

cv and ml

segmentation continue

Владимир Глазачев

cv в rosebud.ai



Input Image



Semantic Segmentation



Boundary Segmentation



Semantic Instance Segmentation



Input Image



Semantic Segmentation



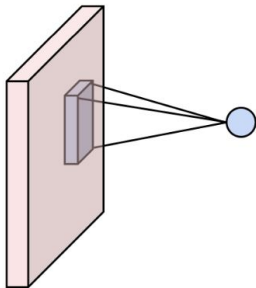
Boundary Segmentation



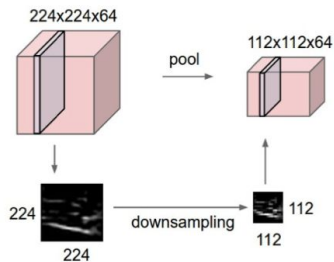
Semantic Instance Segmentation

Компоненты классификации

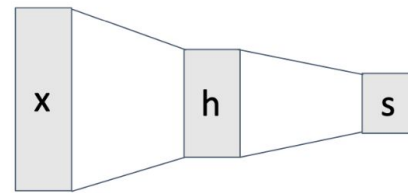
Convolution Layers



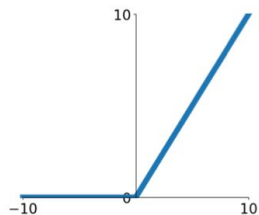
Pooling Layers



Fully-Connected Layers

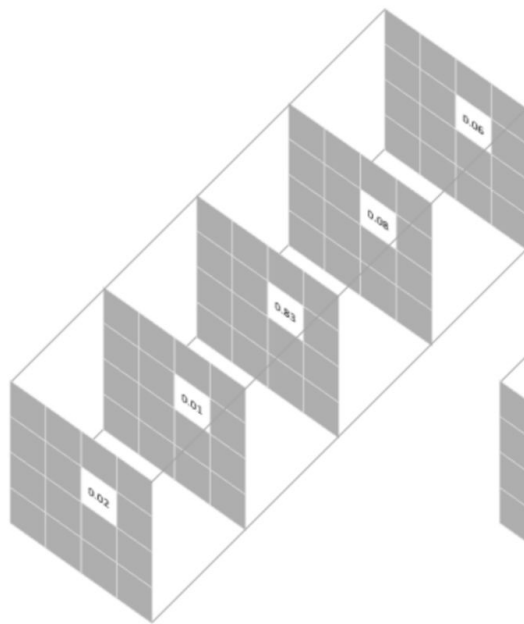


Activation Function

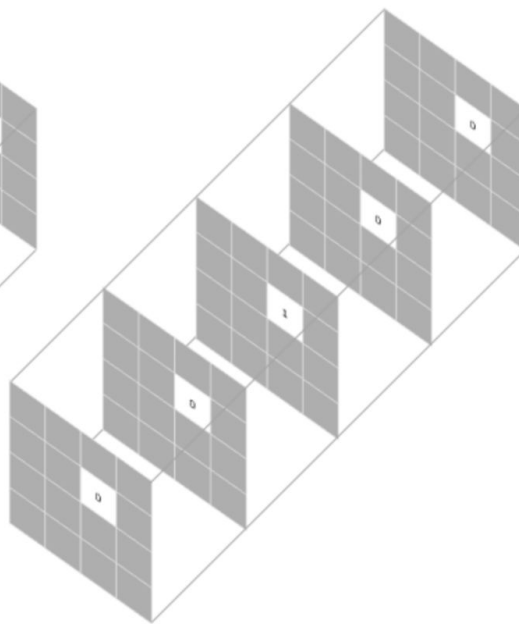


- + GlobalAvgPooling
- + BatchNorm
- + Residual connections
- + ...

Лосс в сегментации



Prediction for a selected pixel



Target for the corresponding pixel

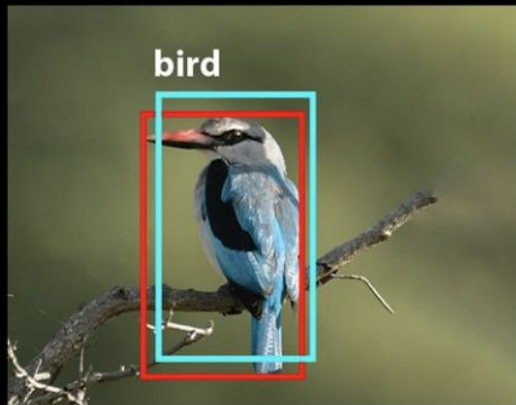
Pixel-wise loss is calculated as the log loss, summed over all possible classes

$$-\sum_{classes} y_{true} \log(y_{pred})$$

This scoring is repeated over all **pixels** and averaged

cross entropy for pixel predictions

Метрика в сегментации

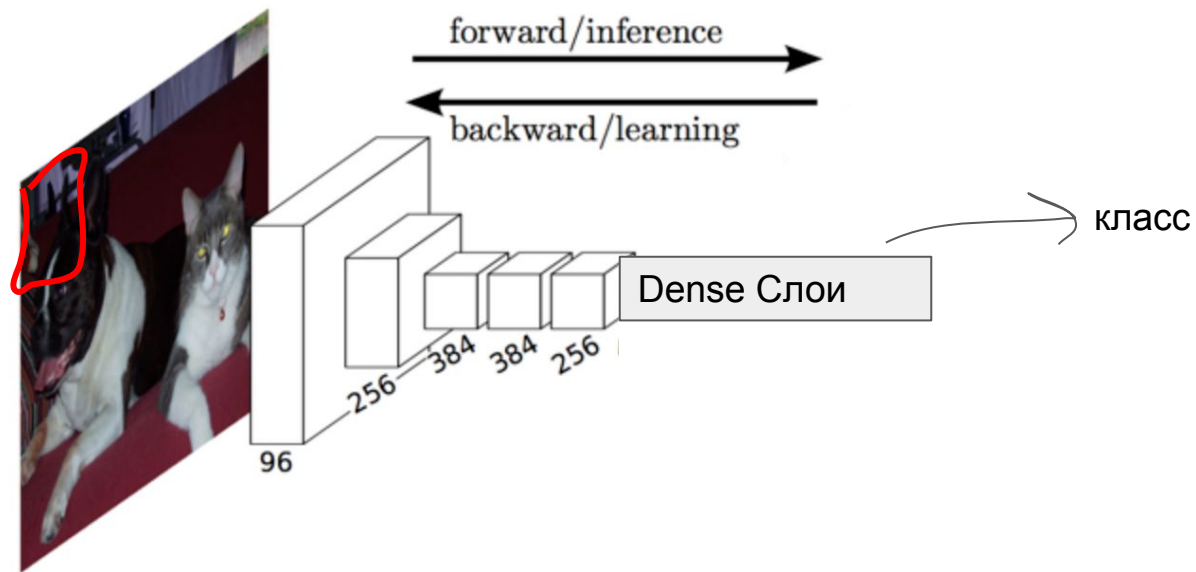


$$\text{IoU} = \frac{\text{Intersection}}{\text{Union}}$$

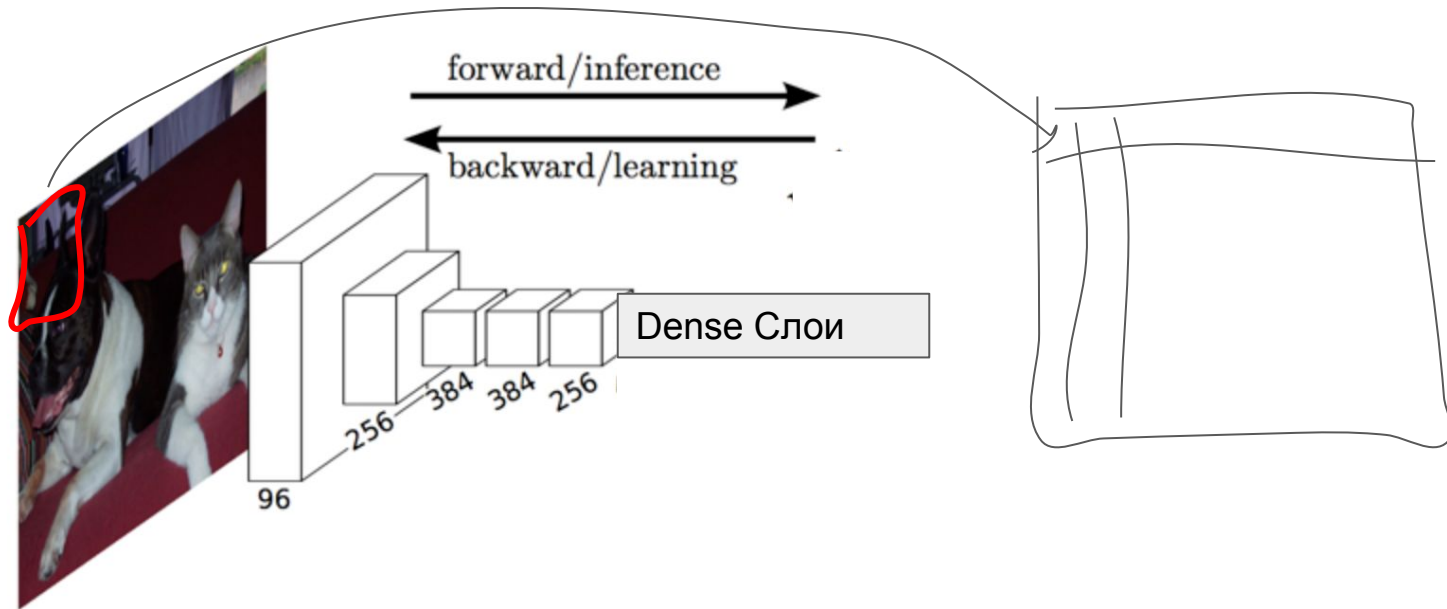

The equation $\text{IoU} = \frac{\text{Intersection}}{\text{Union}}$ is shown in large white text. To the right of the equals sign is a diagram consisting of two overlapping green squares. The top-left square is outlined in red, and the bottom-right square is outlined in cyan. The intersection of the two squares is shaded in a darker green.



