

Семантическая сегментация

Semantic segmentation



Задача семантической сегментации картинок

Задача

Пусть X — пространство картинок.

Y — набор классов, например {человек, машина, дорога, тротуар, знак ПДД}.

\hat{X} — пространство картинок, где каждый пиксель имеет значение из Y .
Требуется построить модель $f: X \rightarrow \hat{X}$, определяющую, к какому классу из Y принадлежит каждый пиксель изображения X .



Лоссы и метрики качества

Предсказание сегментации

Задача бинарной сегментации: сегментируем картинку на два класса 0 и 1

Предсказание сети:
Картинка 3x3x1

0.5	0	0.4
0.1	0.9	0.1
0.6	0.2	0.5

Истинная разметка:
Картинка 3x3x1

1	0	1
0	1	0
1	0	1

Предсказание сегментации: LogLoss

Задача бинарной сегментации: сегментируем картинку на два класса 0 и 1

Предсказание сети:
Картинка 3x3x1

0.5	0	0.4
0.1	0.9	0.1
0.6	0.2	0.5

Истинная разметка:
Картинка 3x3x1

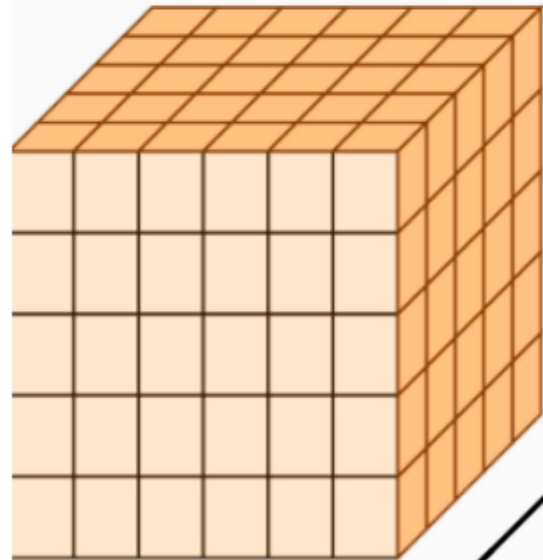
1	0	1
0	1	0
1	0	1

Попиксельный LogLoss

Предсказание сегментации

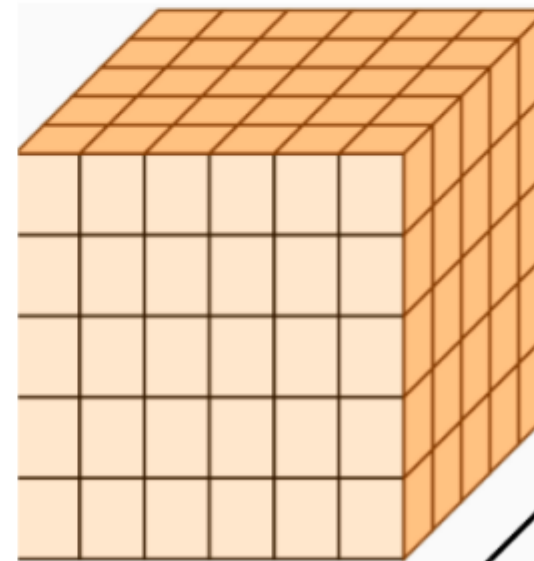
Задача **многоклассовой** сегментации:
сегментируем картинку на два класса 0 и 1

Предсказание сети:
Картинка 6x6x5



Кол-во классов

Истинная разметка:
Картинка 6x6x5

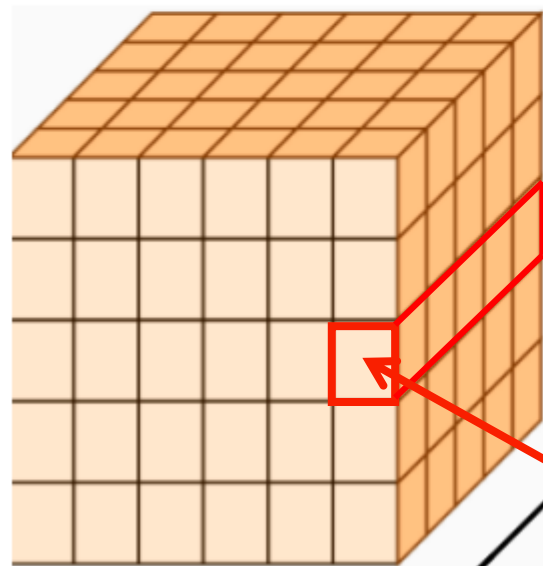


Кол-во классов

Предсказание сегментации: CrossEntropy Loss

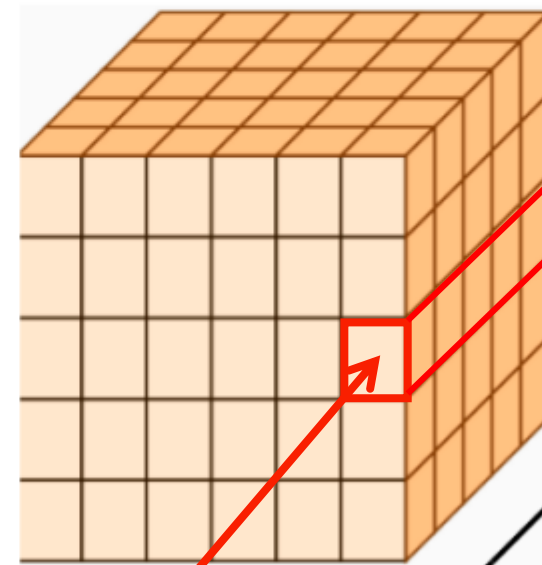
Задача **многоклассовой** сегментации:
сегментируем картинку на два класса 0 и 1

Предсказание сети:
Картинка 6x6x5



Кол-во классов

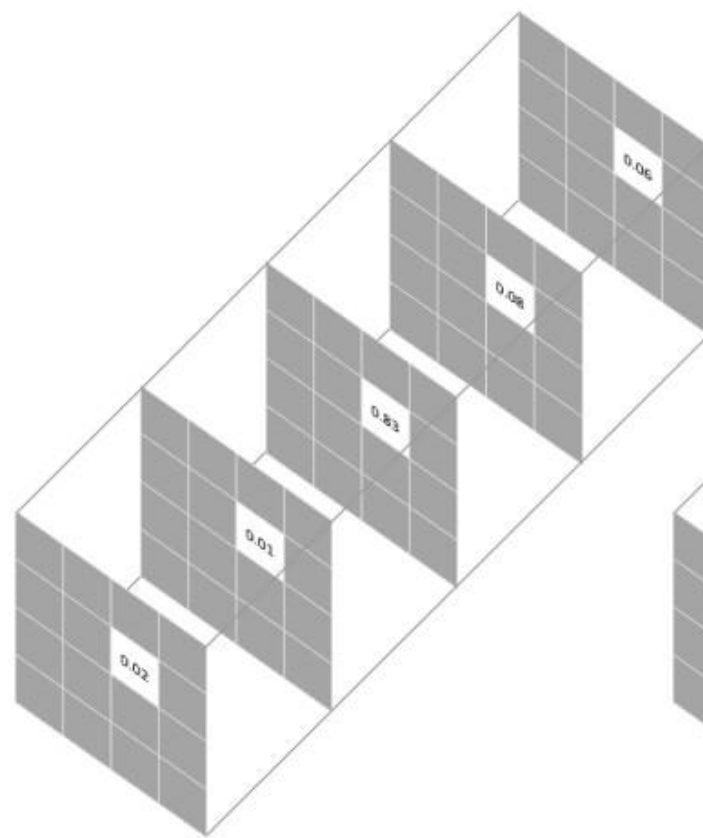
Истинная разметка:
Картинка 6x6x5



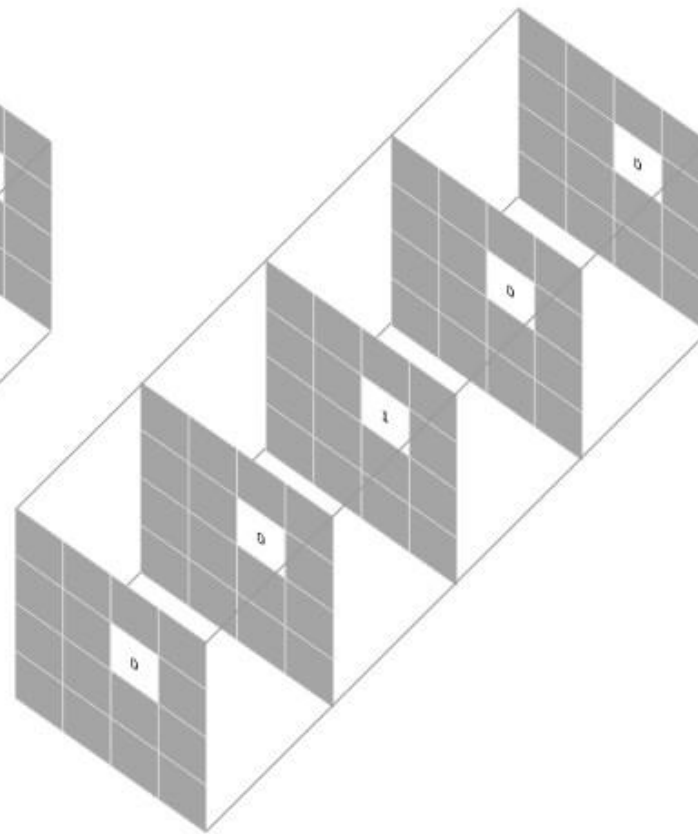
Кол-во классов

Попиксельный CrossEntropy Loss

Функция потерь CrossEntropy



Prediction for a selected pixel



Target for the corresponding pixel

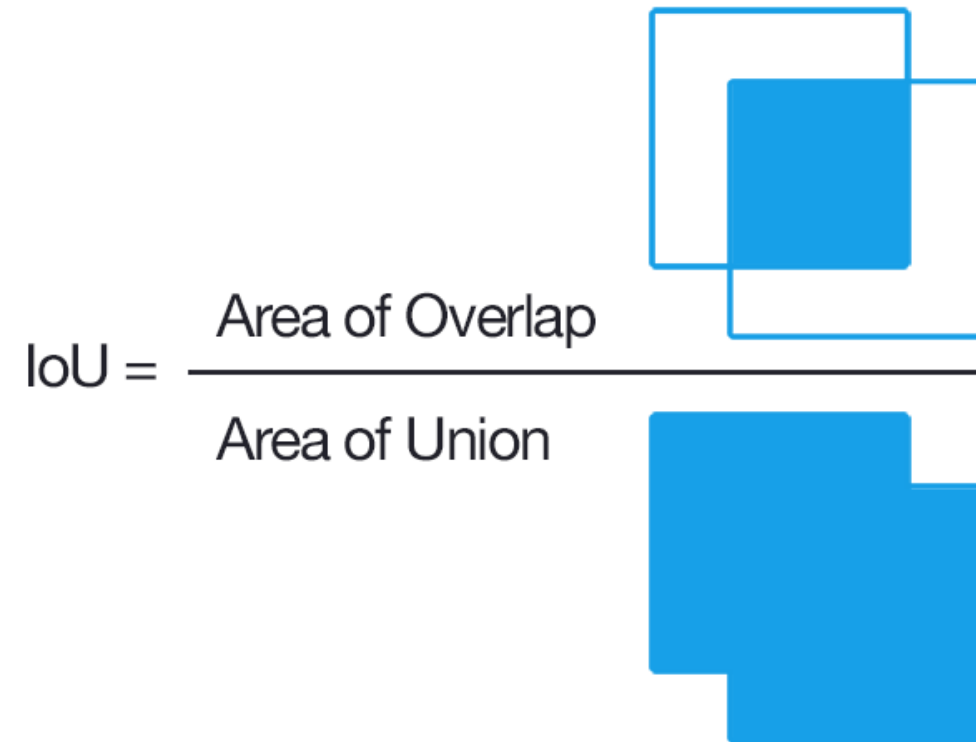
Pixel-wise loss is calculated as the log loss, summed over all possible classes

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

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

Метрики качества: Intersection over Union

$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

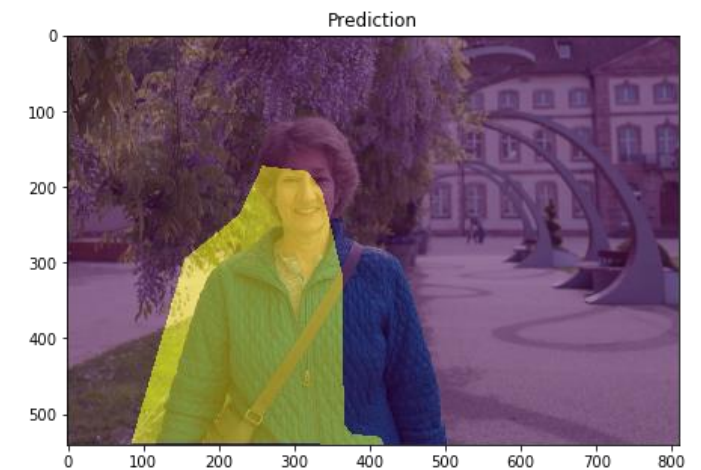


Метрики качества: Intersection over Union

$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

В отличие от Dice, IoU обычно считают для предсказания, которое уже приведено к маске со значениями $\{0, 1\}$.

Маска получается путем ограничения значений по threshold.

