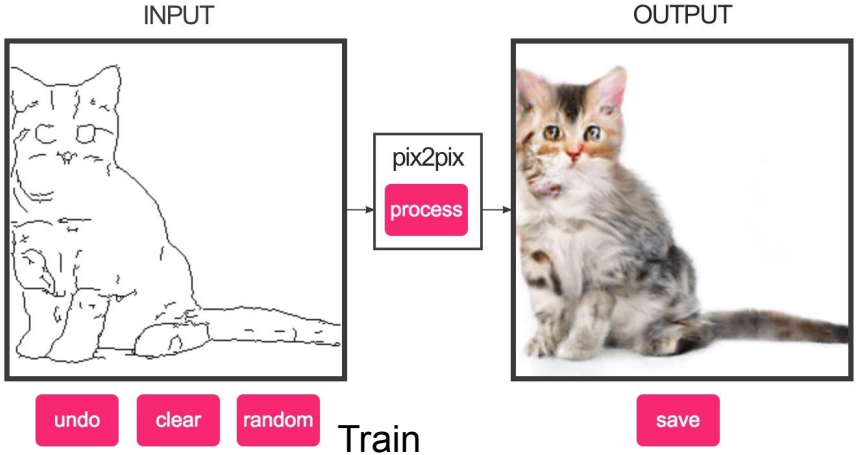
Но мы можем засунуть сюда любую задачу обработки/редактирования изображений

Из “картинки” в “картинку”

- Мы можем засунуть любое изображения (n канальный 3 мерный тензор) и получить любое изображение
- Разница только в том где мы такой датасет возьмем и как сформулируем loss (ну и domain specific тонкости :))



edges2image



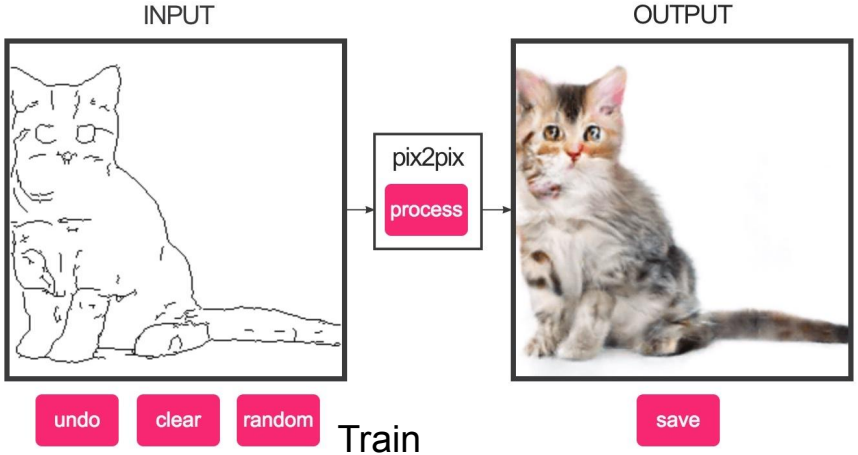
Train

Inference

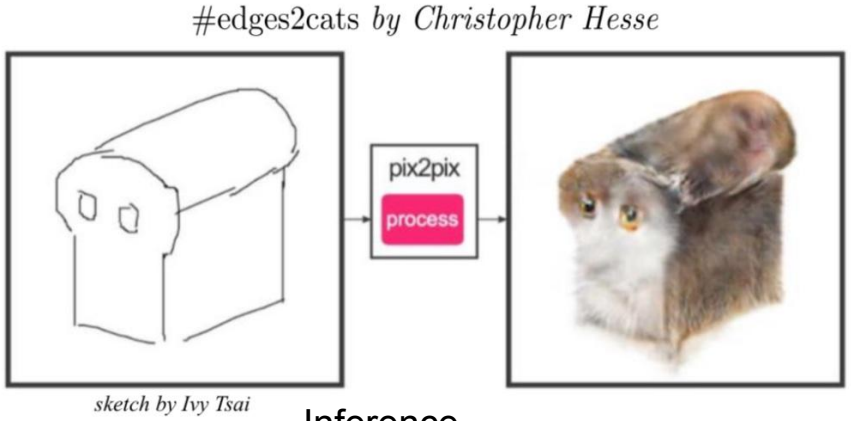
Датасет: X - любые картинки , Y - через детектор границ нагенерим границы