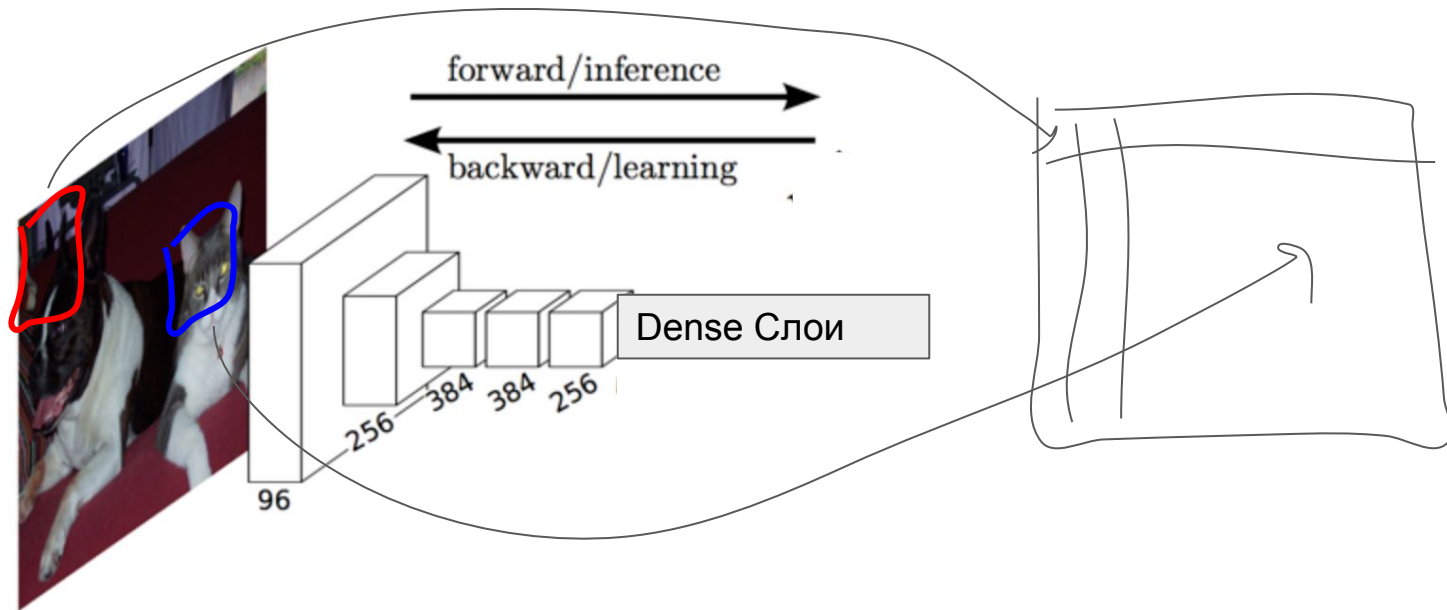
Сегментация - наивно



Сегментация - наивно

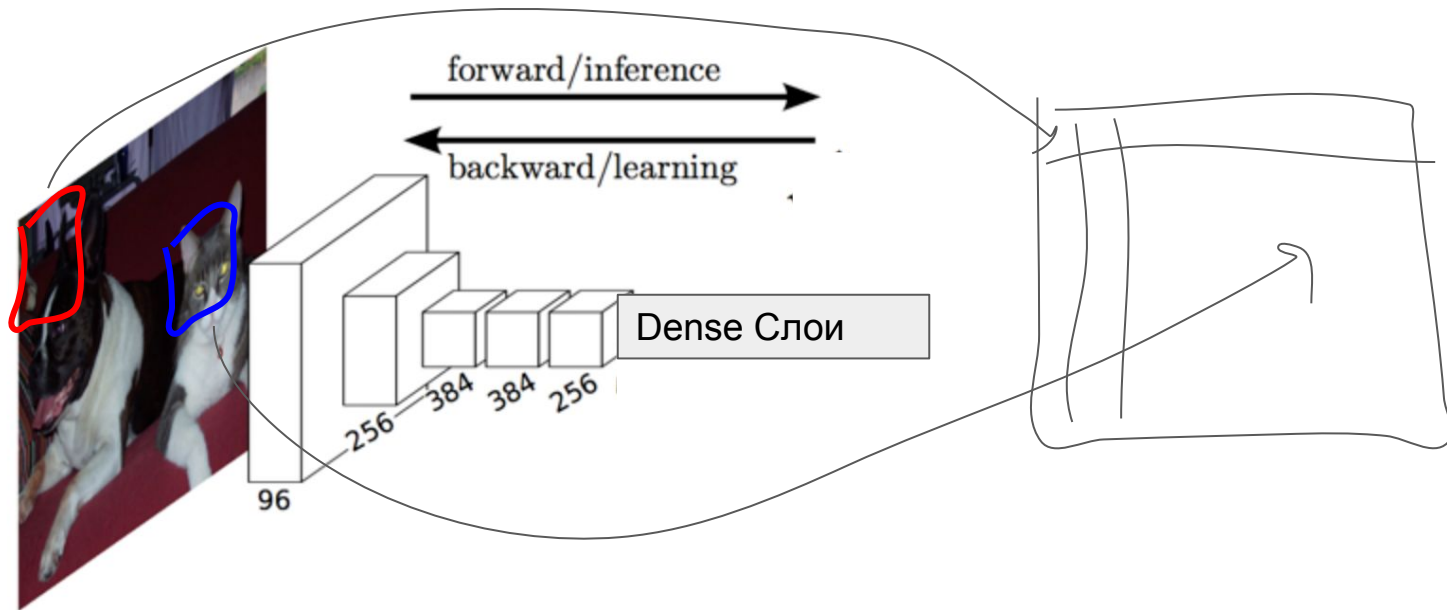


Сегментация - наивно



Медленно, не используем маски во время обучения

Сегментация - наивно

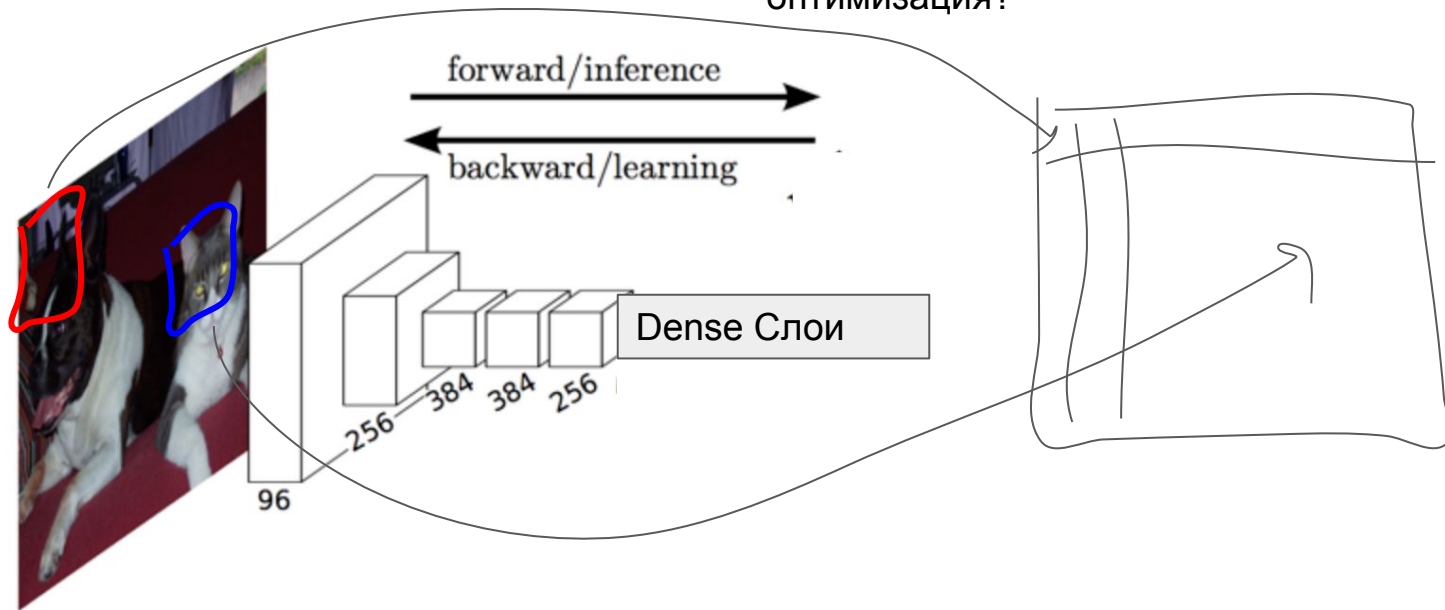


Медленно, не используем маски во время обучения

Сегментация - наивно

Для картинки 256x256, с окном 64x64 и шагом 8:

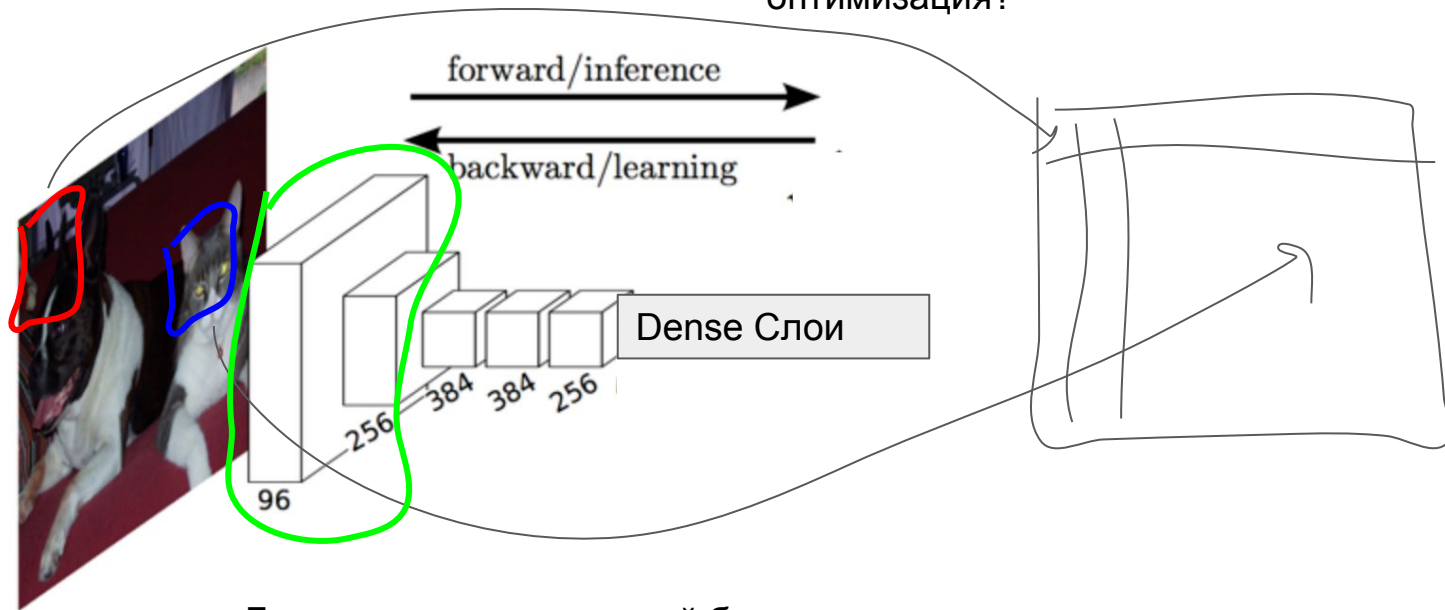
- 32*32 раз нужно прогнать инференс полностью
- результат будет 64x64 ($256/8 \times 256/8$)
- какая тут может быть мгновенная оптимизация?



Сегментация - наивно

Для картинки 256x256, с окном 64x64 и шагом 8:

- 32*32 раз нужно прогнать инференс полностью
- результат будет 64x64 ($256/8 \times 256/8$)
- какая тут может быть мгновенная оптимизация?

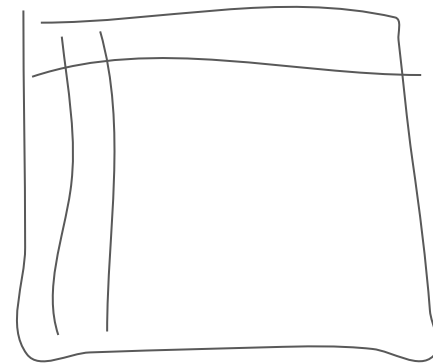
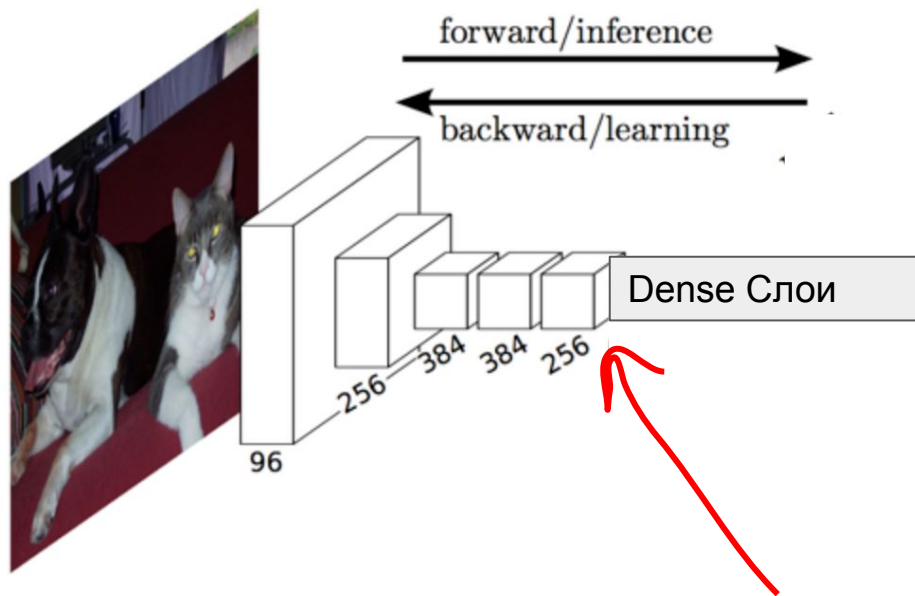


Большинство вычислений будет в conv

Сегментация - наивно

Для картинки 256x256, с окном 64x64 и шагом 8:

- 32*32 раз нужно прогнать инференс полностью
- результат будет 64x64 ($256/8 \times 256/8$)
- какая тут может быть мгновенная оптимизация?

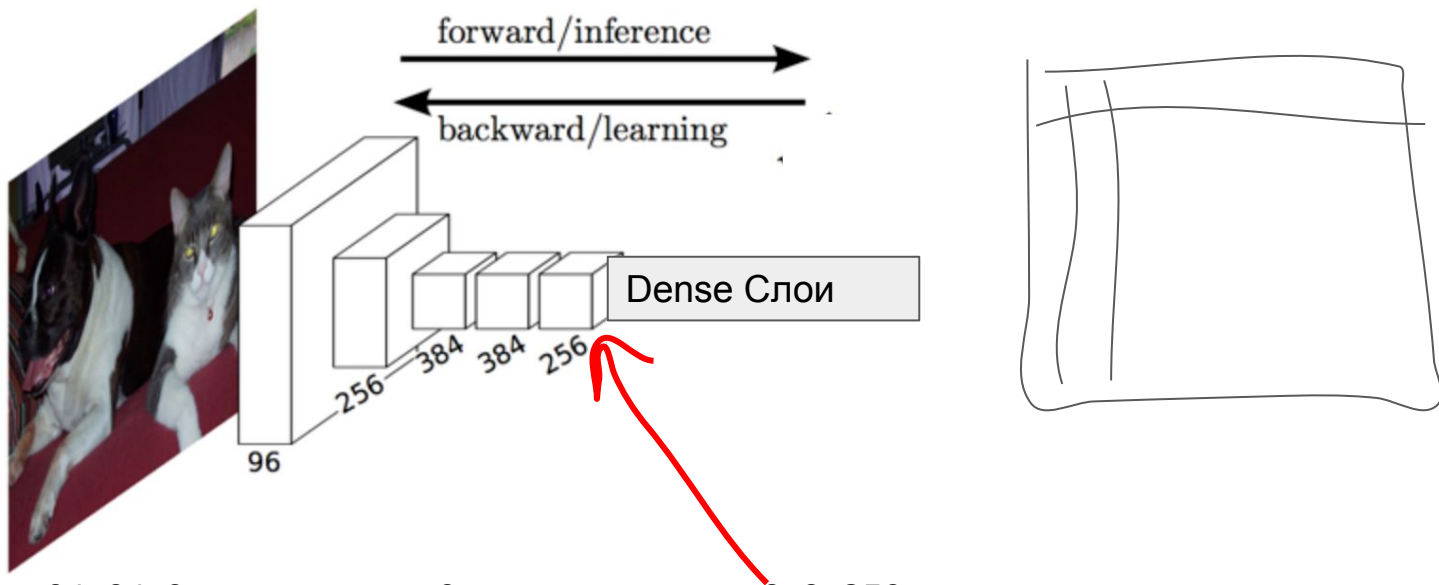


окно (вход) был 64x64x3, мы сжимаем 3 раза становится 8x8x256

Сегментация - наивно

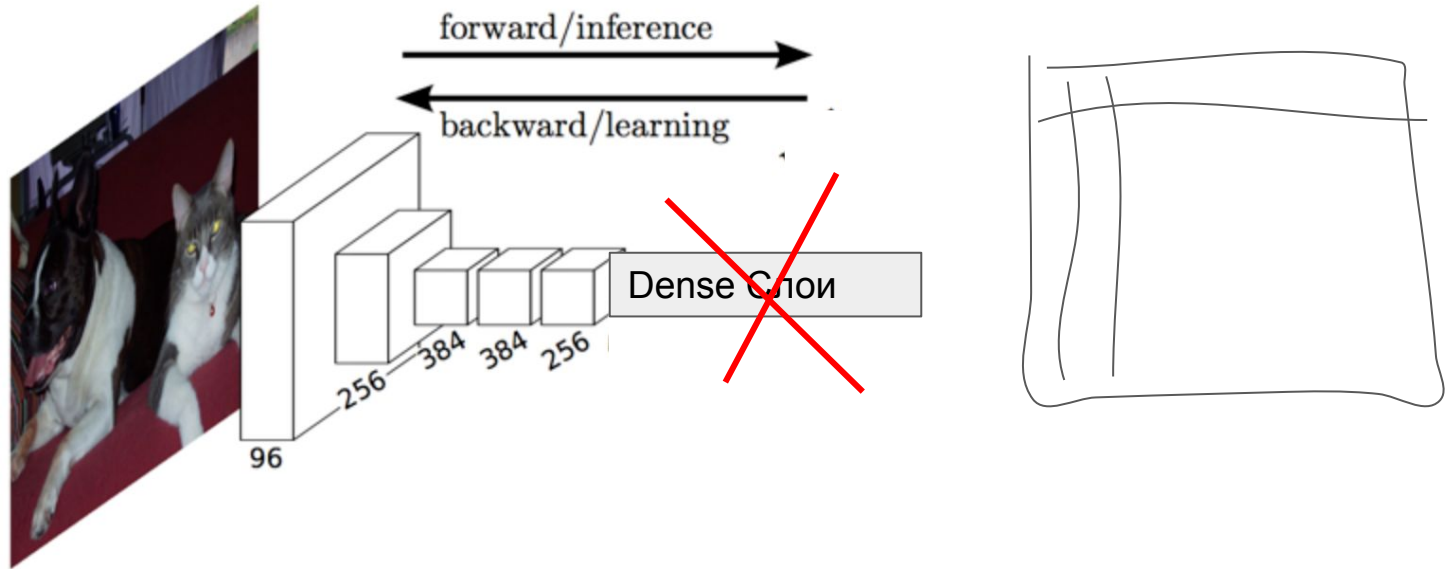
Для картинки 256x256, с окном 64x64 и шагом 8:

- 32*32 раз нужно прогнать инференс полностью
- результат будет 64x64 ($256/8 \times 256/8$)
- какая тут может быть мгновенная оптимизация?