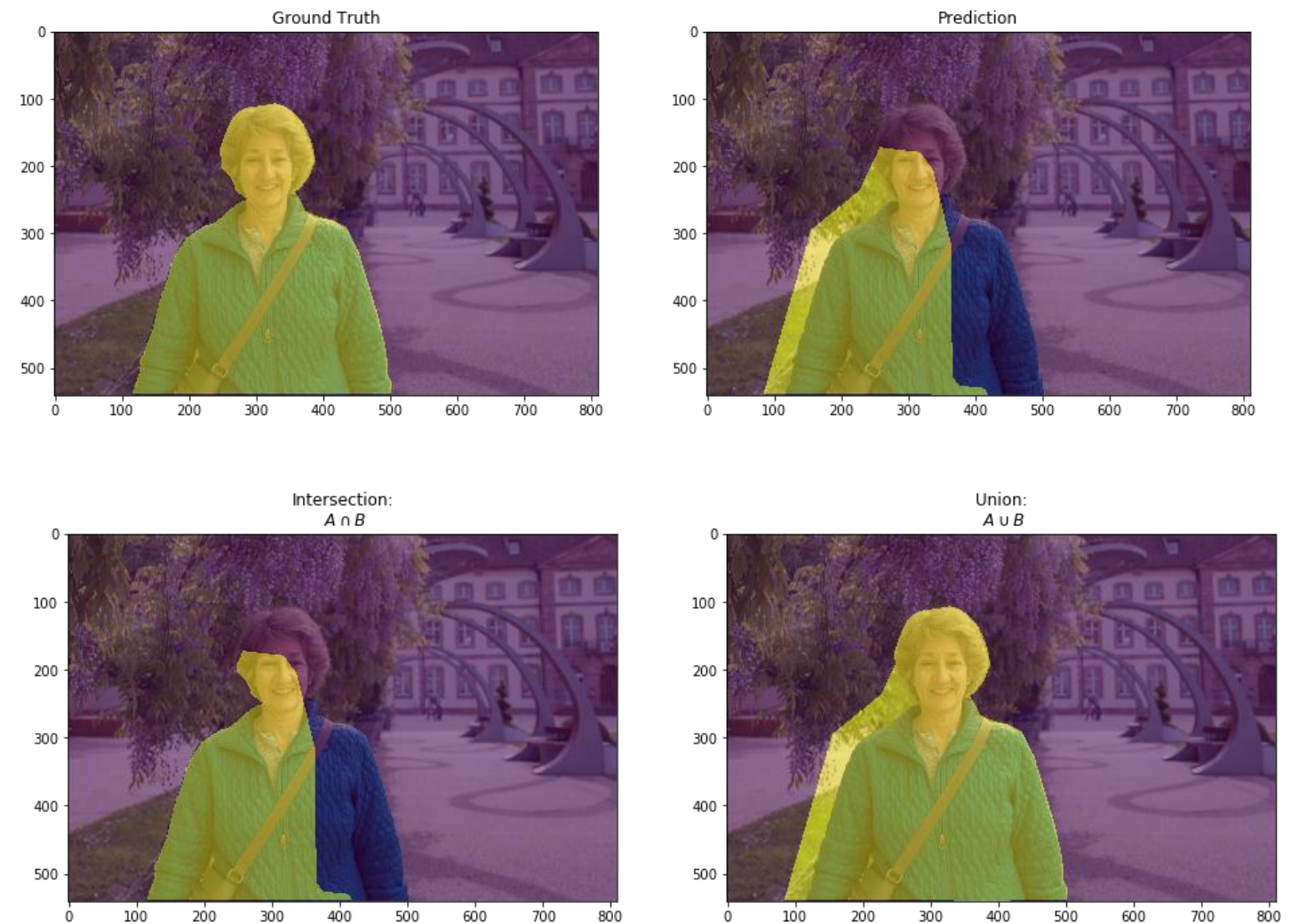


Метрики качества: Intersection over Union

$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

В отличие от Dice, IoU обычно считают для предсказания, которое уже приведено к маске со значениями $\{0, 1\}$.

Маска получается путем ограничения значений по threshold.



Построение модели

Задача семантической сегментации картинок

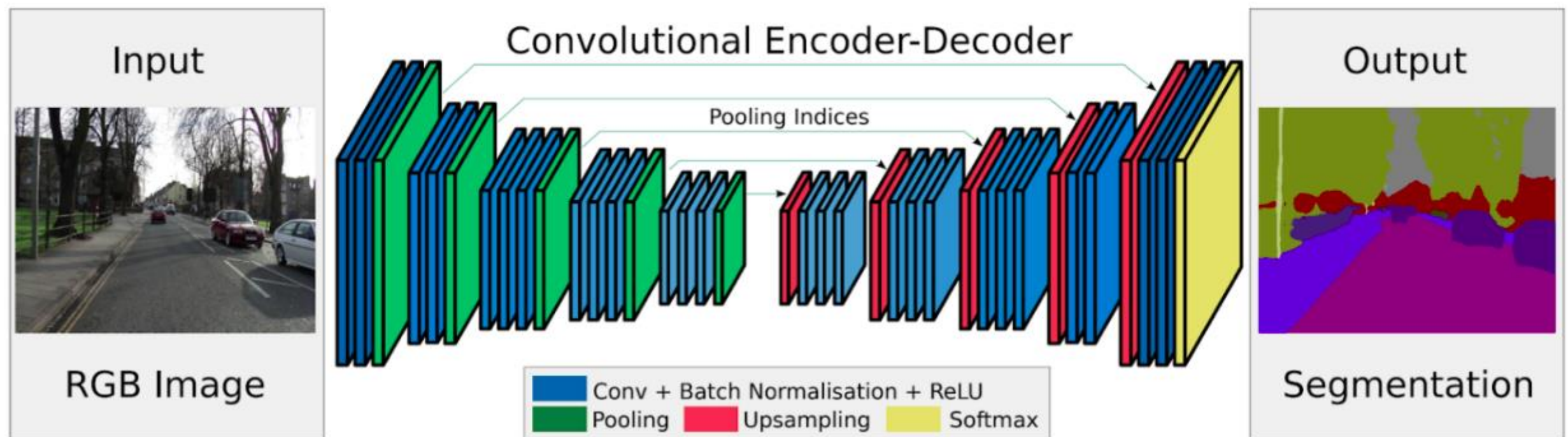
Задача

Требуется построить модель $f: X \rightarrow \hat{X}$, определяющую, к какому классу из Y принадлежит каждый пиксель изображения X .

Большинство моделей семант. сегментации устроено по принципу “encoder-decoder”.

Encoder — это Feature Extractor-часть любой сети для классификации.

Как делать decoder? Он должен разворачивать тензор признаков в картинку.

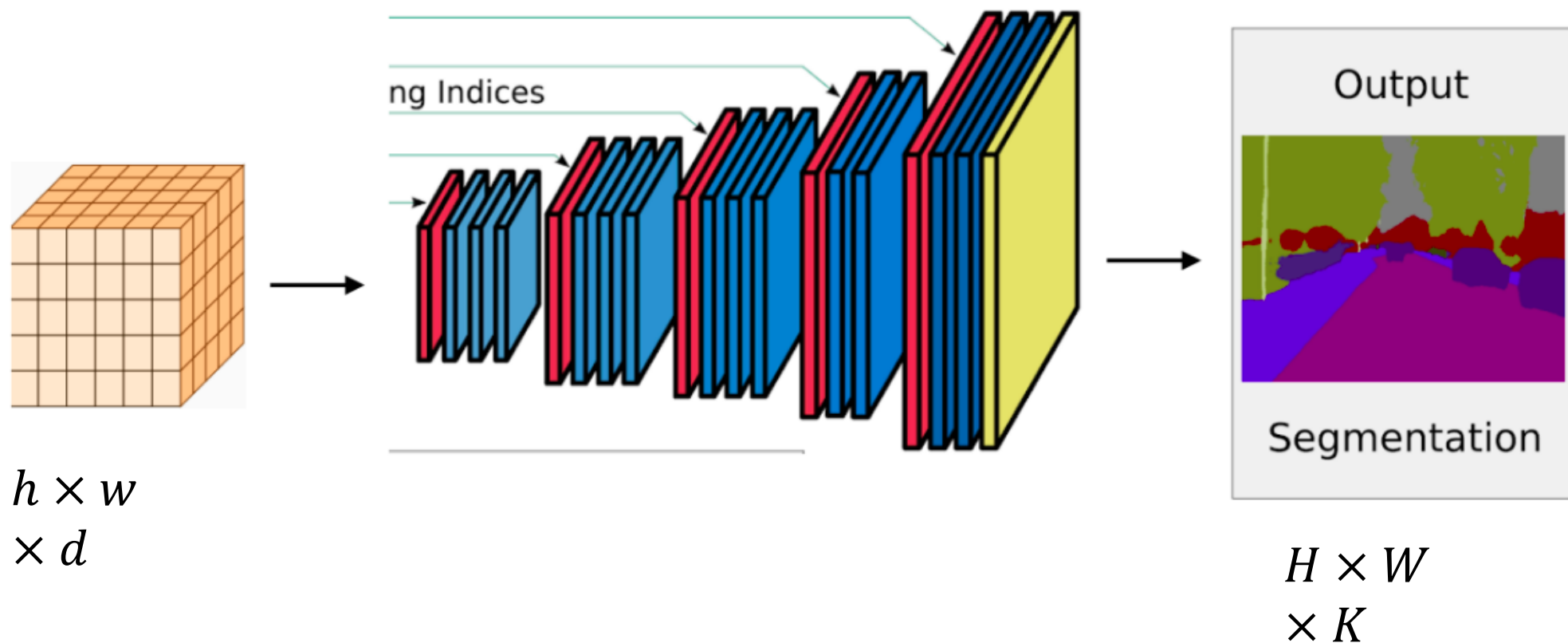


Семантическая сегментация: Decoder

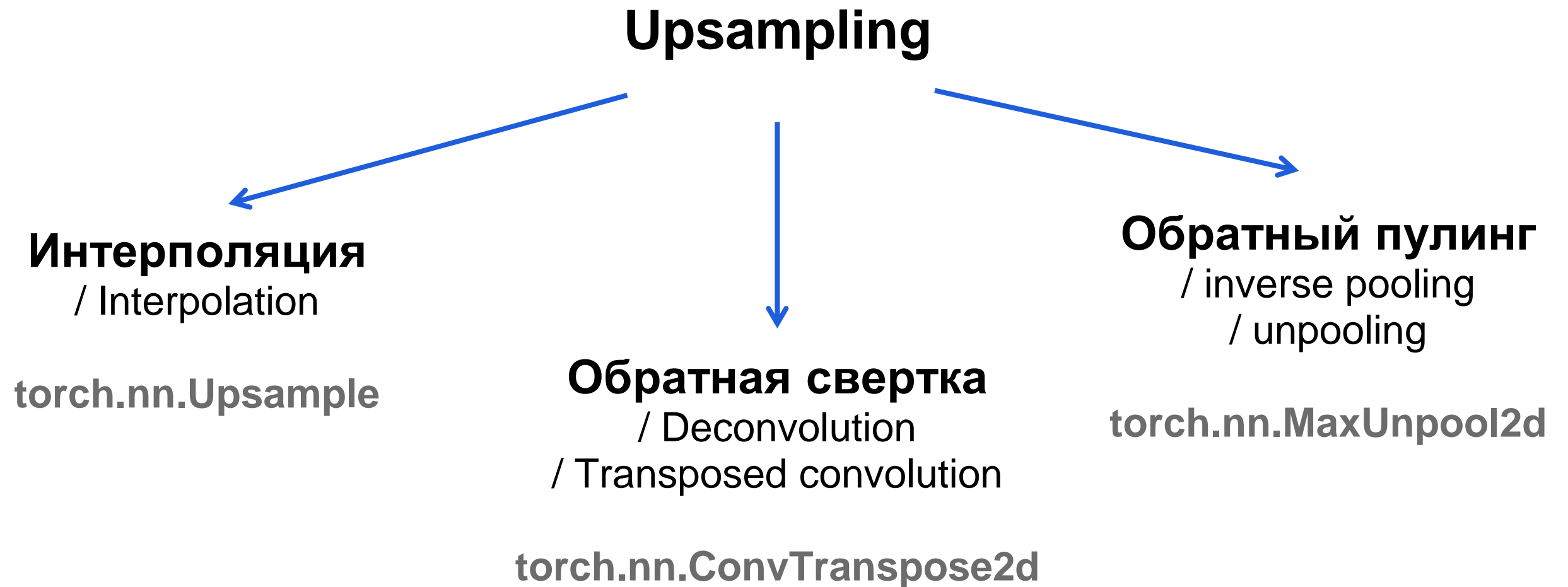
Как делать decoder?

Нужна операция, которая из тензора размера $h \times w \times d$, получает тензор размер $H \times W \times K$, где $h < H$, $w < W$, H и W высота и ширина исходной картинки, K — количество классов.

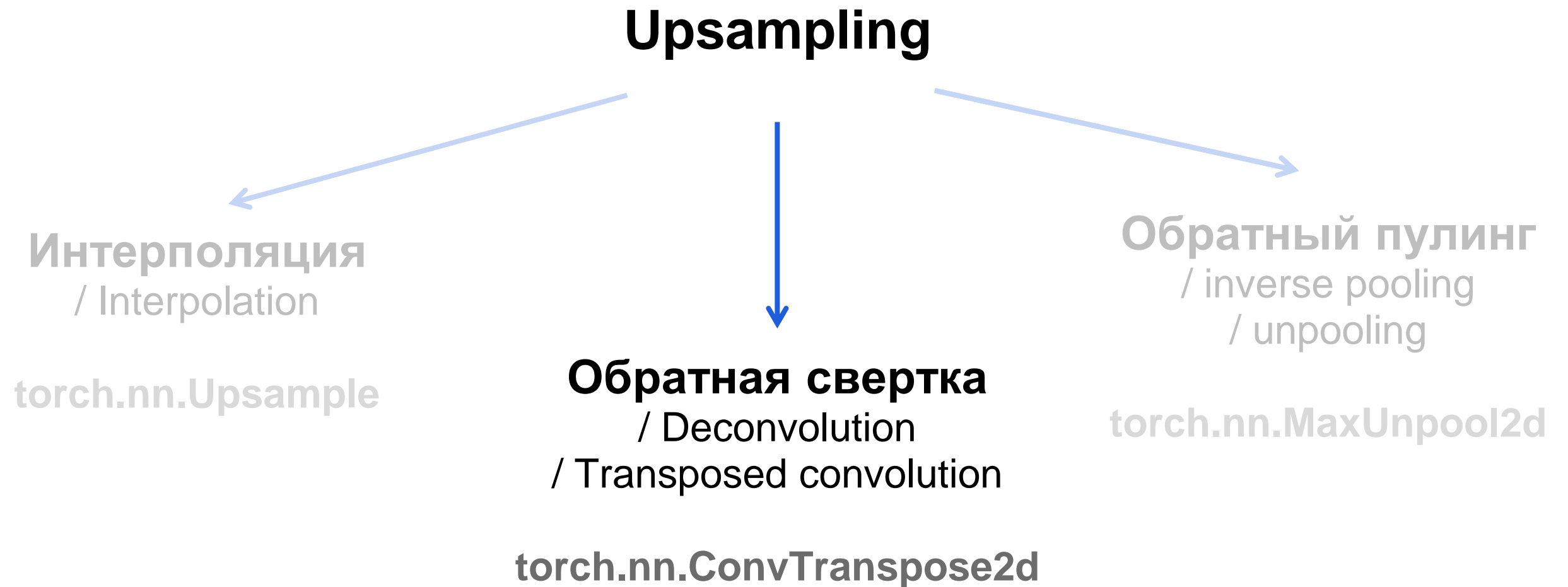
Такая операция называется **Upsampling**.