loss - например mse

edges2image



Train

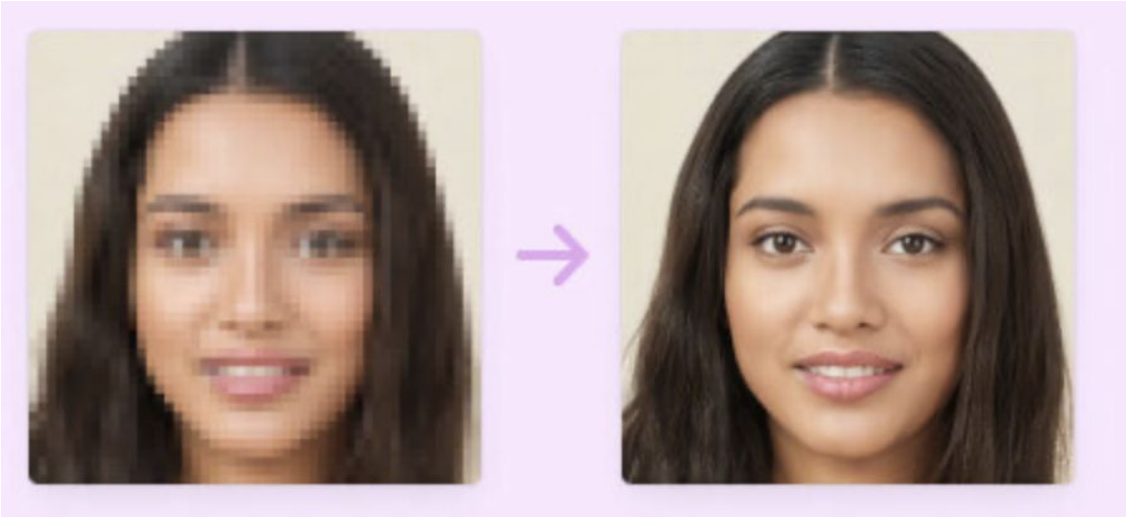


Inference

Датасет: Y - любые картинки , X - через детектор границ нагенерим границы

loss - например mse (с mse будут проблемы мы к этому сейчас прийдем) ; в целом если результат изображение (в общем смысле того что такое изображение :)) - нужен способ измерять похожесть картинок

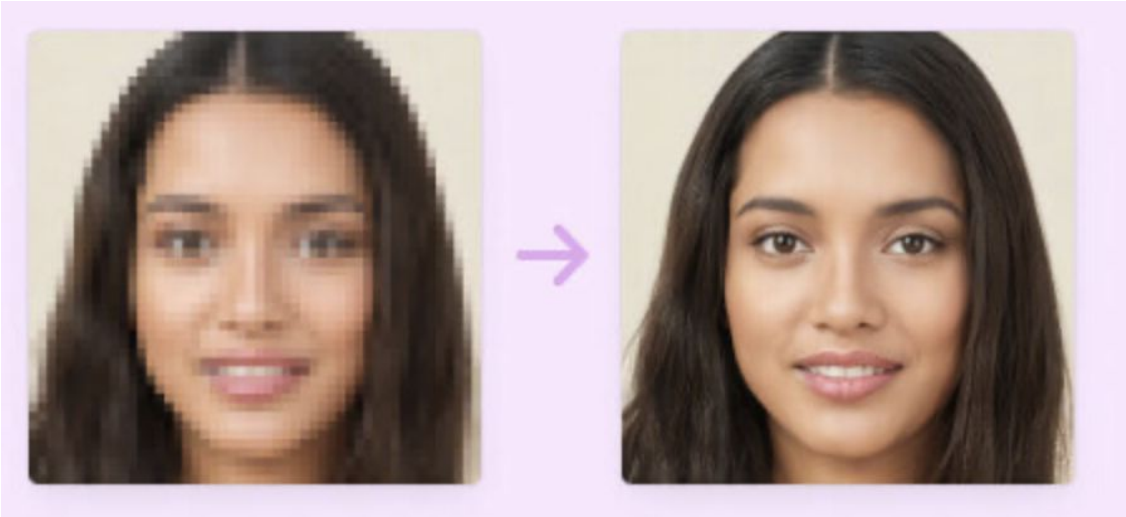
superresolution



Датасет: Y - хайрез картинки, X - сжимаем/ломаем картинки аугментациями

проблемы?