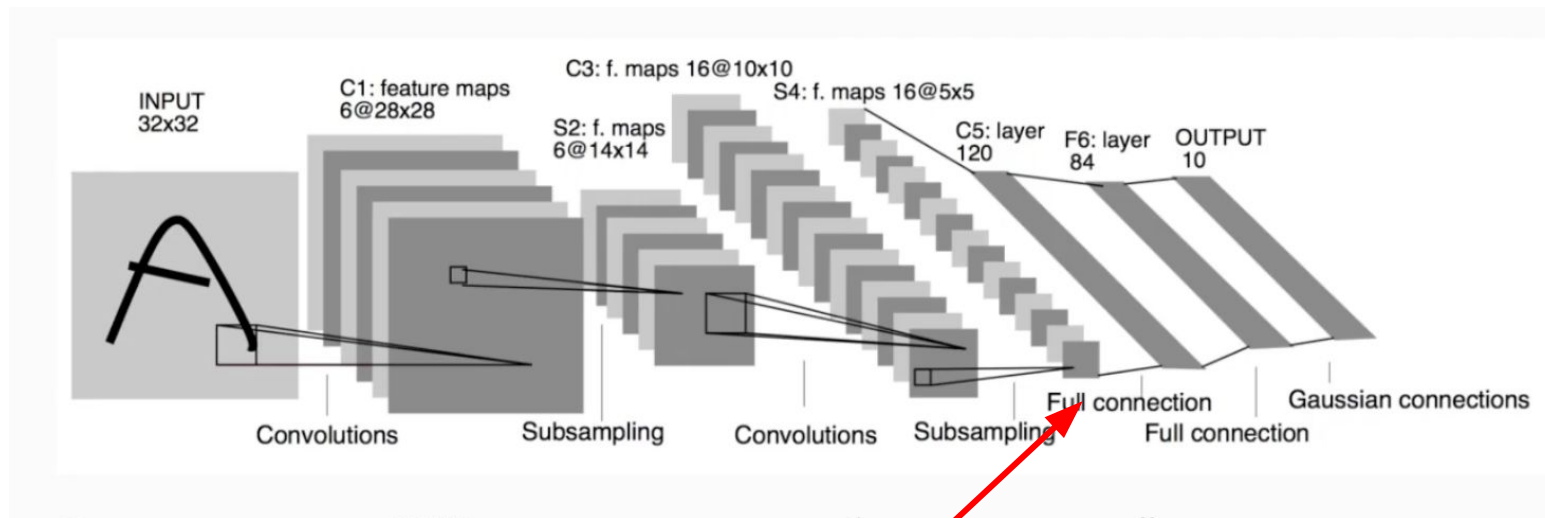
окно (вход) был 64x64x3, мы сжимаем 3 раза становится 8x8x256
conv все равно на размер входа, монжо прогнать сразу всю картинку 256x256x3 -> 32x32x256
и применить dense к каждому 8x8 окошку

Сегментация - наивно

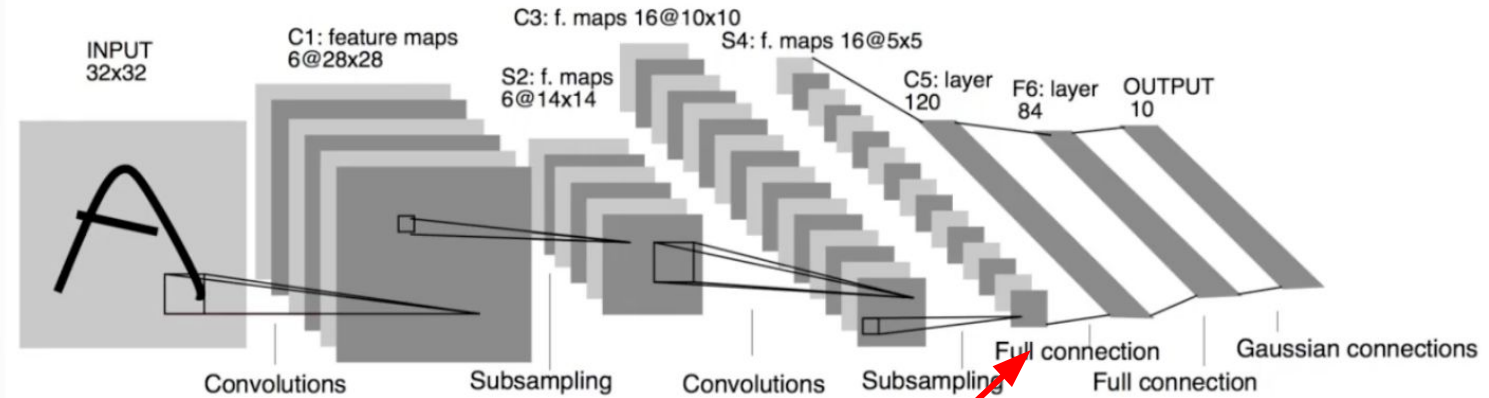


Хотим выкинуть dense слои с fix входом вообще чтобы считать за один проход

“Конволюционизация”

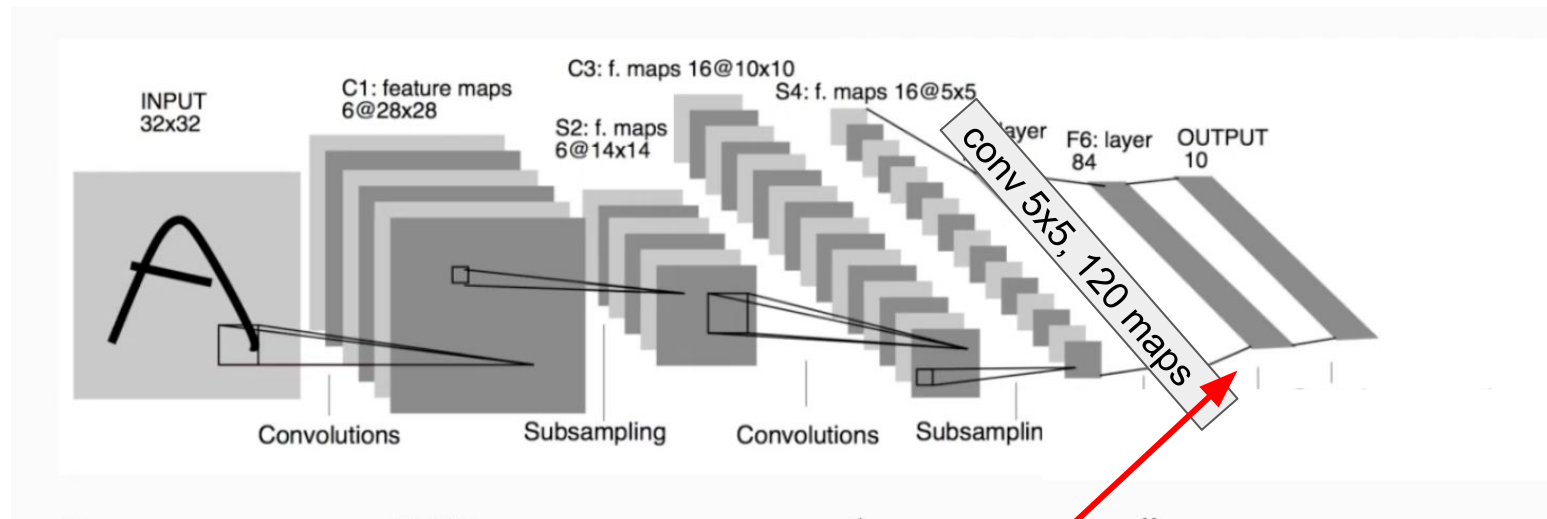


“Конволюционизация”