Виды Upsampling

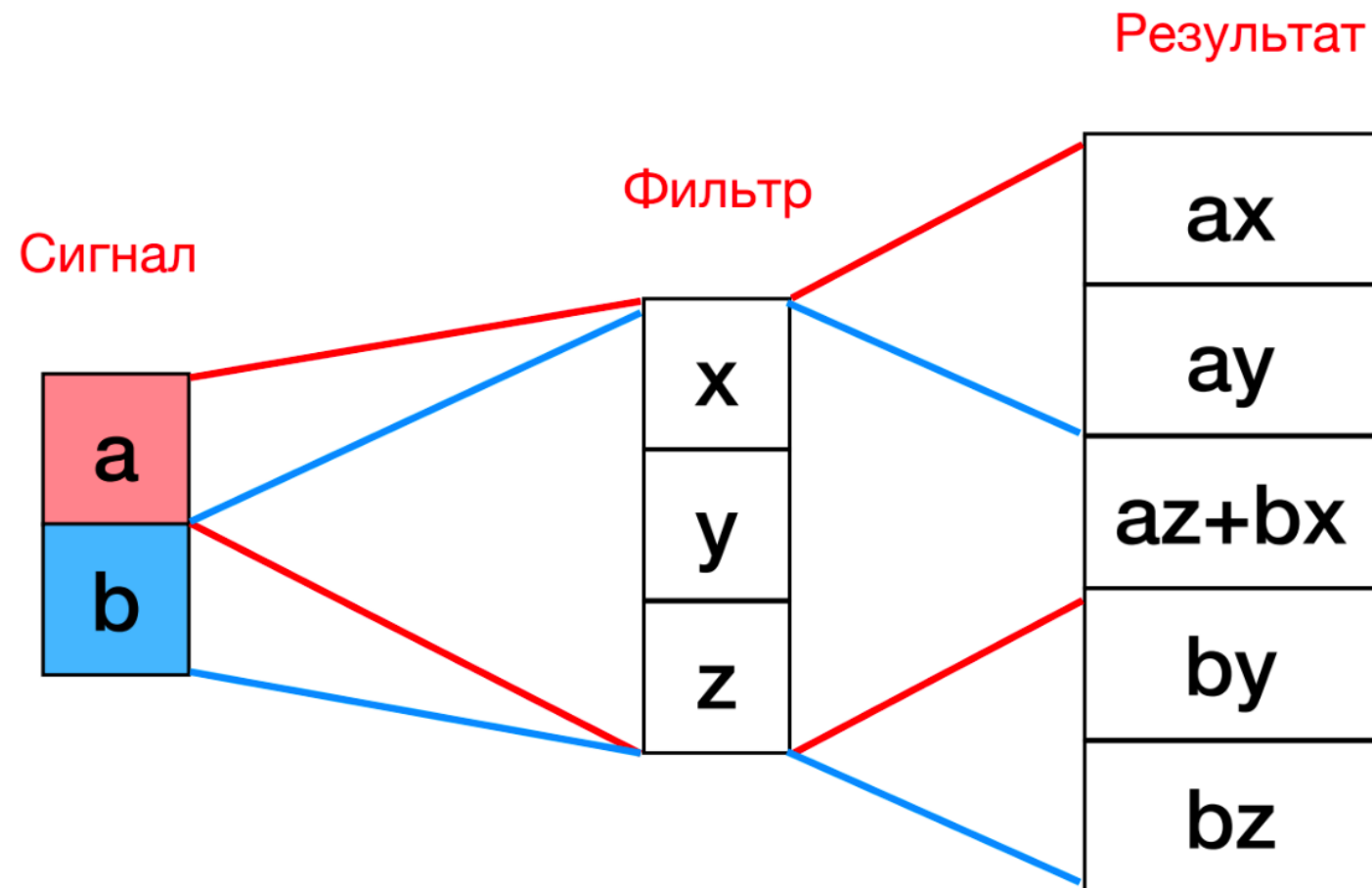


Виды Upsampling



Виды Upsampling: обратная свертка

Двумерный случай



Виды Upsampling: обратная свертка

- Операцию обратной свертки можно представить в виде линейной операции, по аналогии с обычной сверткой.

Значит, она **обучаема!**

Padding и stride в обратной свертке можно интерпретировать по-разному.

Рассмотрим то, как они **представлены в pytorch**.

- **Padding** —
используется для приведения к нужному размеру.
Удаляем из полученного feature map нейроны так,
чтобы если бы применили свертку (не обратную)
с такими же параметрами padding и stride к новому feature map,
то получили бы feature map такого же размера как исходное
изображение.
- **Stride** — шаг, с которым результаты сверток накладываются друг на друга.
Чем больше stride, тем больше feature map на выходе.

Виды Upsampling: обратная свертка

ВХОД

ядро

stride=1

input

2	4
0	1

k

3	1
1	5

s=1, p=0, выход 3x3

s=1

6	2	-
2	10	-
-	-	-

+

-	12	4
-	4	20
-	-	-

+

-	-	-
0	0	-
0	0	-

+

-	-	-
-	3	1
-	1	5

=

6	14	4
2	17	21
0	1	5

s=1, p=1, выход 1x1

s=1 p=1

6	2	-
2	10	-
-	-	-

+

-	12	4
-	4	20
-	-	-

+

-	-	-
0	0	-
0	0	-

+

-	-	-
-	3	1
-	1	5

=

6	14	4
2	17	21
0	1	5

Виды Upsampling: обратная свертка

ВХОД

ядро

stride=2

input

2	4
0	1

k

3	1
1	5

s=2, p=0, выход 4x4

s=2

6	2	-	-	+	-	-	12	4	+	-	-	-	-	+	-	-	-	-	=	6	2	12	4
2	10	-	-	+	-	-	4	20	+	-	-	-	-	+	-	-	-	-	=	2	10	4	20
-	-	-	-	+	-	-	-	-	+	0	0	-	-	+	-	-	3	1	=	0	0	3	1
-	-	-	-	+	-	-	-	-	+	0	0	-	-	+	-	-	1	5	=	0	0	1	5

s=1, p=1, выход 3x3

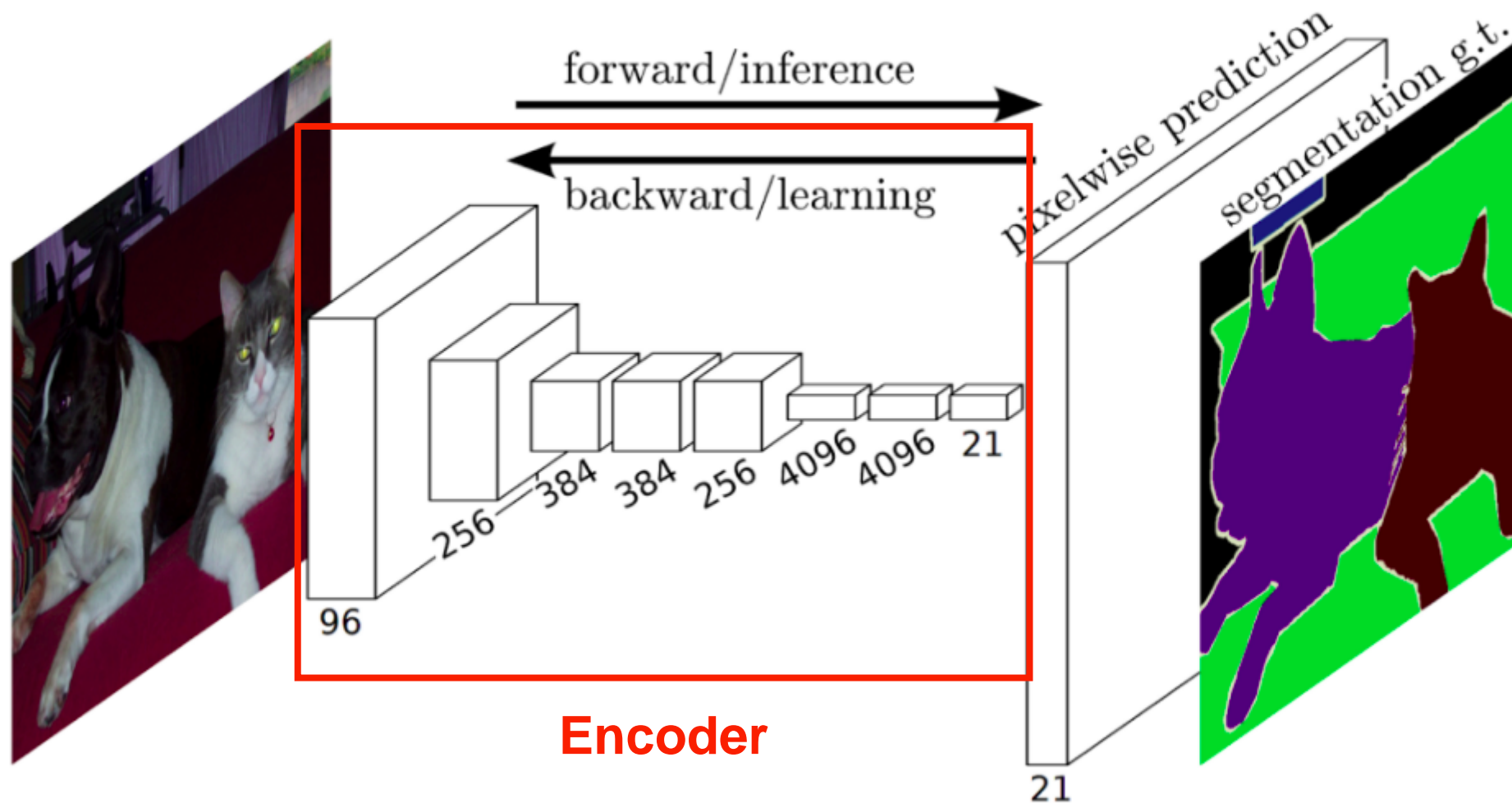
s=2 p=1

6	2	-	-		-	-	12	4		-	-	-	-		6	2	12	4
2	10	-	-		-	-	-	4	20		-	-	-		2	10	4	20
-	-	-	-		-	-	-	-	-		0	0	-	-	-	0	0	-
-	-	-	-		-	-	-	-	-		0	0	-	-	-	-	1	5

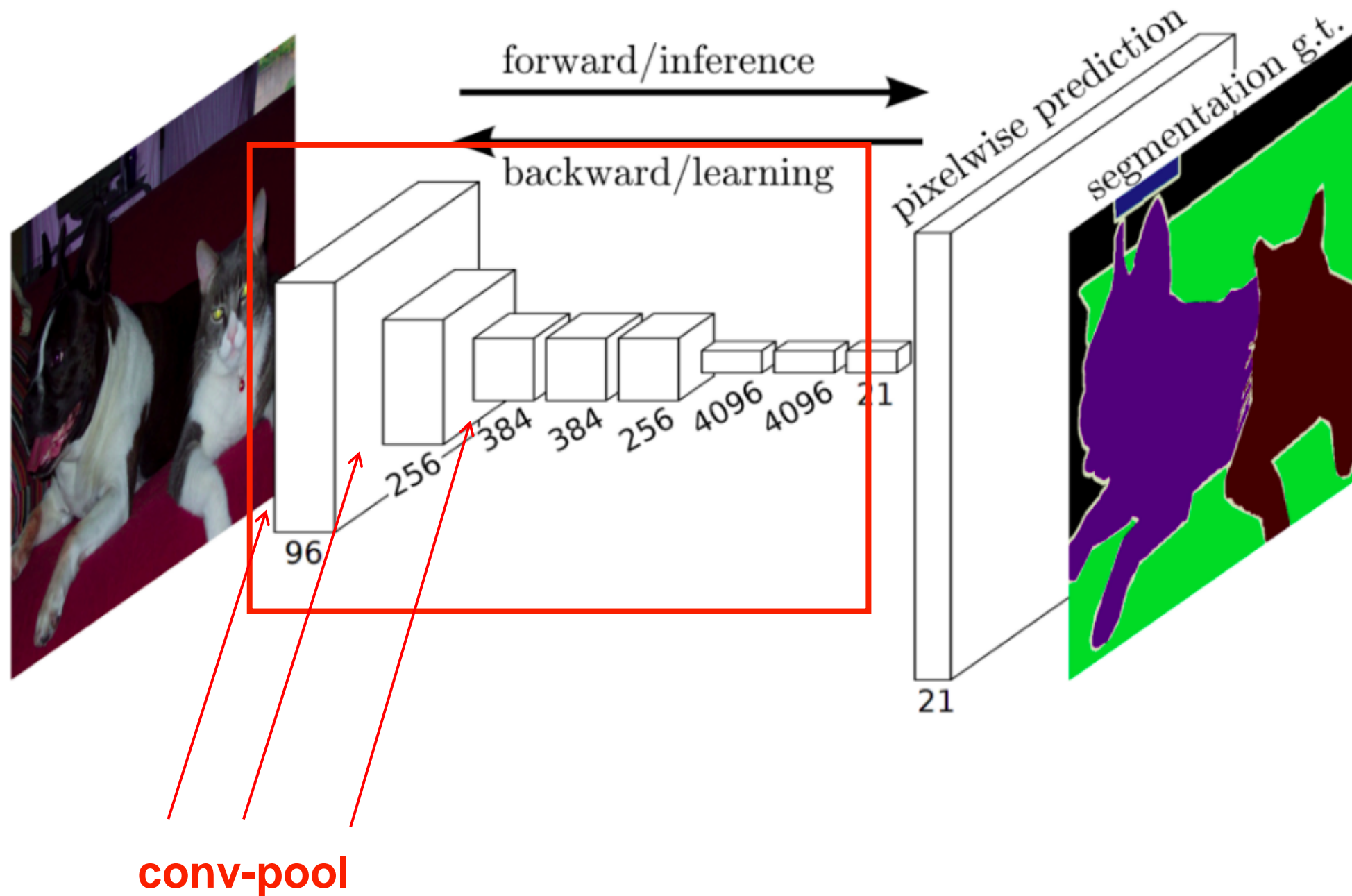
FCN

Fully-Convolutional Network (FCN)