superresolution



Датасет: Y - хайрез картинки, X - сжимаем/ломаем картинки аугментациями

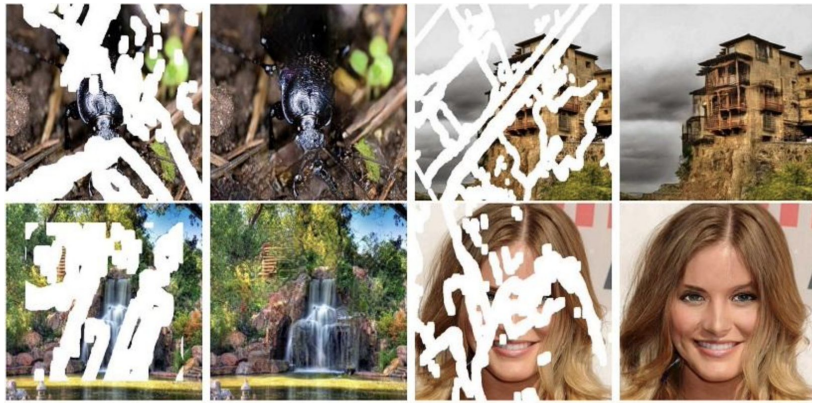
проблемы? - real world деградация картинок часто не похожа на аугментации, но все равно можно придумать неплохие аугментации

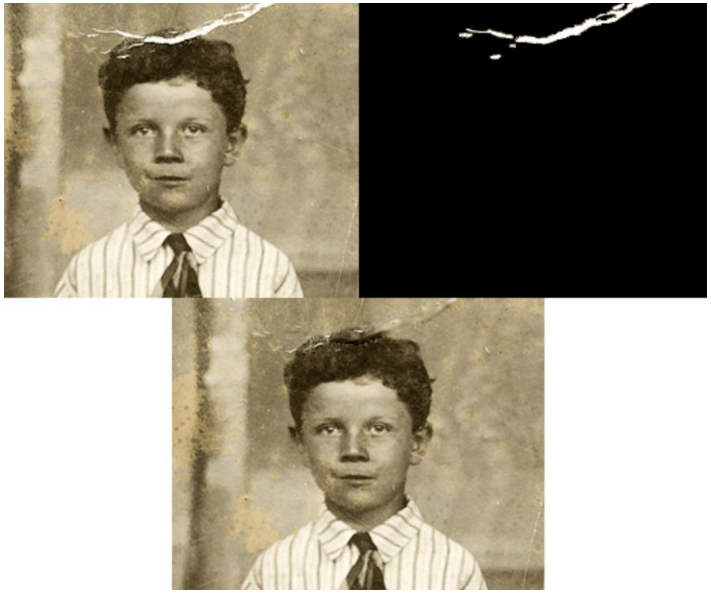
colorization

Датасет: Y - картинки, X - чернобелим картинки

проблемы? - real world чернобелые картинки не получены из обесцвечивания цветных, а были спецификой старых камер и проявочной химии (и часто были не чб а монохромными, типо оттенки красного)

inpainting





Датасет: Y - картинки, X - вырезаем рандомные маски

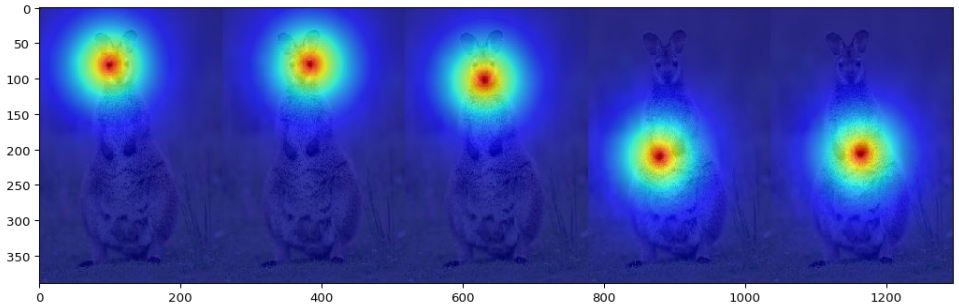
проблемы? - маску на инференсе надо будет как то найти ; дырки могут быть не похожи на те что вы вырезали на трейне и это может влиять

собираем в кучу

сегментация, детектим дырки
инпейнтинг - заполняем дырки
колоризация - раскрашываем картинку
super resolution - делаем качество повыше