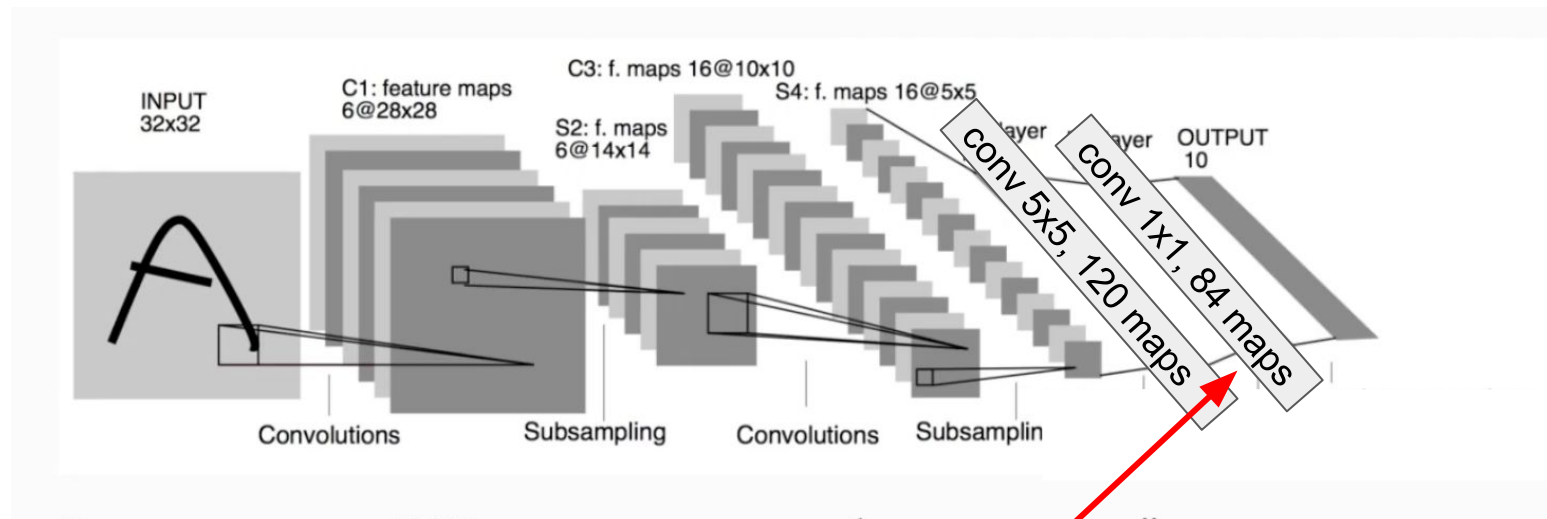
$16 \times 5 \times 5 \rightarrow 400 \times 1 \rightarrow$ dense layer $\rightarrow 120 \times 1$
в dense 400×120 параметров ;
фиксированный размер входа и
выхода

“Конволюционизация”



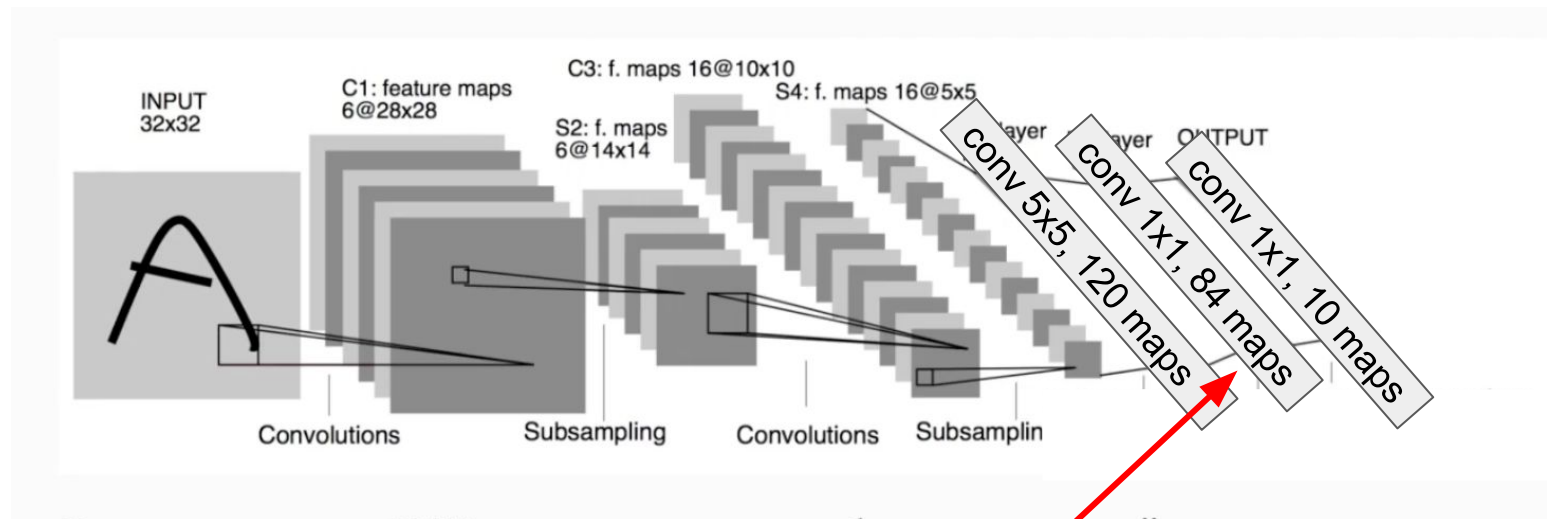
$16 \times 5 \times 5 \rightarrow 120 \times 1 \times 1 \rightarrow 120 \times 1 \rightarrow$
dense layers

“Конволюционизация”



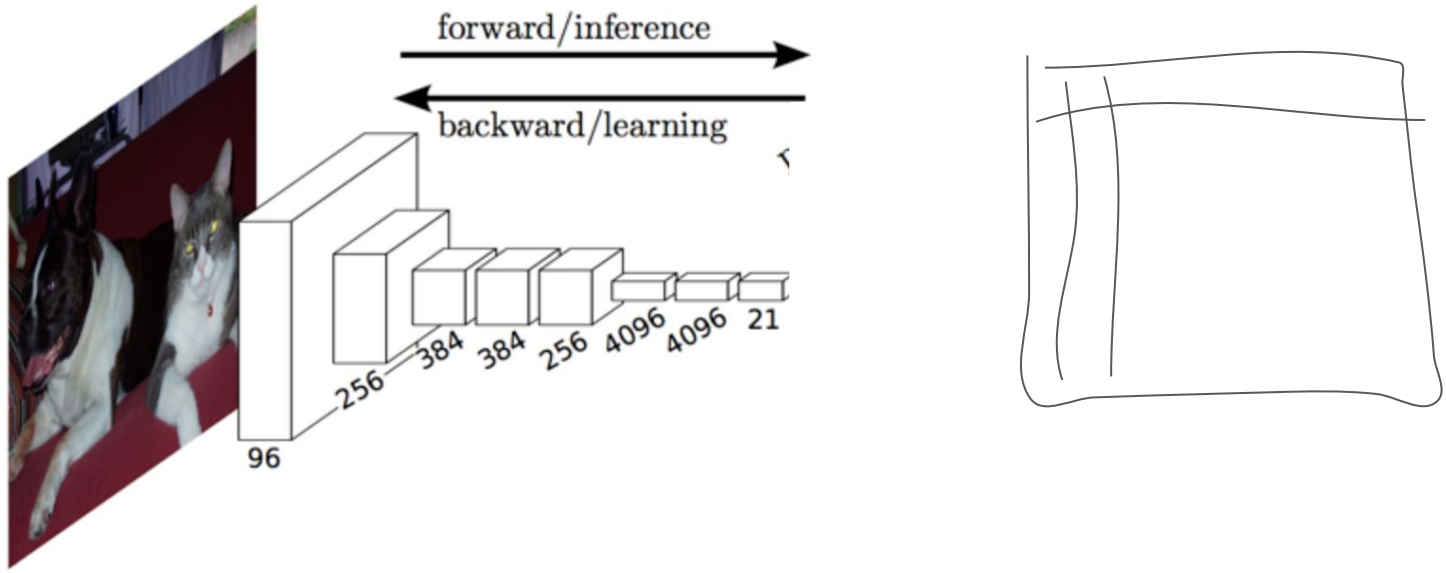
16x5x5 -> 120x1x1 -> 84x1x1 ->
dense layer

“Конволюционизация”



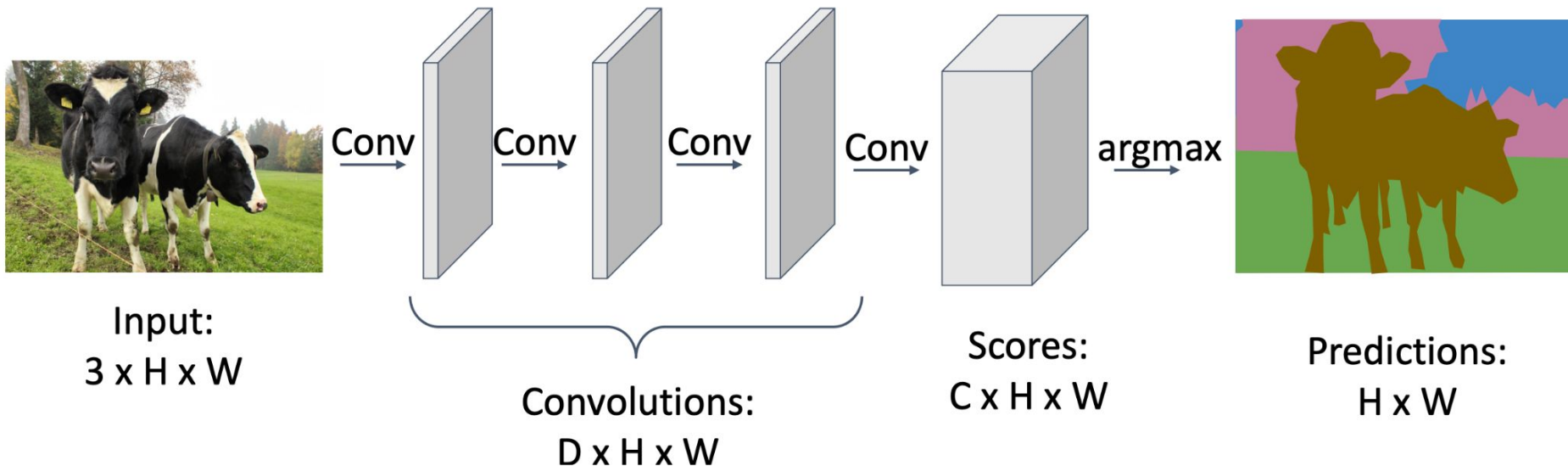
$16 \times 5 \times 5 \rightarrow 120 \times 1 \times 1 \rightarrow 84 \times 1 \times 1 \rightarrow$
 $10 \times 1 \times 1$, 10 classes classifier