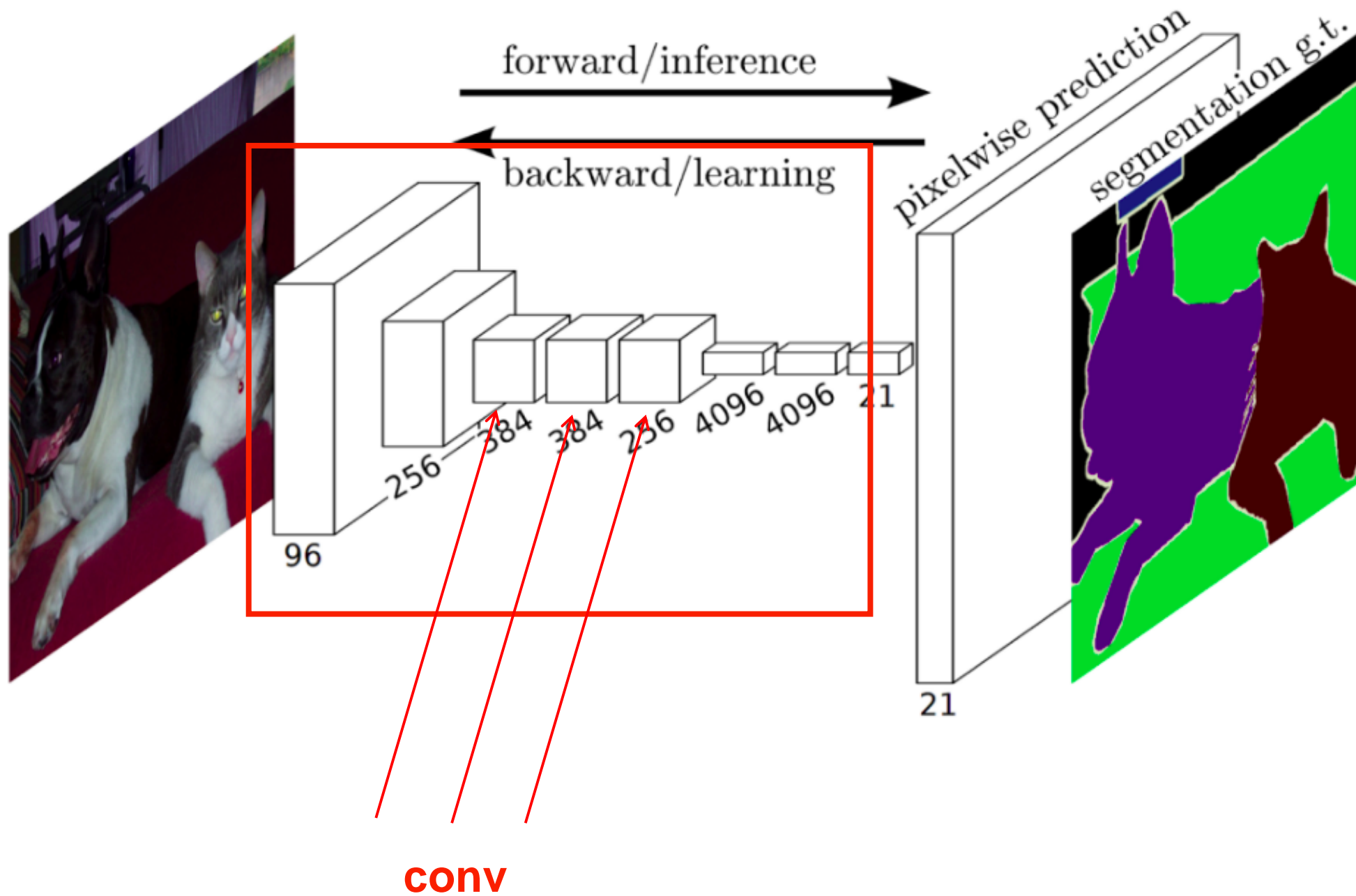
[Статья](#)



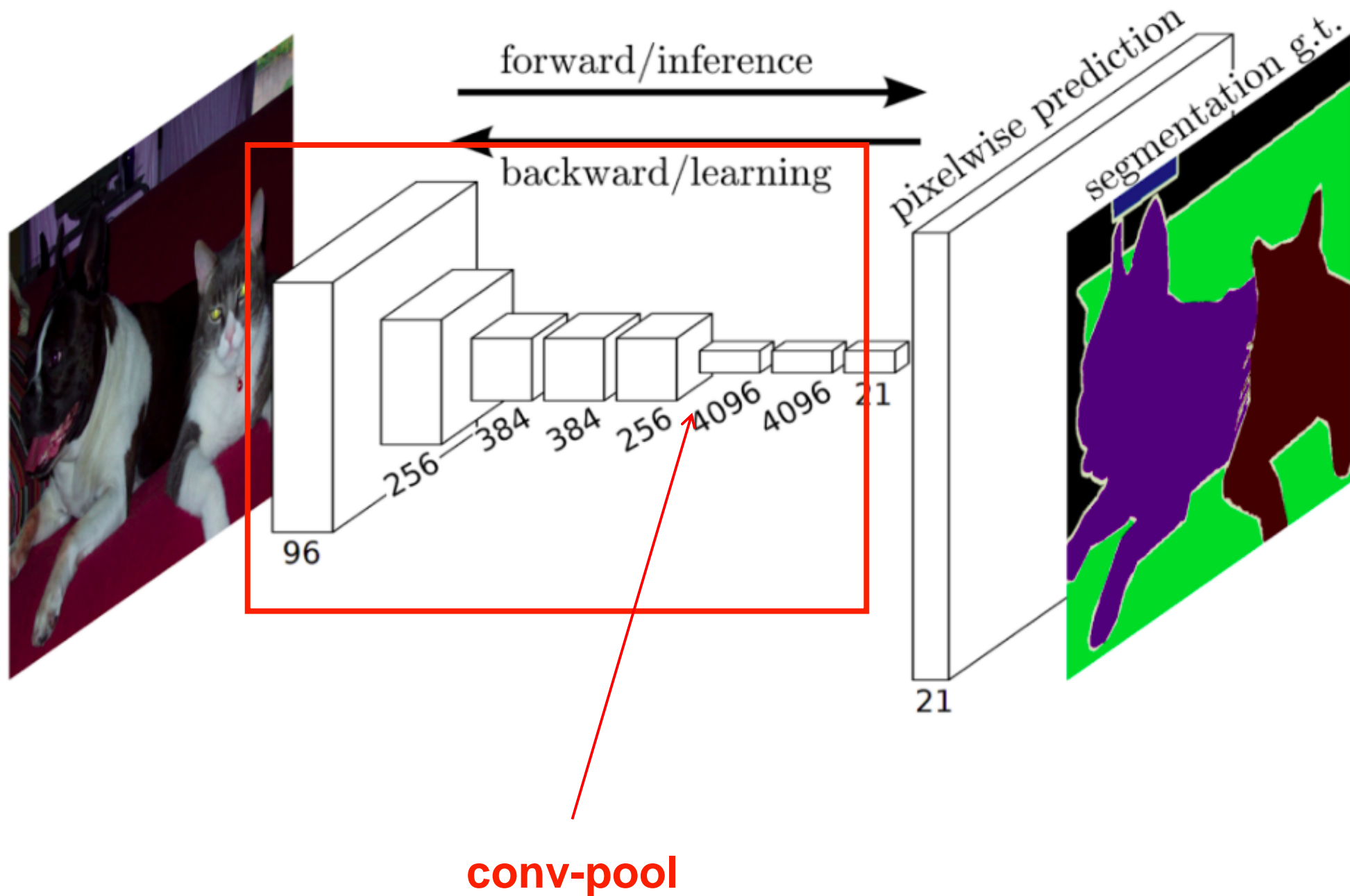
Fully-Convolutional Network (FCN)



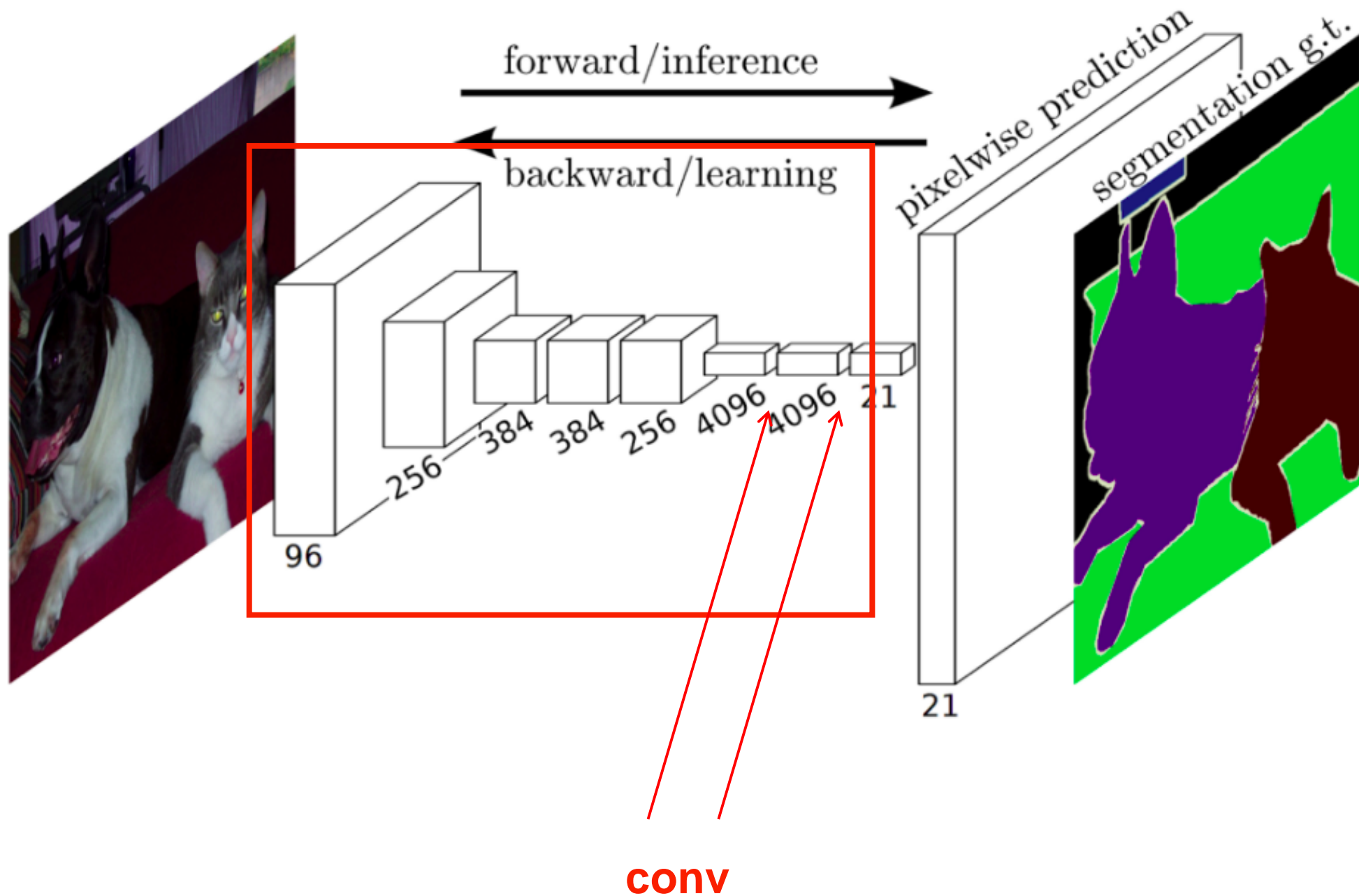
Fully-Convolutional Network (FCN)



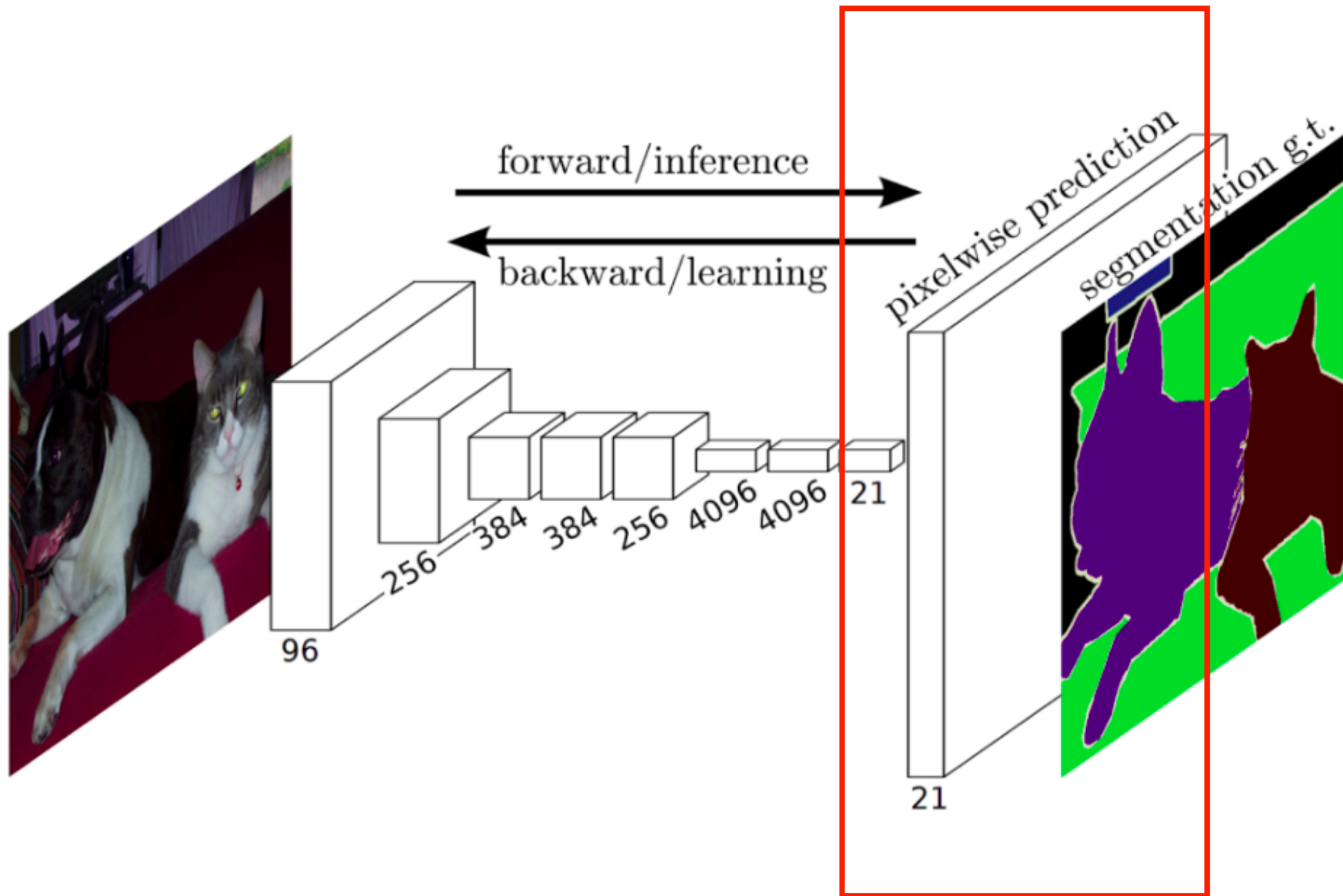
Fully-Convolutional Network (FCN)



Fully-Convolutional Network (FCN)