через несколько моделей решаем задачу restoration ; можно было бы пытаться решить одной моделью - но сложно измерять прогресс/качество и итеративно улучшать

key points detection



Датасет: X - картинки, Y - ручная разметка, точки на изображении ; предсказать точку сложно - так что можно их размыть и потом брать локальный максимум

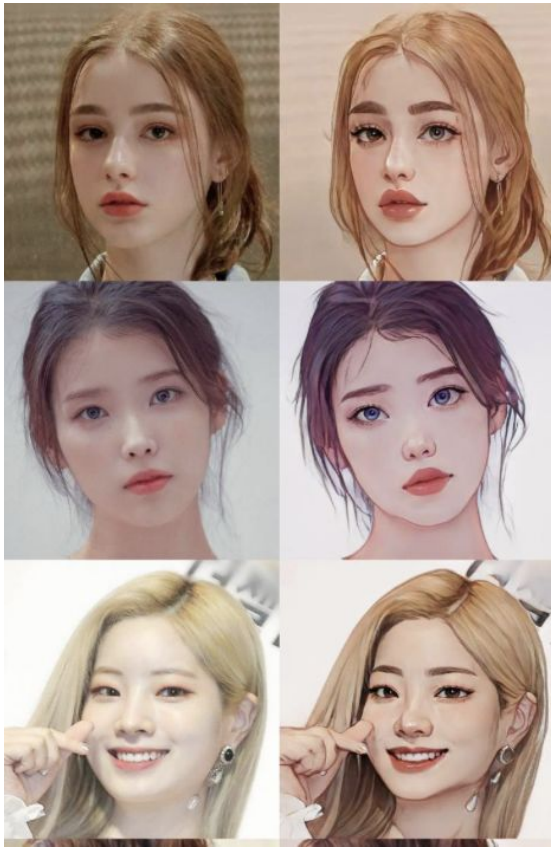
лосс - можно класификационные использовать, можно что то из регрессии взять

depth estimation

Датасет: X - картинки, Y - ground truth глубина с какой то камеры
или X - рендер, Y - ground truth глубина которую мы можем из рендера вытащить
loss - l1 или l2 между настоящей и предсказанной маской

style transfer

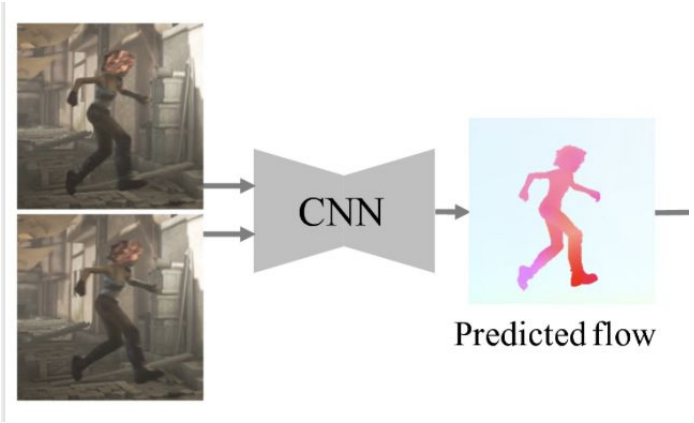
Датасет: X - картинки, Y - где то надо взять стилизацию



style transfer

примеры из faceapp приложения, это все в каком то прибрежении можно отнести к style transfer задачам (похожие методы внутри)

optical flow



Датасет: X - картинки, пары соседних кадров

Y - руками размеченный flow или

X - рендер игры/какой то 3д сцены, Y - truth flow который мы достаем из рендер движка