Сегментация - наивно

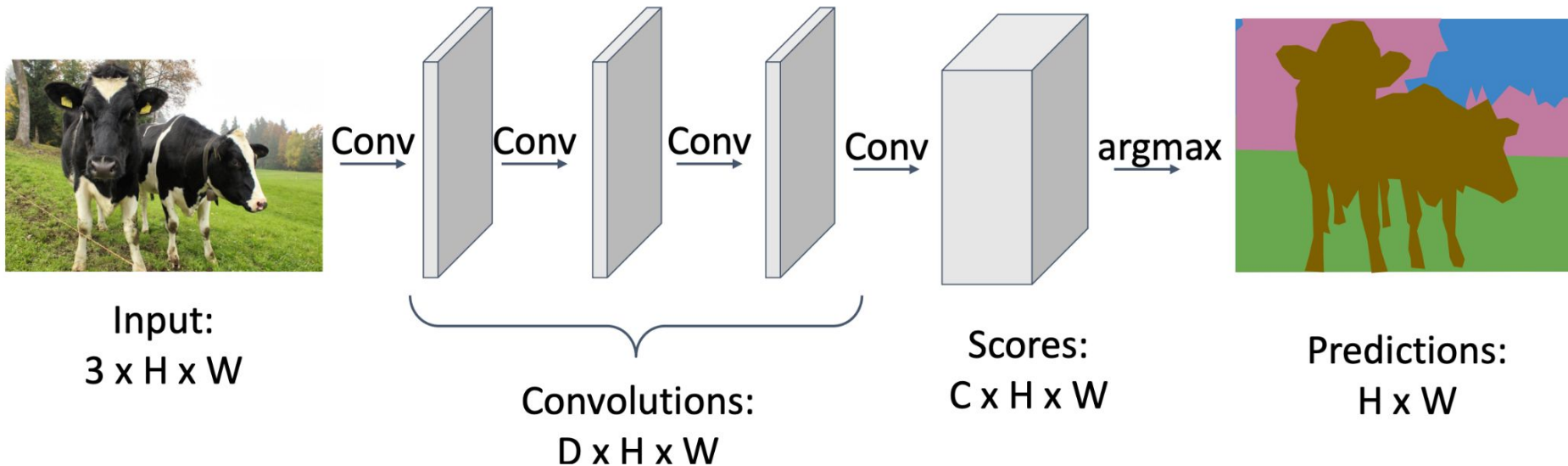


Умеем делать за 1 проход ; разрешение маски меньше чем картинки (сжимаем в сколько то раз)

Сегментация - а зачем сжимали?



Сегментация - а зачем сжимали?

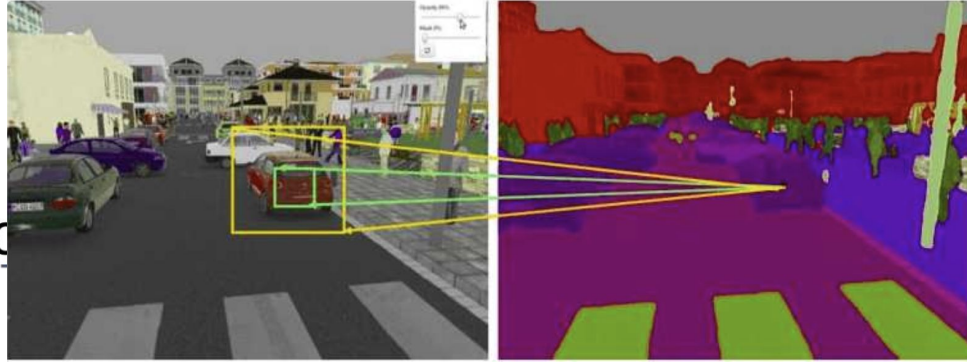


Будет ли так работать? Какие проблемы?

Сегментация - а зачем сжимали?



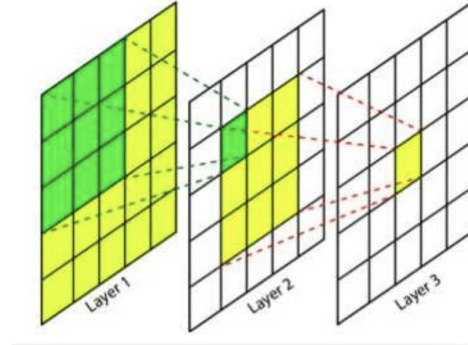
Input:
 $3 \times H \times W$



max



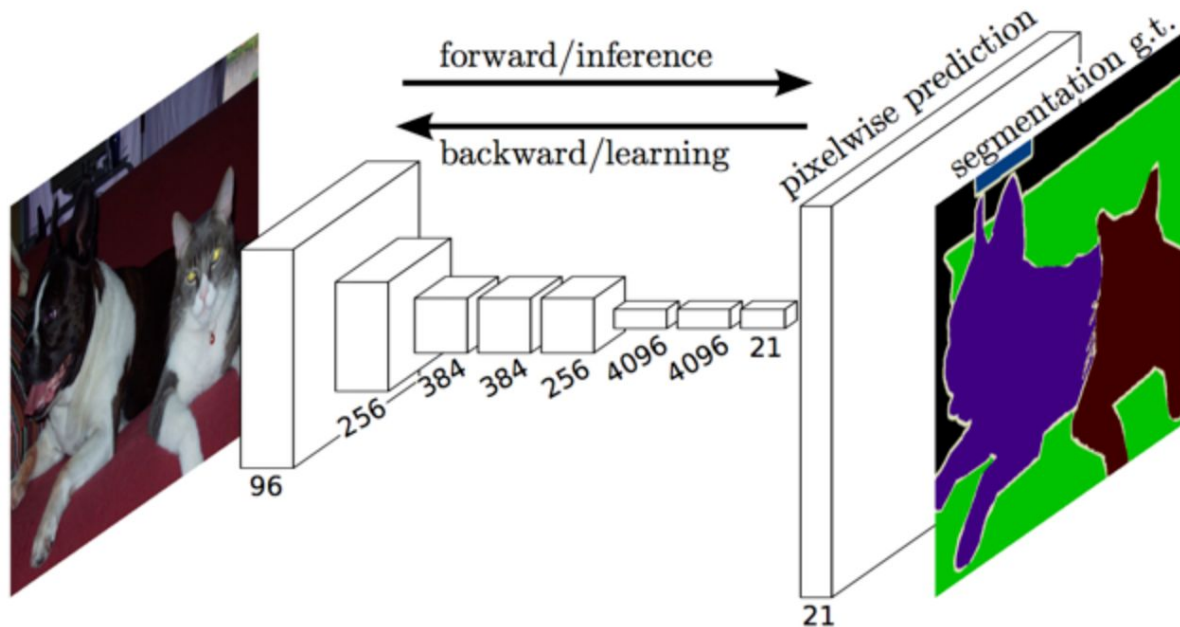
Predictions:
 $H \times W$



рецептивное поле растет примерно линейно от количества и размеров сверток, сжимая увеличиваем мультипликативно -> будем сжимать

Сегментация - сохраняем размеры

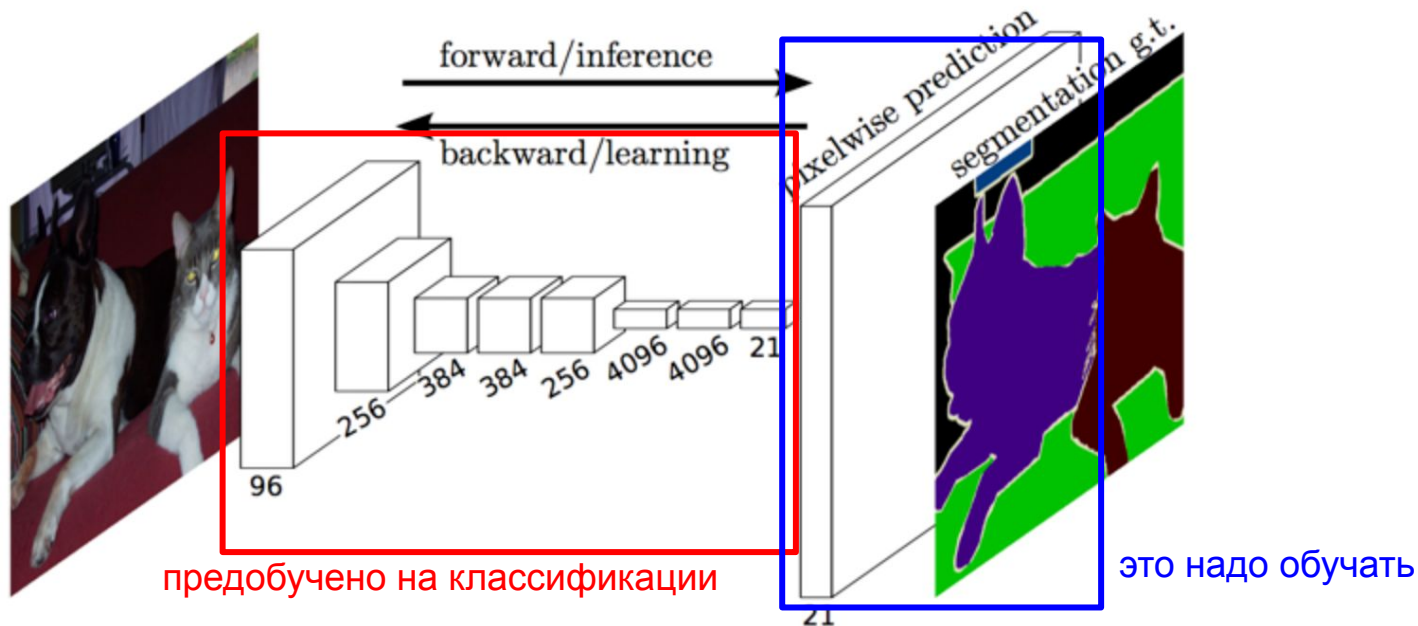
Fully Convolutional Networks for Semantic Segmentation, Shelhamer et al., 2015