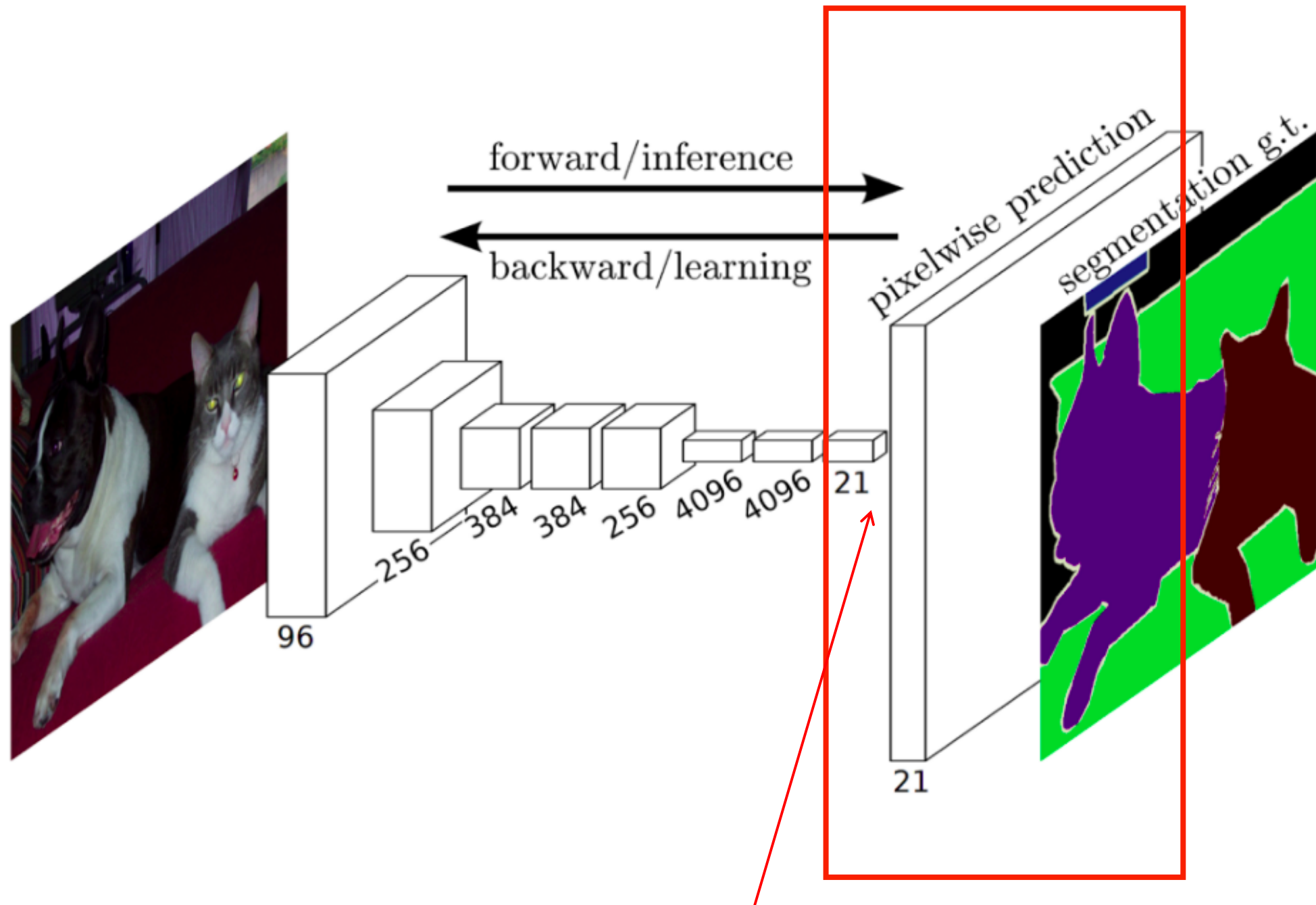


Fully-Convolutional Network (FCN)



Decoder

Fully-Convolutional Network (FCN)



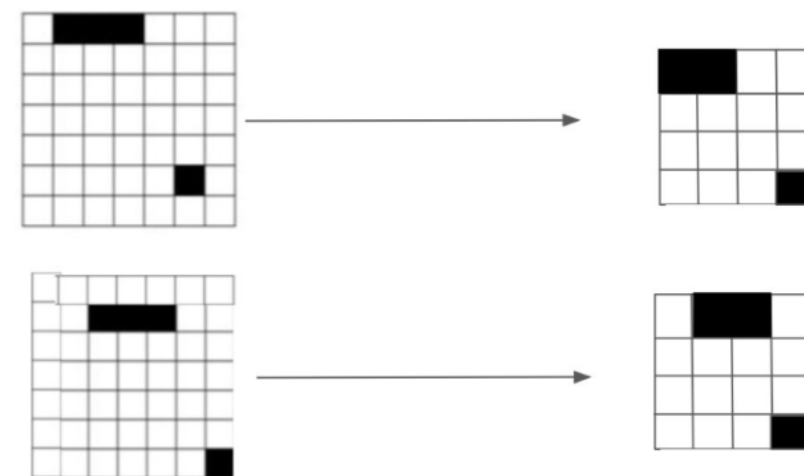
deconv — обратная свертка

Fully-Convolutional Network (FCN)

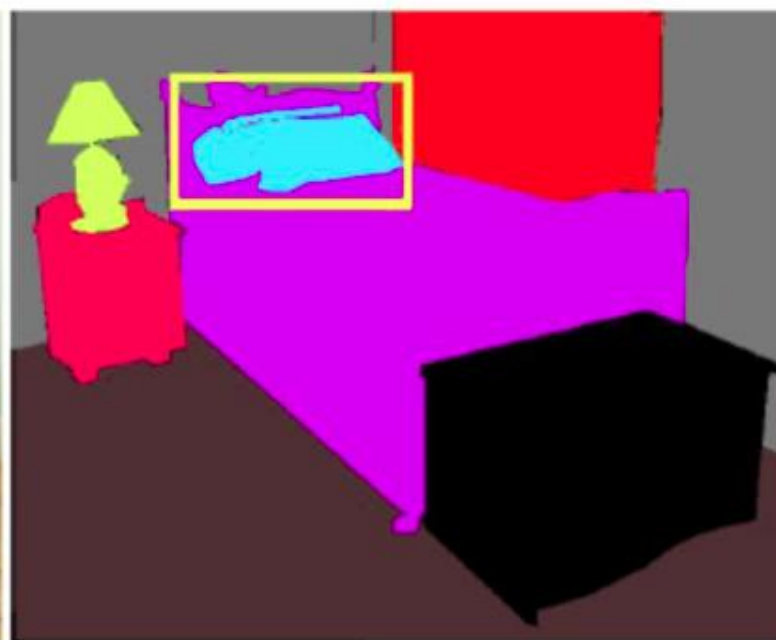
Недостатки FCN

Сильно сжимает изображение, upsampling **плохо восстанавливает пространственную информацию** из сжатого представления.

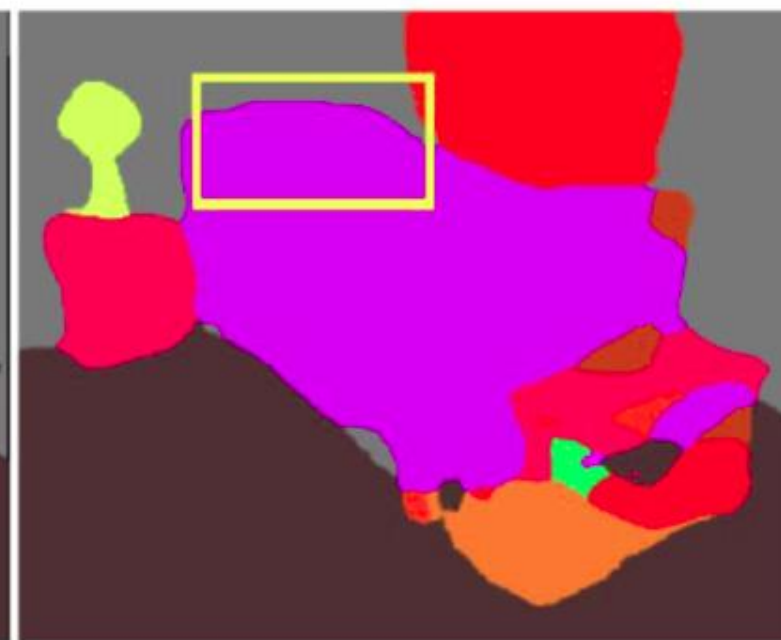
При постоянном уменьшении размера возникает проблема **Scale Variability**.