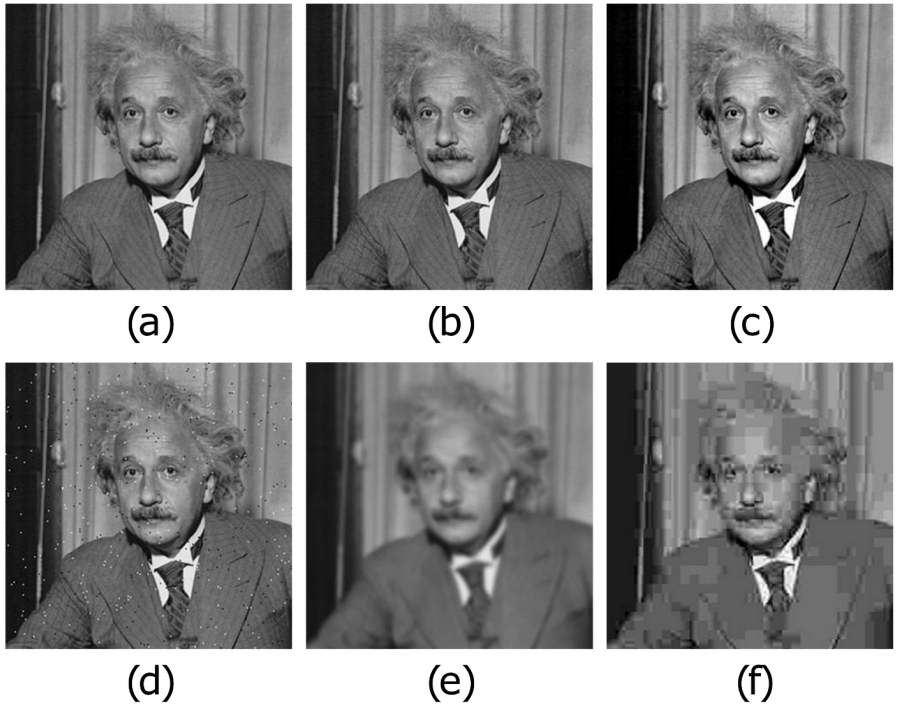
loss - mse между настоящим флоу и посчитанным

image2image problems

- Если есть парный датасет или мы его можем синтетически сгенерировать из картинок - проблему можно нормально решить
- Многие проблемы comp vision можно сформулировать как image2image translation
- Осталось пара проблем:
  - Нету парного датасета
  - Как нормально измерять похожесть картинок в лоссе

Измеряем качество

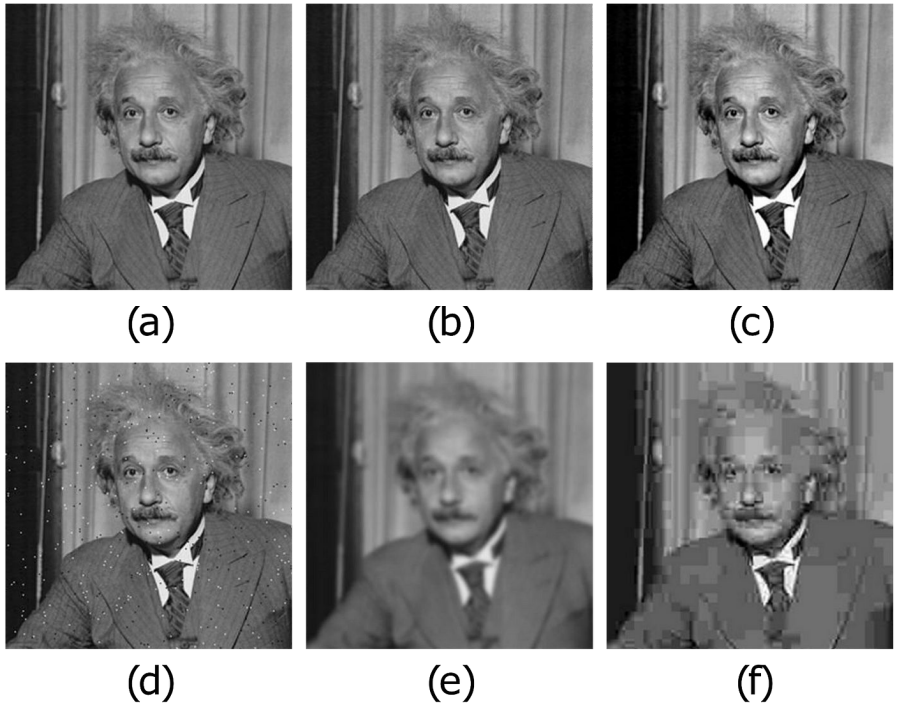
- регрессионные лоссы не очень хорошо работают в генерации изображений