Для начала попробуем просто сделать upscaling нашей карты вероятностей

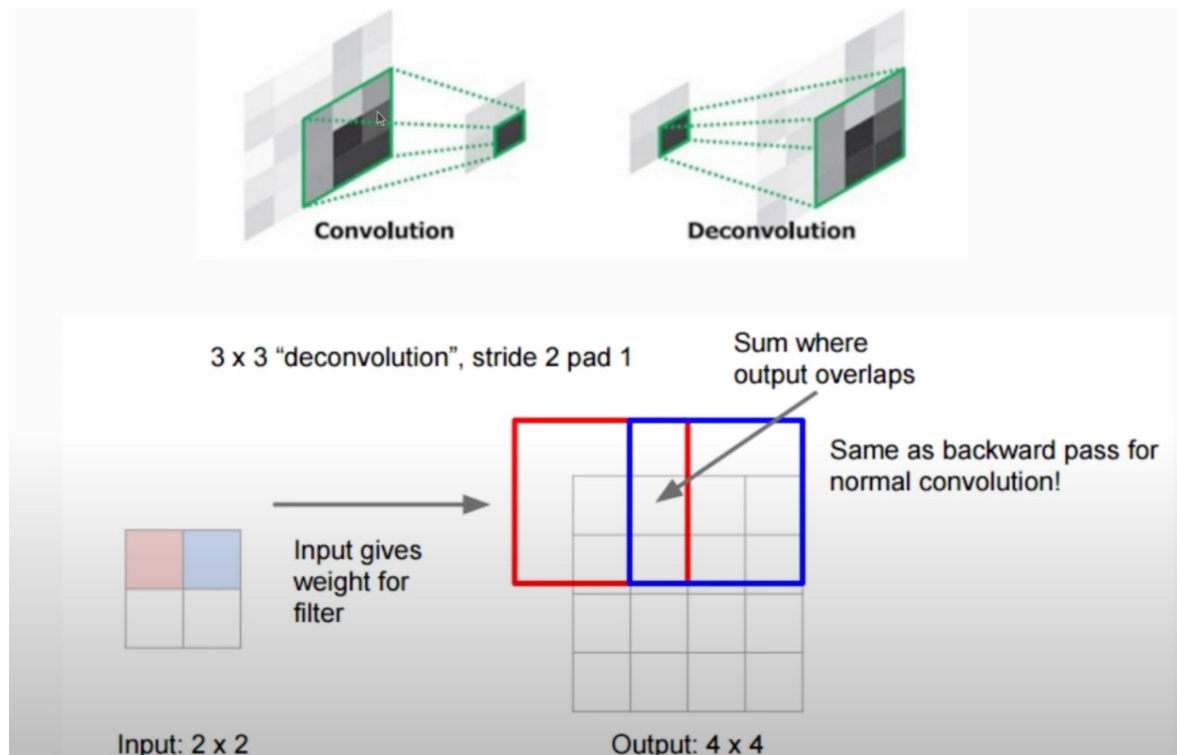
Сегментация - сохраняем размеры

Fully Convolutional Networks for Semantic Segmentation, Shelhamer et al., 2015



Для начала попробуем просто сделать upscaling нашей карты вероятностей

Как работает upsampling в FCN



Проблемы?

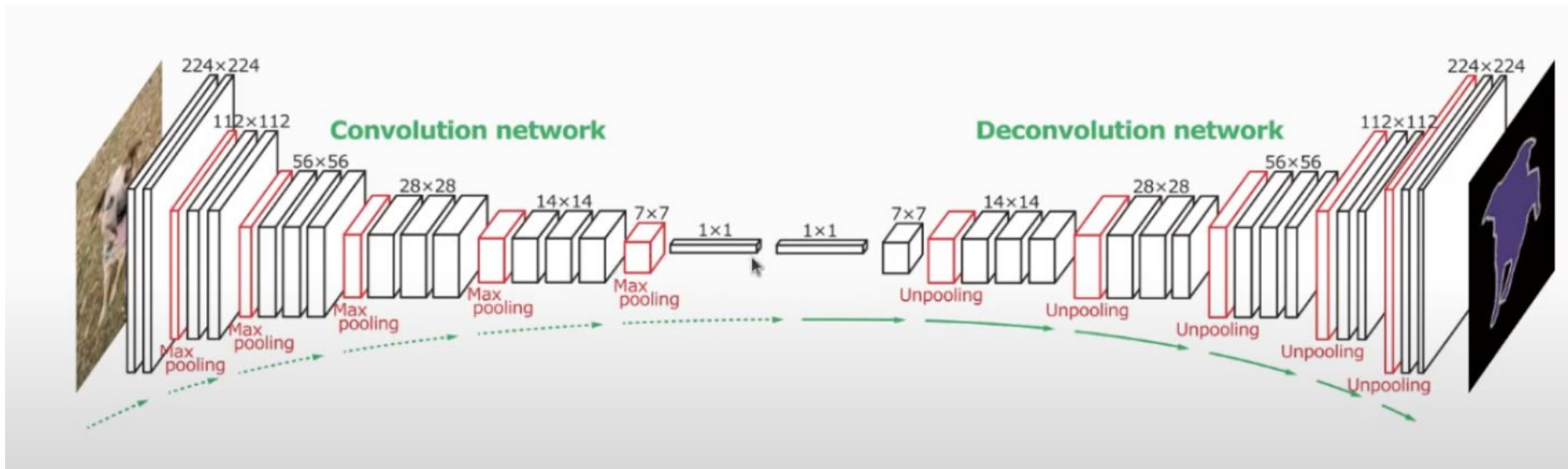


(a) Inconsistent labels due to large object size



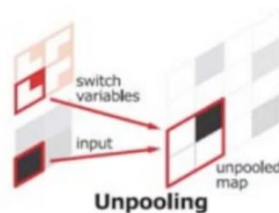
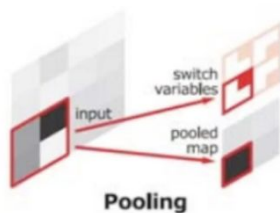
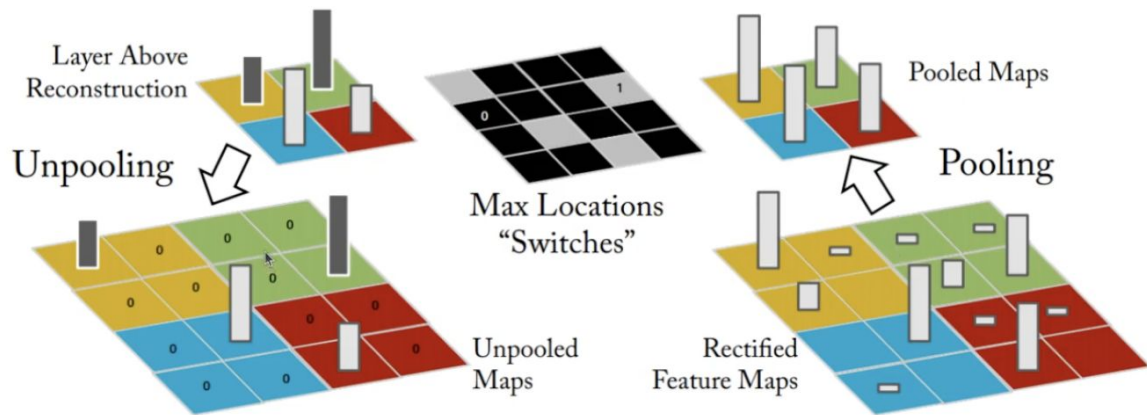
(b) Missing labels due to small object size