(a) Image



(b) Ground Truth

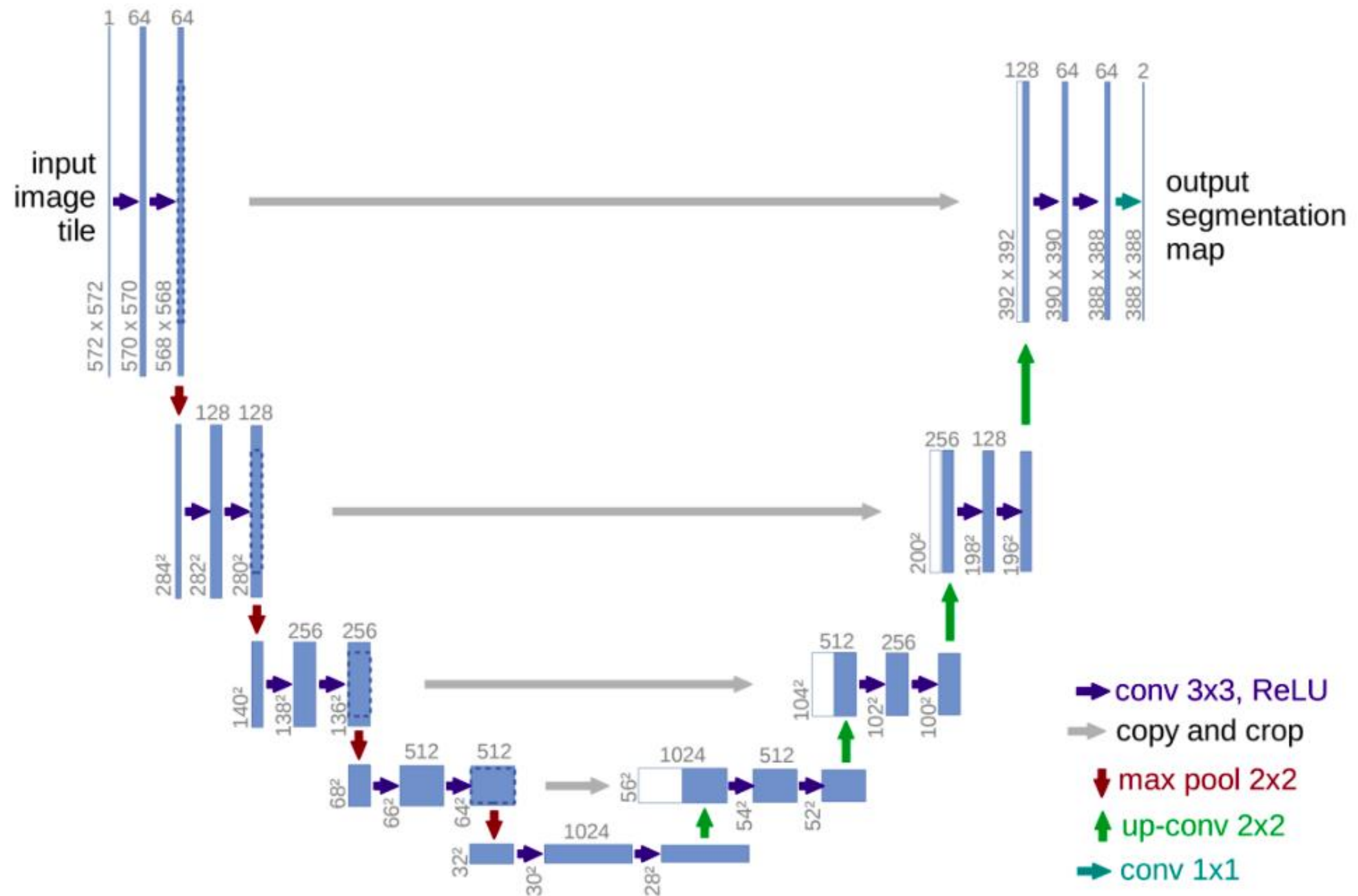


(c) FCN

U-Net

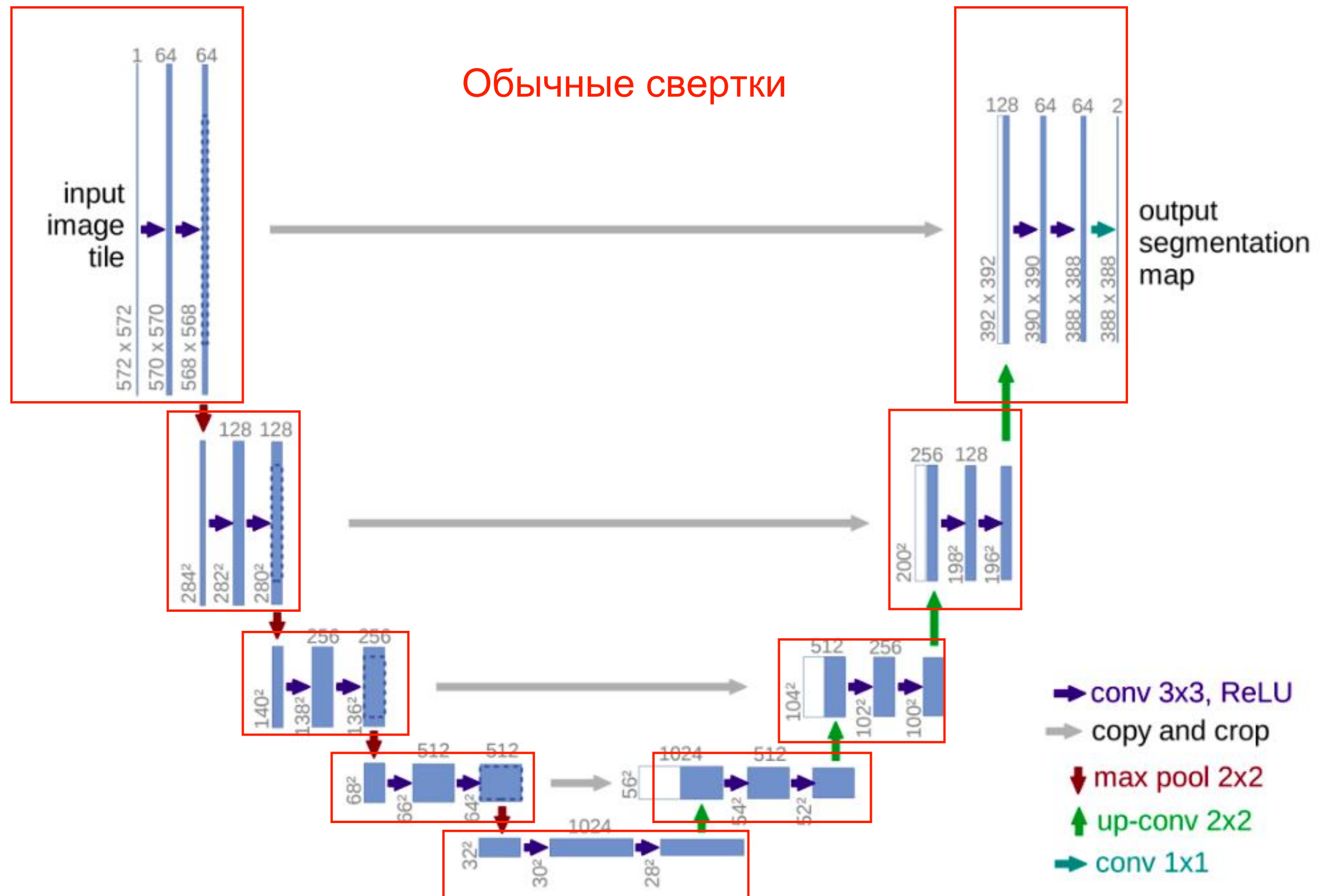
U-Net

Рассмотрим первую версию UNet

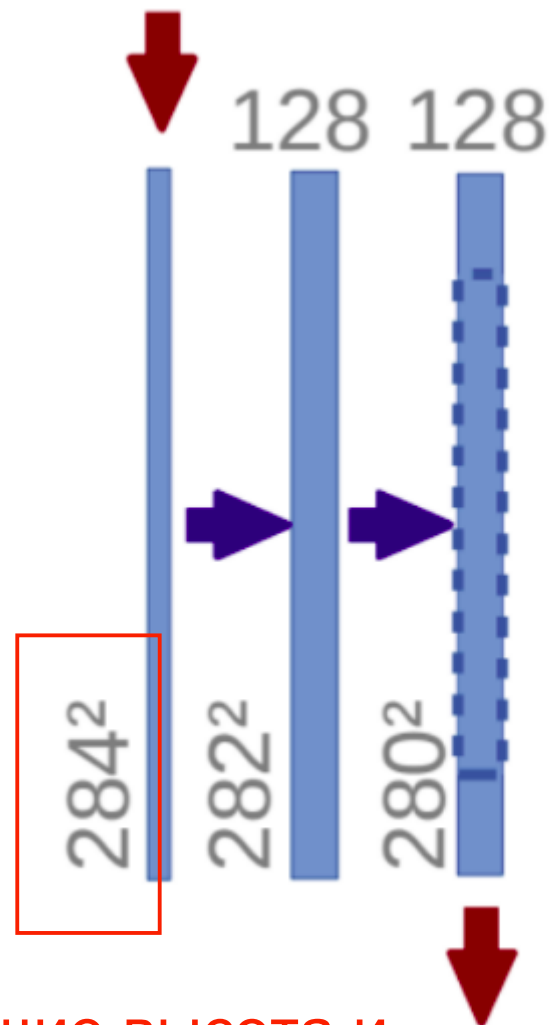


U-Net

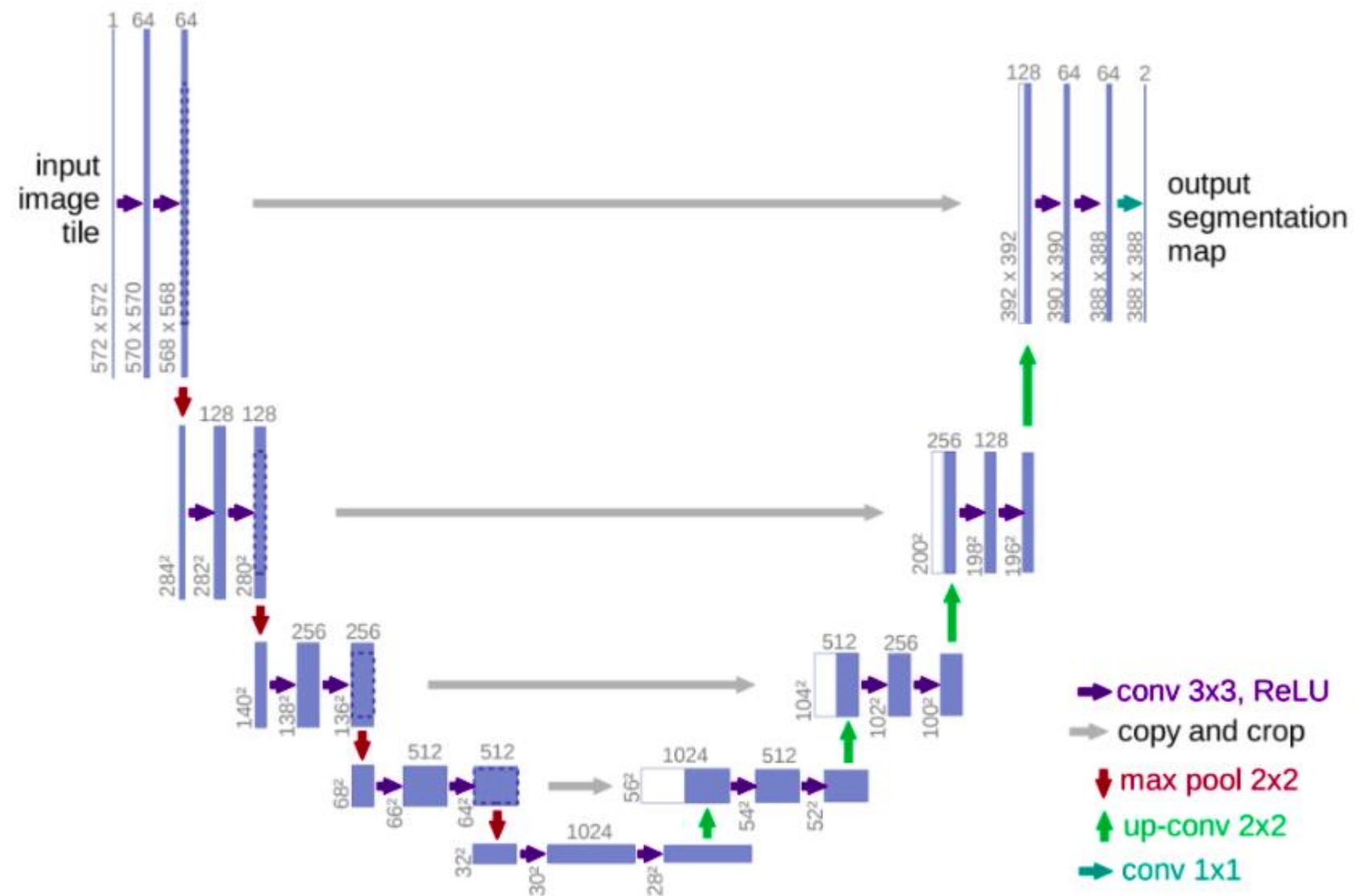
Обычные свертки



U-Net

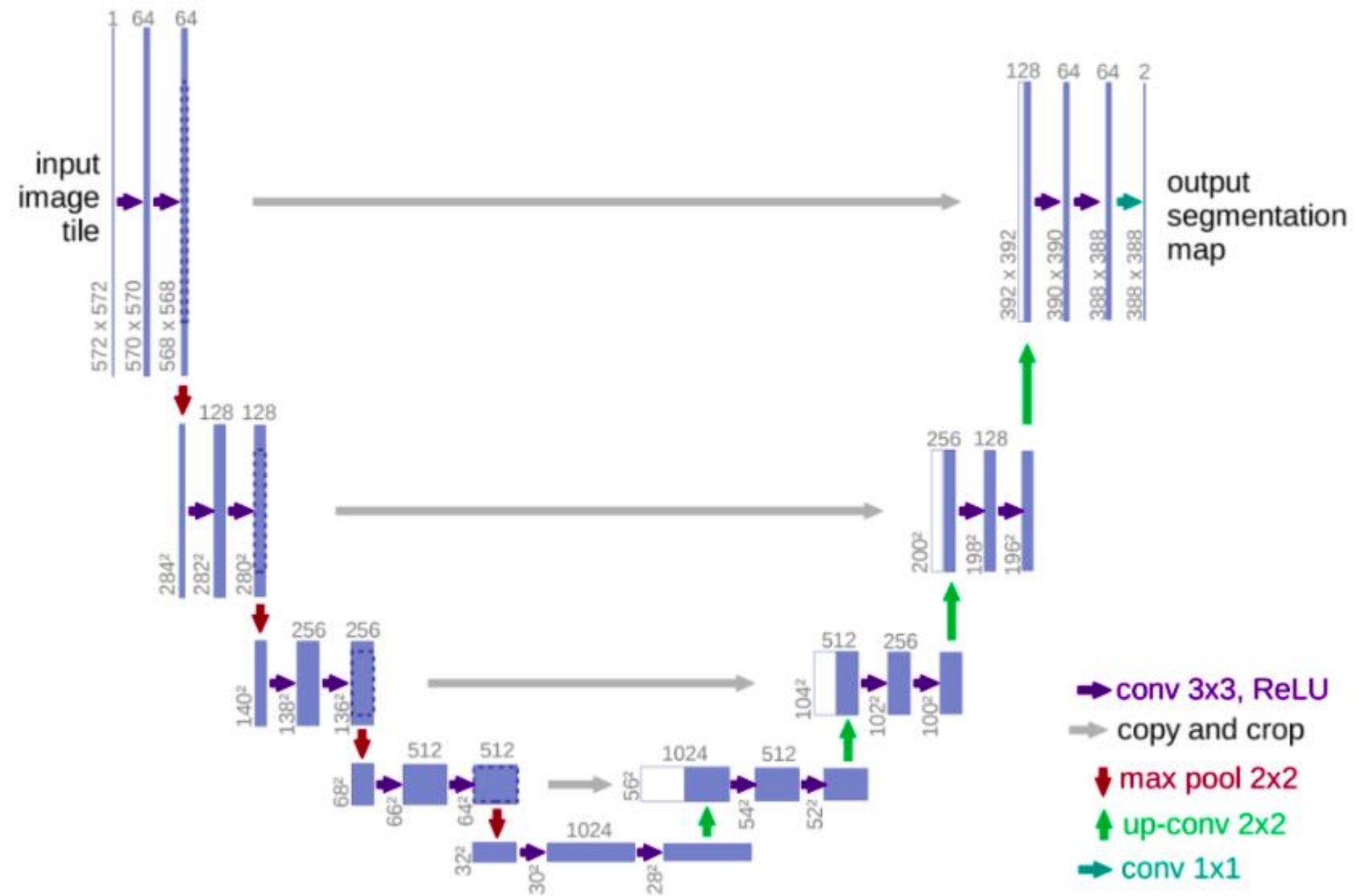
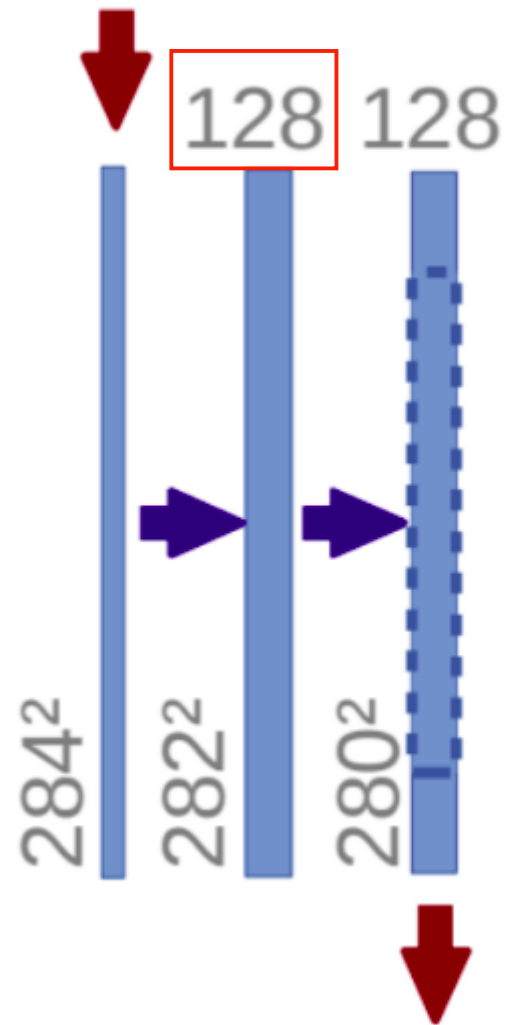