mse одинаковый

Измеряем качество

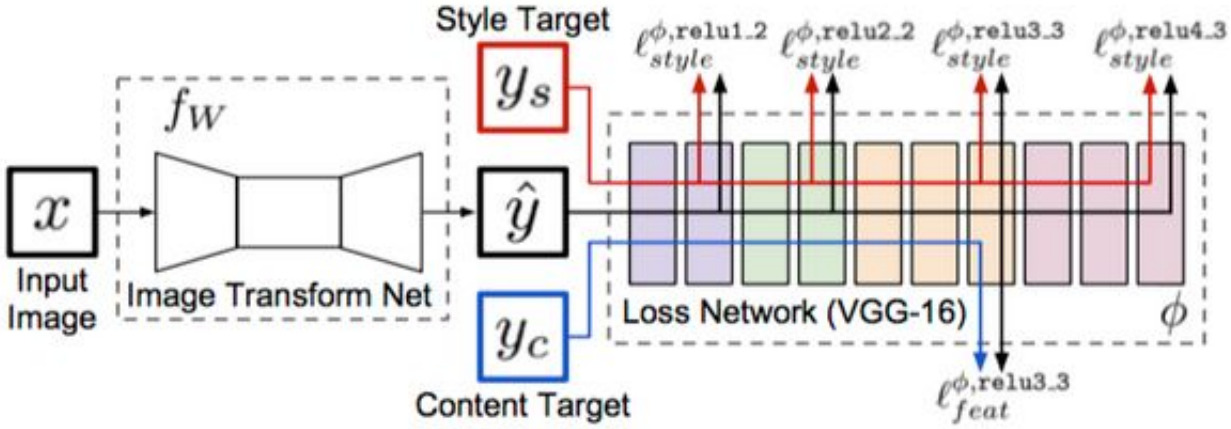
- регрессионные лоссы не очень хорошо работают в генерации изображений
- С точки зрения mse / l1 / многих других pixel лоссов, пространство решений которое норм для них - не удовлетворяет человеческому восприятию



mse одинаковый

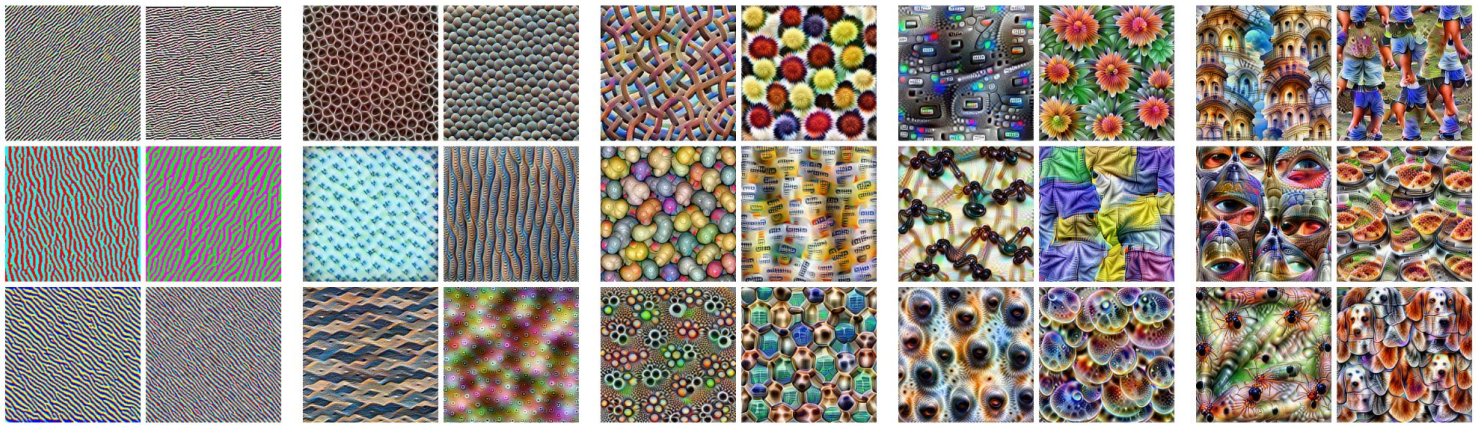
perceptual loss

Мы помним что весь dl про feature learning :)
Мы обсуждали что фичи которые извлекает какая нибудь convnet на разных слоях - имеет похожие свойства. На первых слоях low level фичи типо углов, дальше - фичи начинают активироваться на объекты
локальные фичи первых слоев - это и есть некоторое “приближение” стиля
фичи слоев дальше (где то посередине) - “приближение” описания сцены



Напоминание про фичи

- https://distill.pub/2017/feature-visualization/



Feature visualization allows us to see how GoogLeNet [1], trained on the ImageNet [2] dataset, builds up its understanding of images over many layers. Visualizations of all channels are available in the appendix.