Давайте разжимать

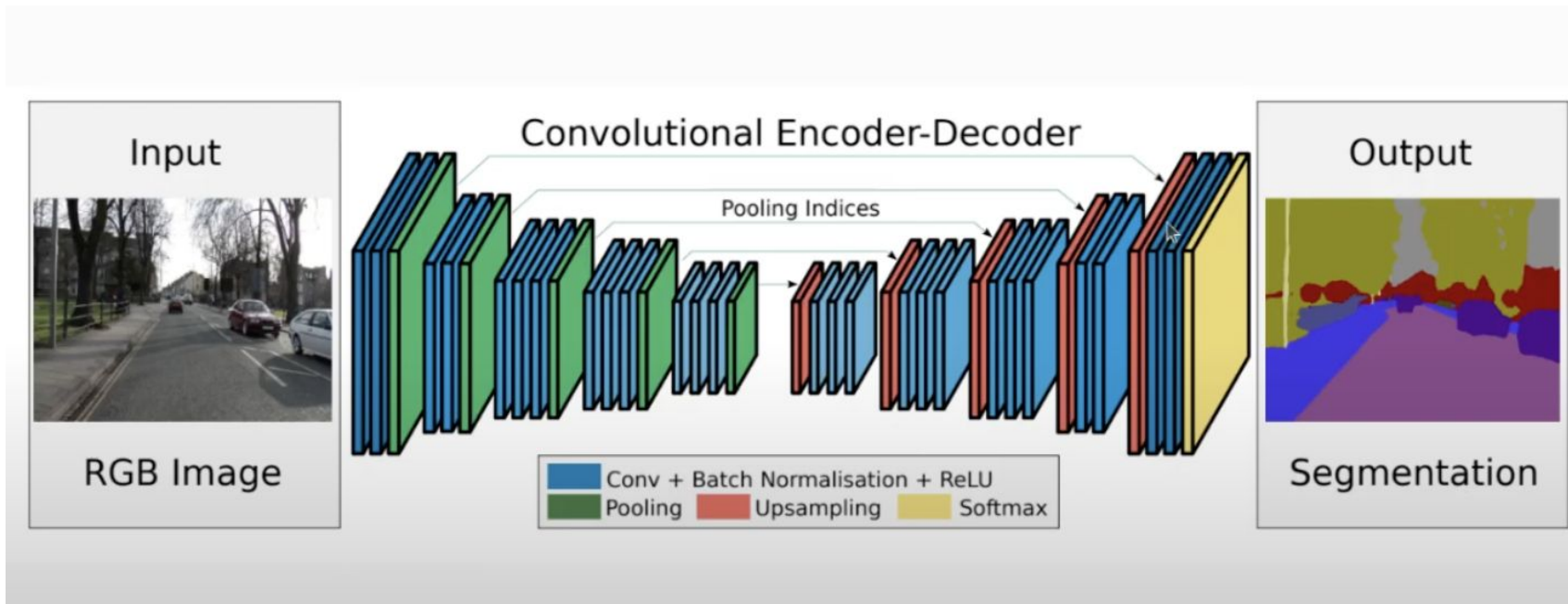


Learning Deconvolution Network for Semantic Segmentation, 2015

Давайте разжимать - unpooling

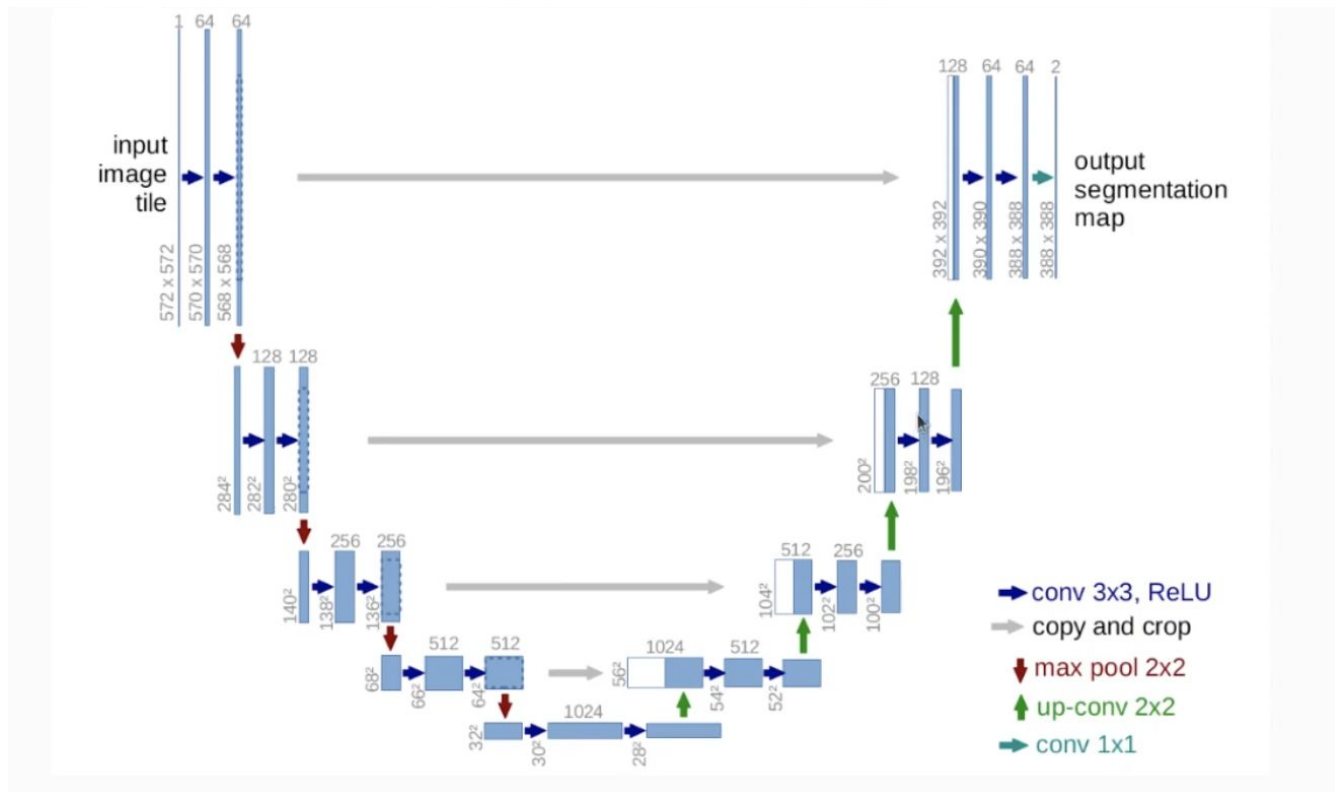


Давайте не очень сильно сжимать



segnet, 2016

UNET



ключевая идея - skip connection

Upsampling

Сейчас часто используют просто комбинацию из
upsampling+conv

Nearest Neighbor

1	2
3	4



1	1	2	2
1	1	2	2
3	3	4	4
3	3	4	4

Input: 2 x 2

Output: 4 x 4

bilinear

10	20
30	40

2x2

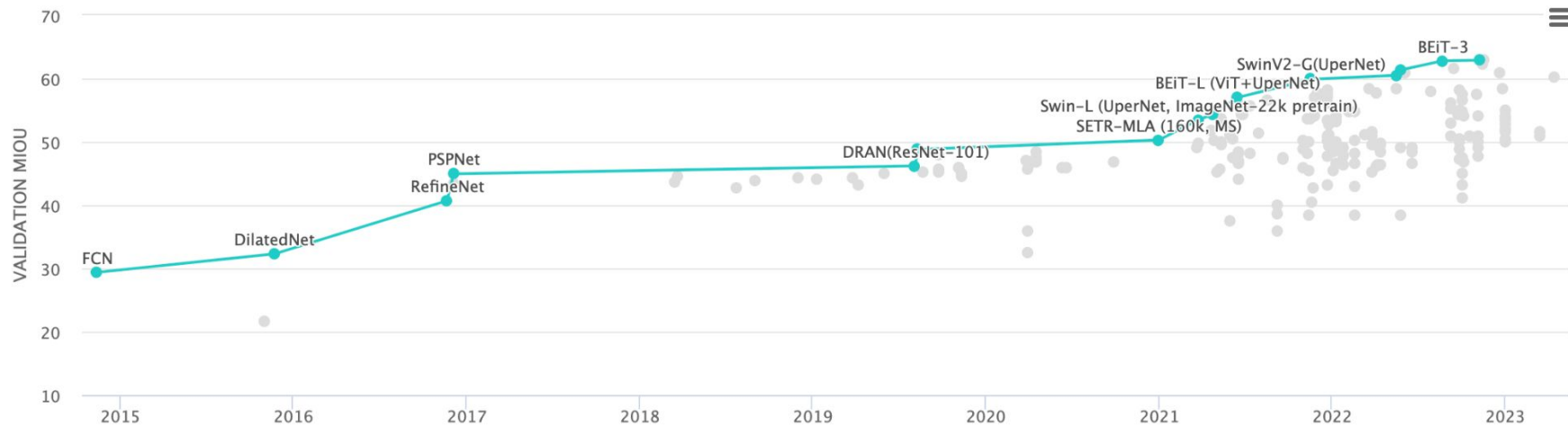
2x



10	12	17	20
15	17	22	25
25	27	32	35
30	32	37	40

4x4

Итого

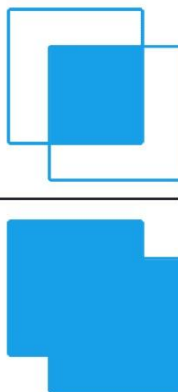


Что сейчас?

- анпулинги в таком виде особо не используются, можно просто использовать какой то upsampling + conv layer
- сегментация хорошо решается, если у вас нормальный датасет то сделать бейзлайн - практически однокнопочная задача ; потом уже в зависимости от бизнес/академ задачи нужно думать
- зоопарк бенчмарков так что сложно сравниться
- <https://storage.googleapis.com/openimages/web/index.html> если общая задача сегментировать все подряд - то можно начать с этого
- вы можете брать претренированные модели и дообучать на вашу задачу, вы можете в своей unet like архитектуре брать энкодер обученный на классификацию

сегментация - лосс и проблемы

- кросентропия (попикселная класификация)
- может быть довольно unbalanced
- в какой то момент вы умеете хорошо решать задачу так что важны становятся границы - а оно не отбалансированно
- iou как метрика - больше похоже на то что мы хотим
- дифф версии iou этого - можно добавлять в лосс


$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Аугментации

Original image



augmentation



Horizontal Flip



Crop



Median Blur



Contrast



Hue / Saturation / Value



Gamma



TestTime аугментации



Все еще кот так что можно агрегировать, всегда (когда имеет смысл)
может улучшить результат, но дорого

Ссылки

- в части про архитектуры многие картинки из курса по compvision michigan university <https://www.youtube.com/watch?v=XaZlIVrIO-Q&t=3282s>
- картинки из лекций Сергей Николенко <https://www.youtube.com/watch?v=xhifn1K8sxM&t=1549s>
- если есть вопросы можно мне написать в телеграмме @vladgl