Текущие высота и ширина тензора

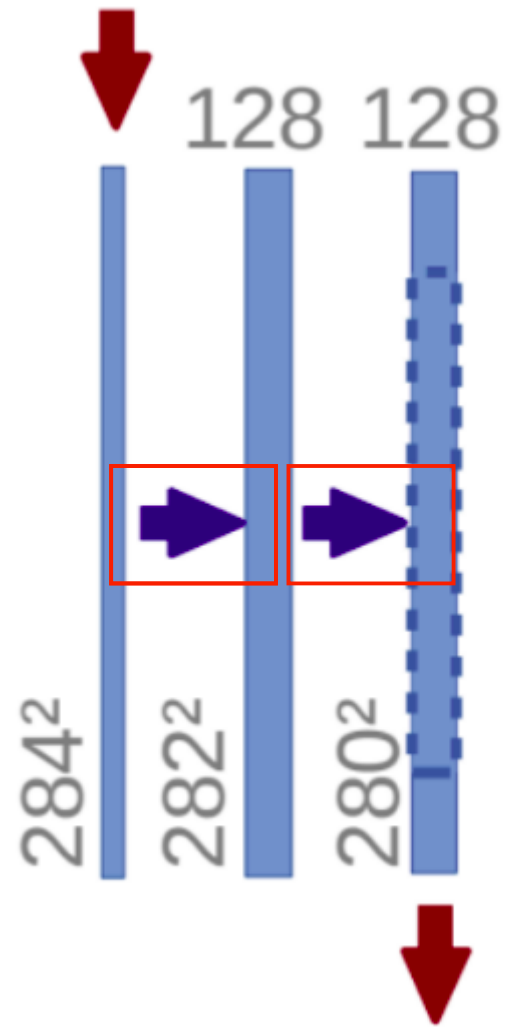


U-Net

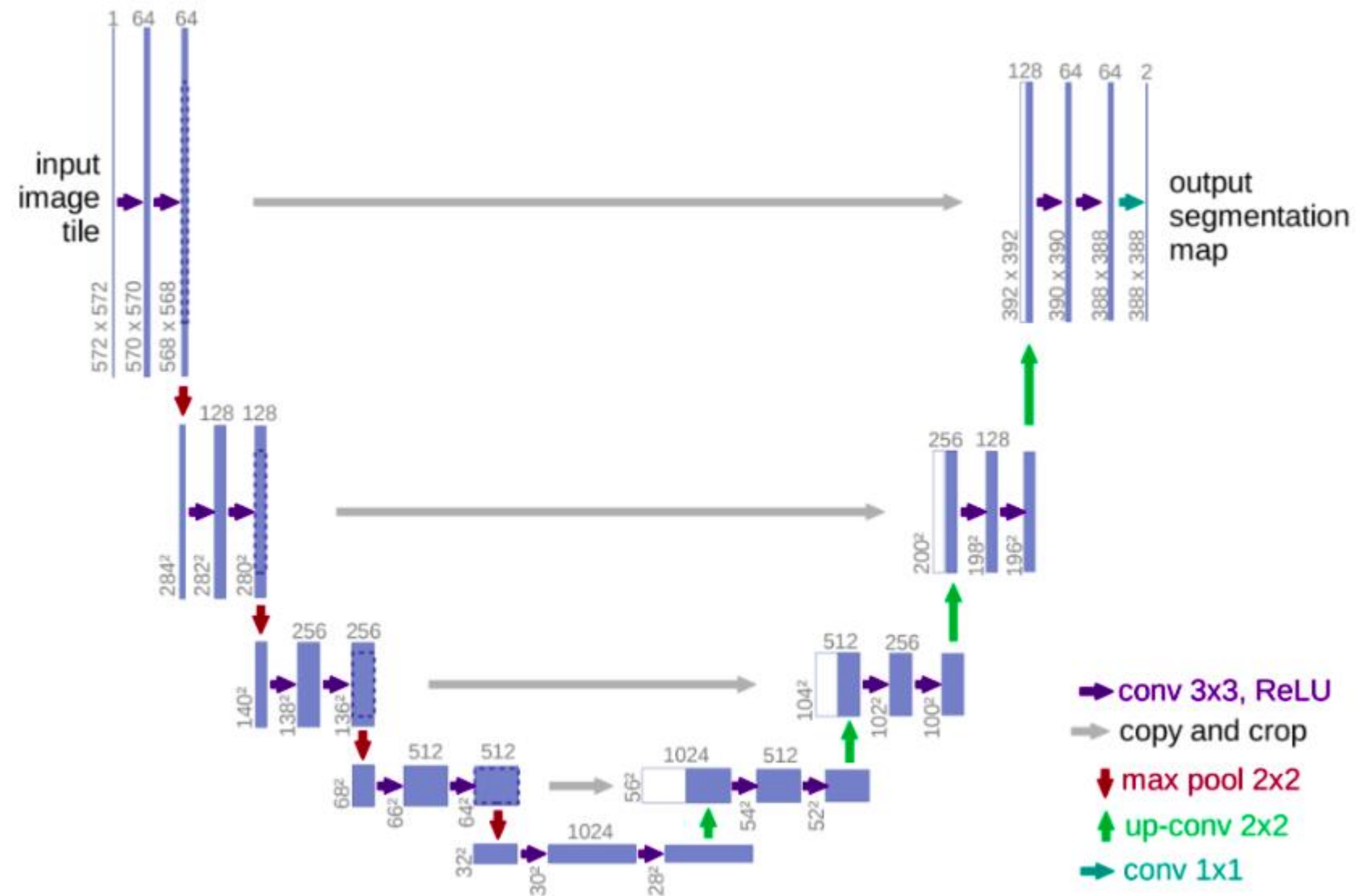
Текущие количество
каналов в тензоре



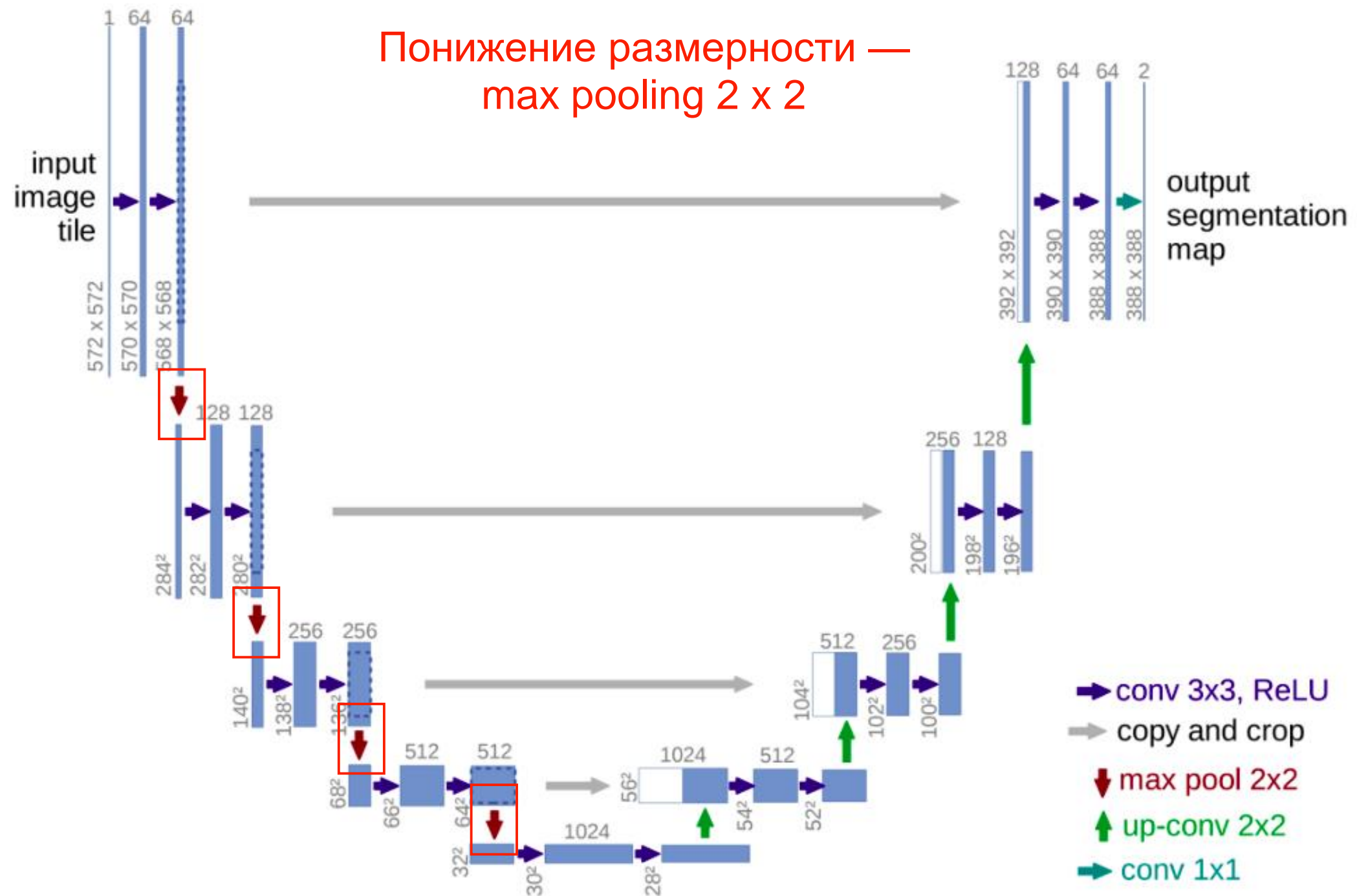
U-Net



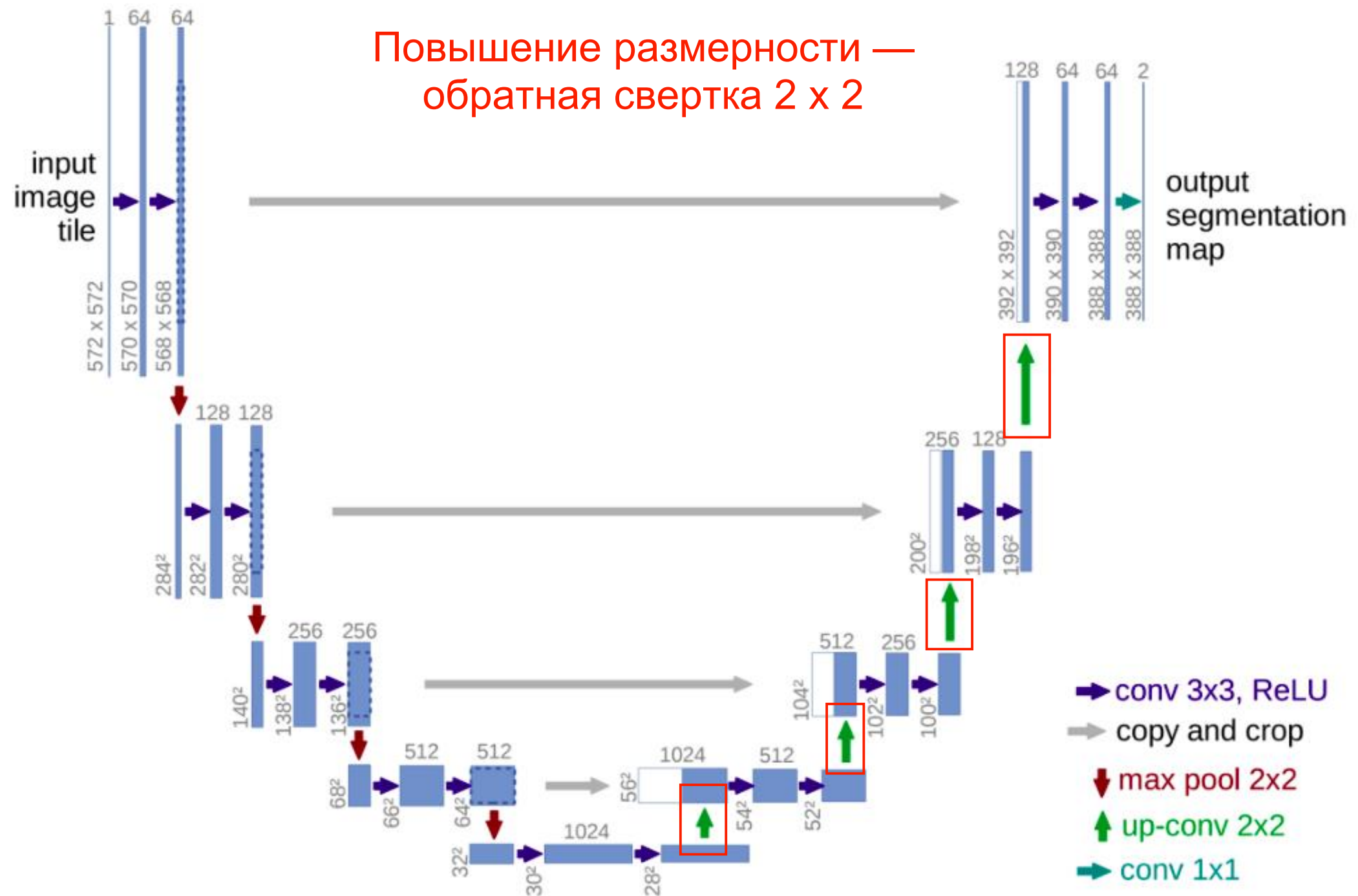
Свертка 3x3, p=0, s=1



U-Net

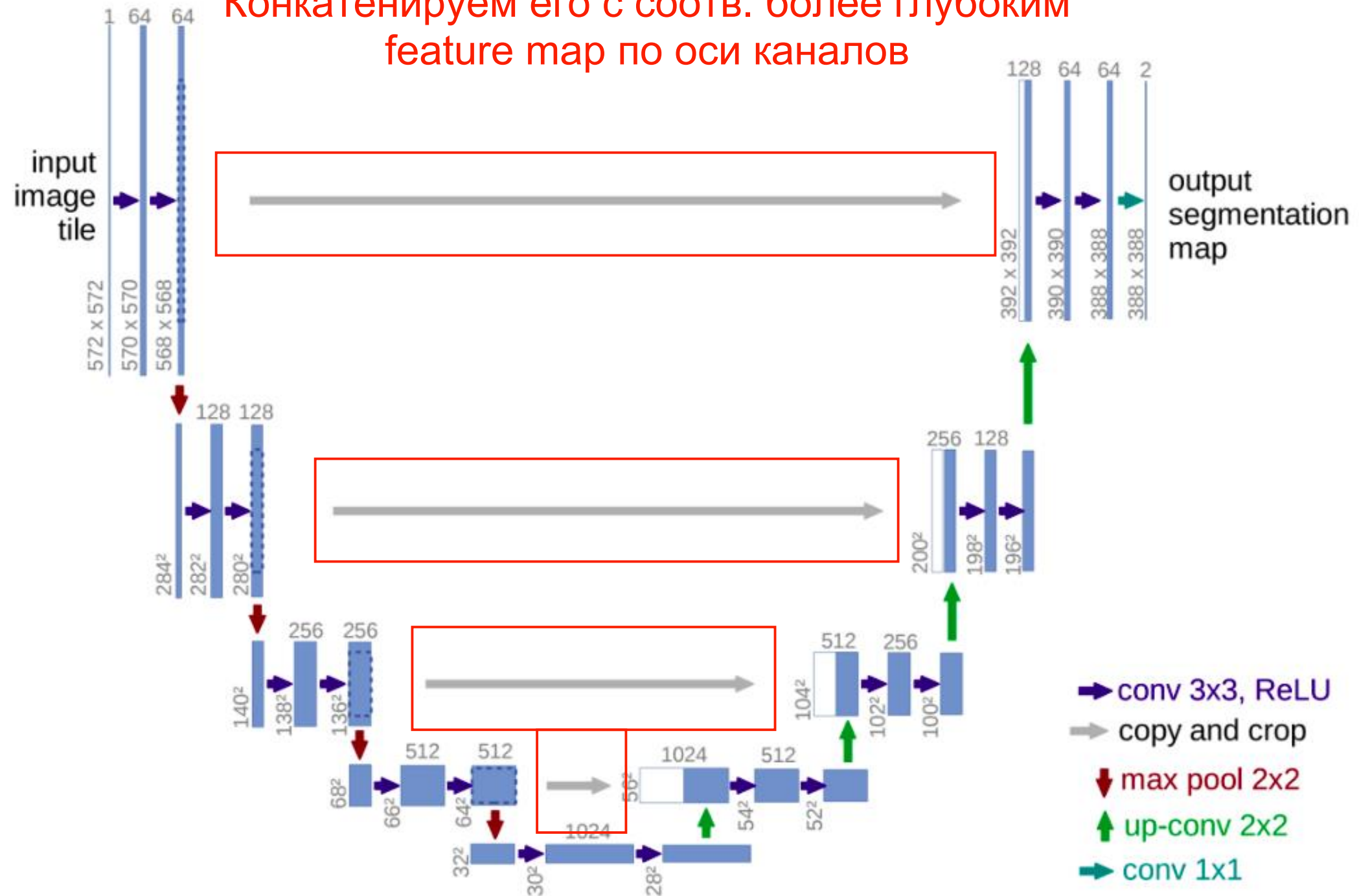


U-Net

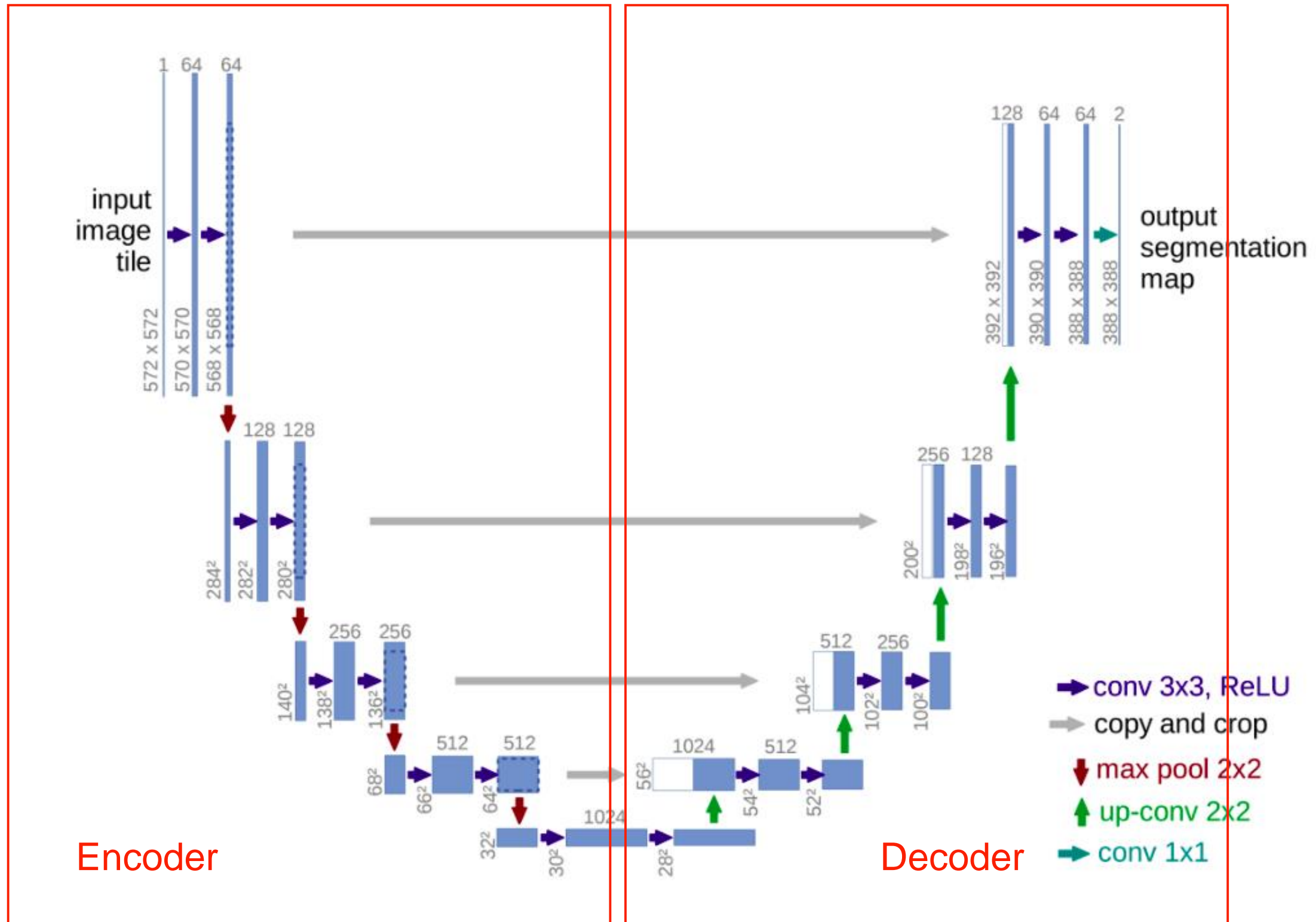


U-Net

Обрезаем feature map с менее глубокого слоя.
Конкатенируем его с соотв. более глубоким
feature map по оси каналов



U-Net

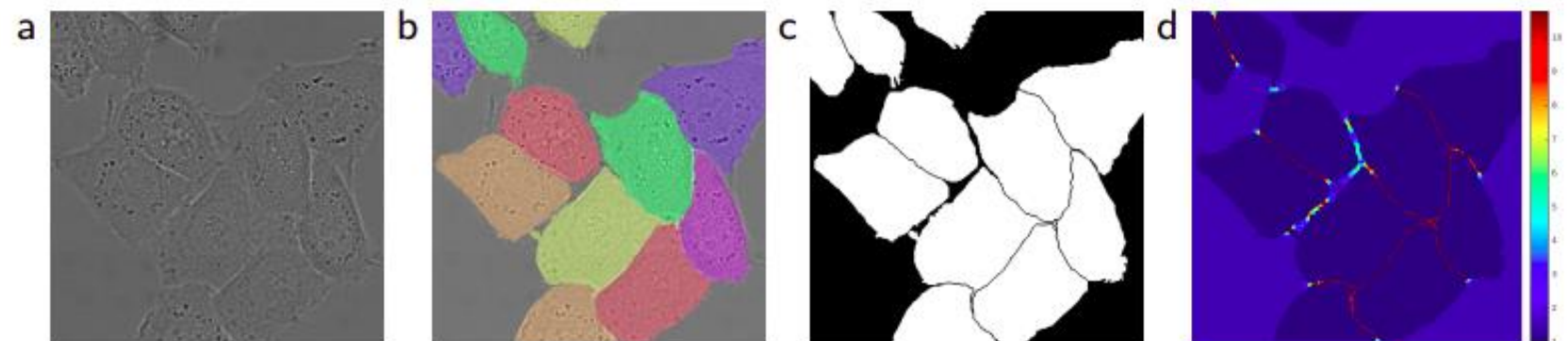
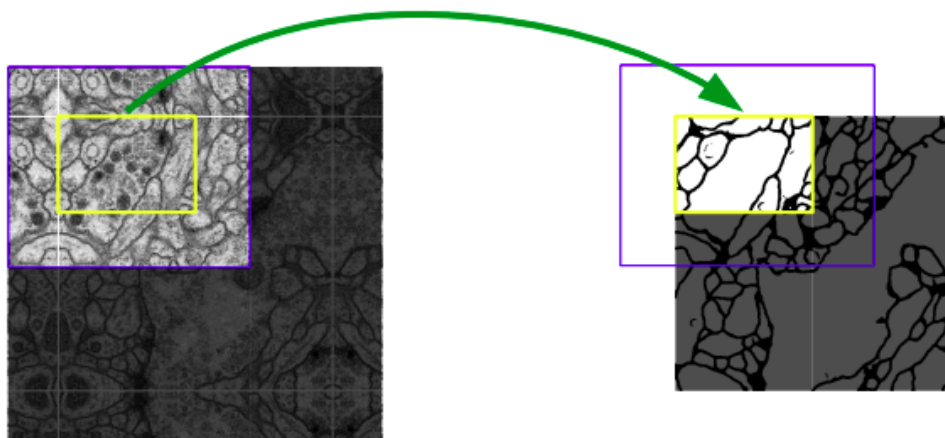


U-Net

Особенности модели

- Модель была предложена для решения задачи **сегментации медицинских картинок**.
- Для того, чтобы не теряла информация на границах изображения, сделали **большой зеркальный padding**.