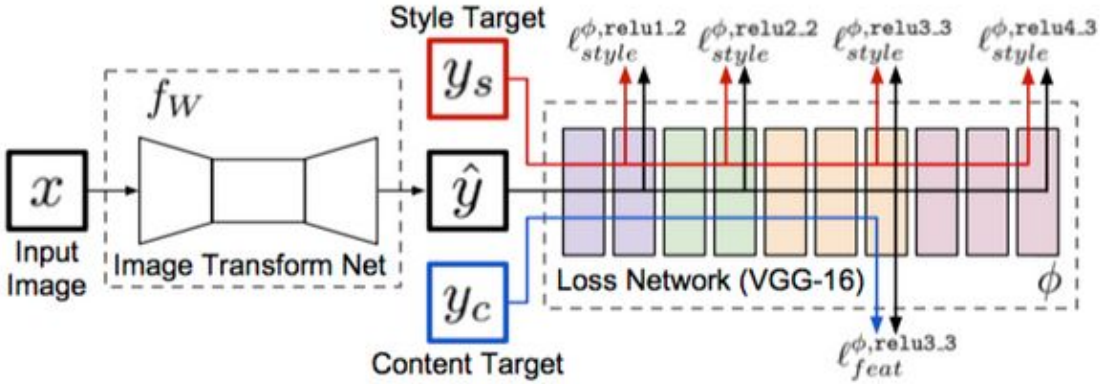
perceptual loss

Ground Truth	Bicubic	Ours (ℓ_{pixel})	SRCNN [13]	Ours (ℓ_{feat})
This image	31.78 / 0.8577	31.47 / 0.8573	32.99 / 0.8784	29.24 / 0.7841
Set5 mean	28.43 / 0.8114	28.40 / 0.8205	30.48 / 0.8628	27.09 / 0.7680

Ground Truth	Bicubic	Ours (ℓ_{pixel})	SRCNN [13]	Ours (ℓ_{feat})
This Image	21.69 / 0.5840	21.66 / 0.5881	22.53 / 0.6524	21.04 / 0.6116
Set14 mean	25.99 / 0.7301	25.75 / 0.6994	27.49 / 0.7503	24.99 / 0.6731
BSD100 mean	25.96 / 0.682	25.91 / 0.6680	26.90 / 0.7101	24.95 / 63.17

Перенос стиля без парного датасета

- можно попробовать оптимизировать стиль через perceptual loss ; нам тут парное соответствие и не нужно
- Датасет: X - любые картинки, style target одно или несколько изображений стиля ; Y - отсутствует



Так делался традиционный быстрый style transfer, на самом деле тут можно вообще не обучать чтобы стилизовать - но будет медленно

pix2pix model

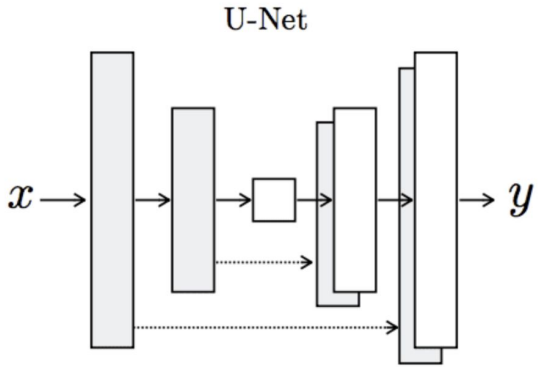
encoder:

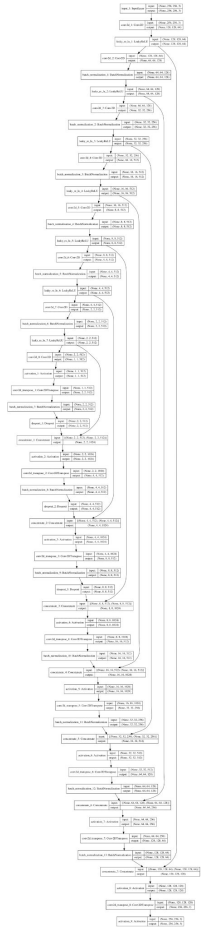
C64-C128-C256-C512-C512-C512-C512-C512

decoder:

CD512-CD512-CD512-C512-C256-C128-C64

большой bottleneck!
важно для style transfer задач





Нифига не видно, но слоев много

Как сделать лучше

- perceptual loss помогает, но только если мы решаем задачу не очень точного style transfer или у нас есть парный датасет
- нужно как то научиться измерять качество картинки не имея с чем ее сравнить :)
- через это прийдем к ганам в следующий раз