- Для того, чтобы границы клеток четко отображались нейронной сетью, для лосса задавали **доп. веса**, которые были больше там, где граница между

[Статья](#)



U-Net

Достоинства UNet

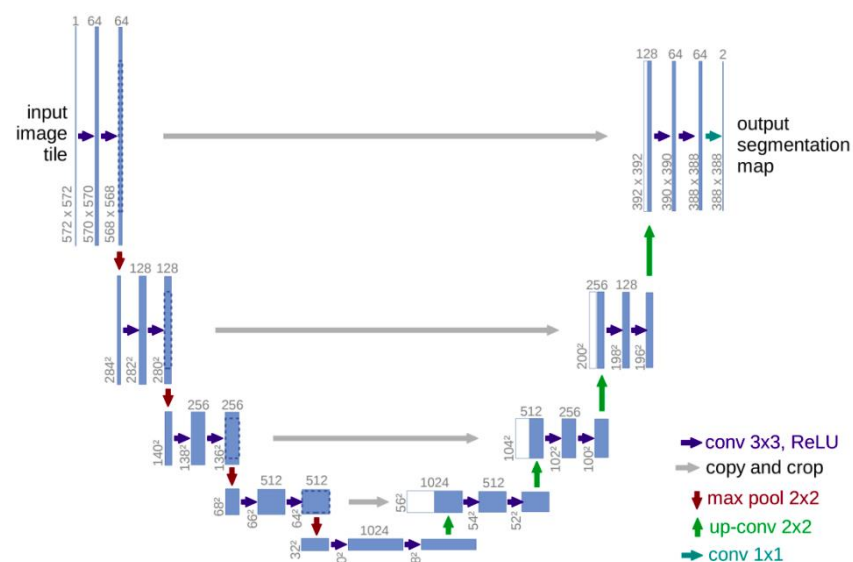
Решается проблема с потерей пространственной информации за счет объединений более глубоких feature map с менее глубокими. Получается хорошее качество сегментации.

Недостатки

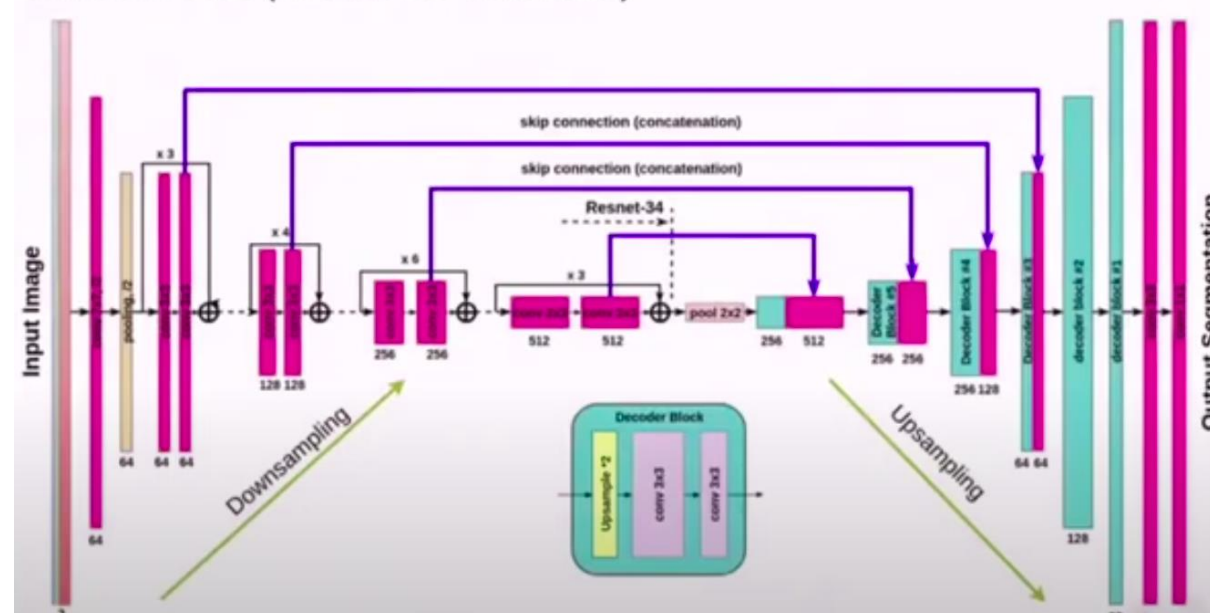
Может переобучиться таким образом, что будут работать только верхние слои U - модели. Тогда модель в основном будет работать с локальными паттернами, но не будет изучать структуру данных в целом.

Модификации U-Net

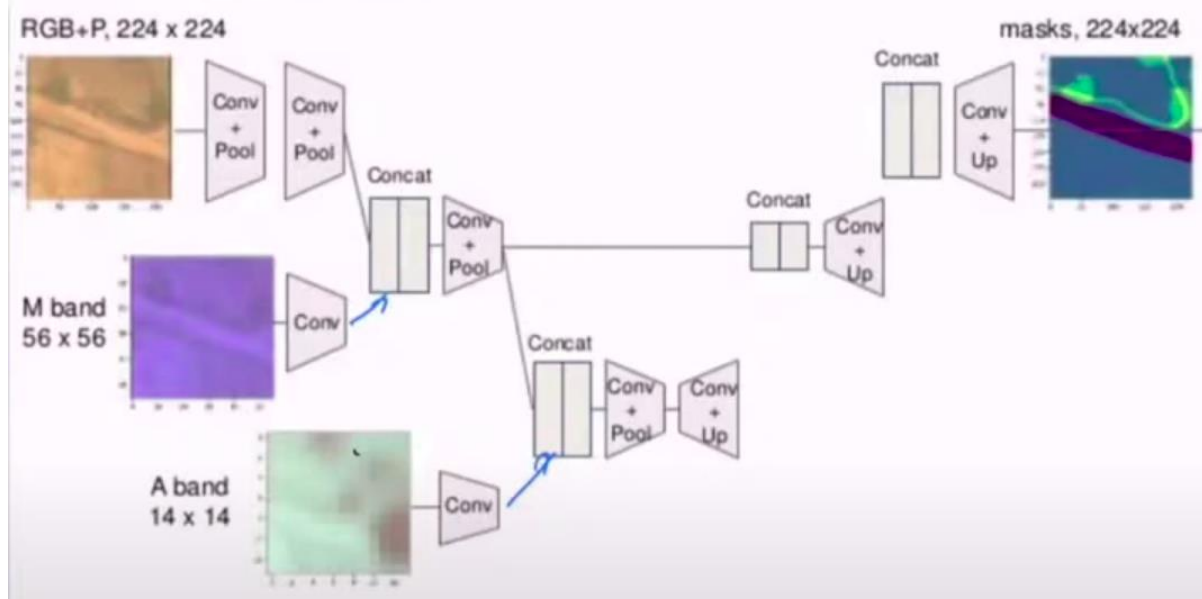
U-Net



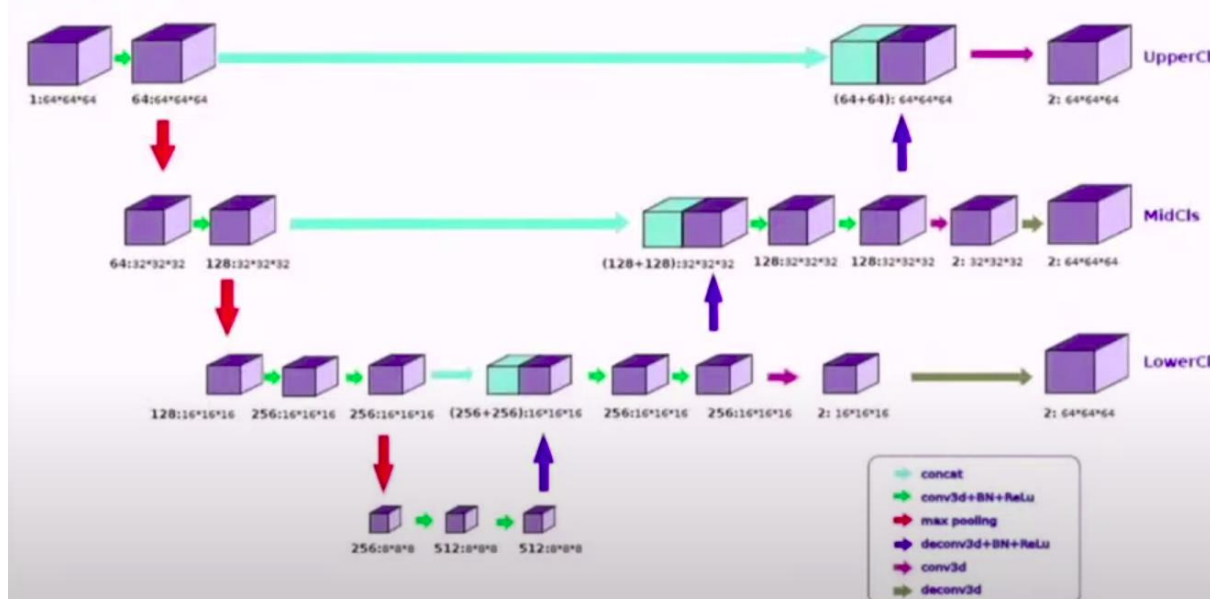
Ternaus Net (U-Net on Steroids)



Multi-Input U-Net



3D U-Net





ВСЁ!