

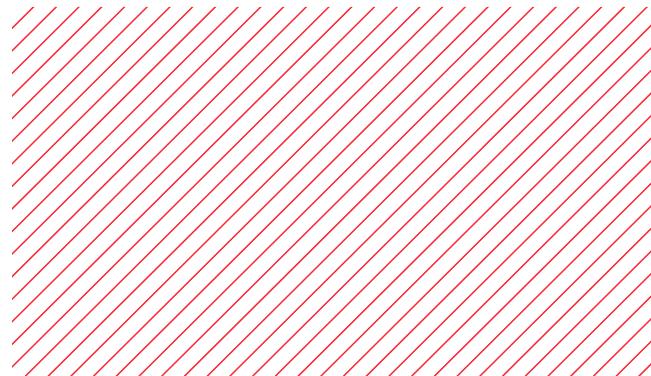
академия
больших
данных



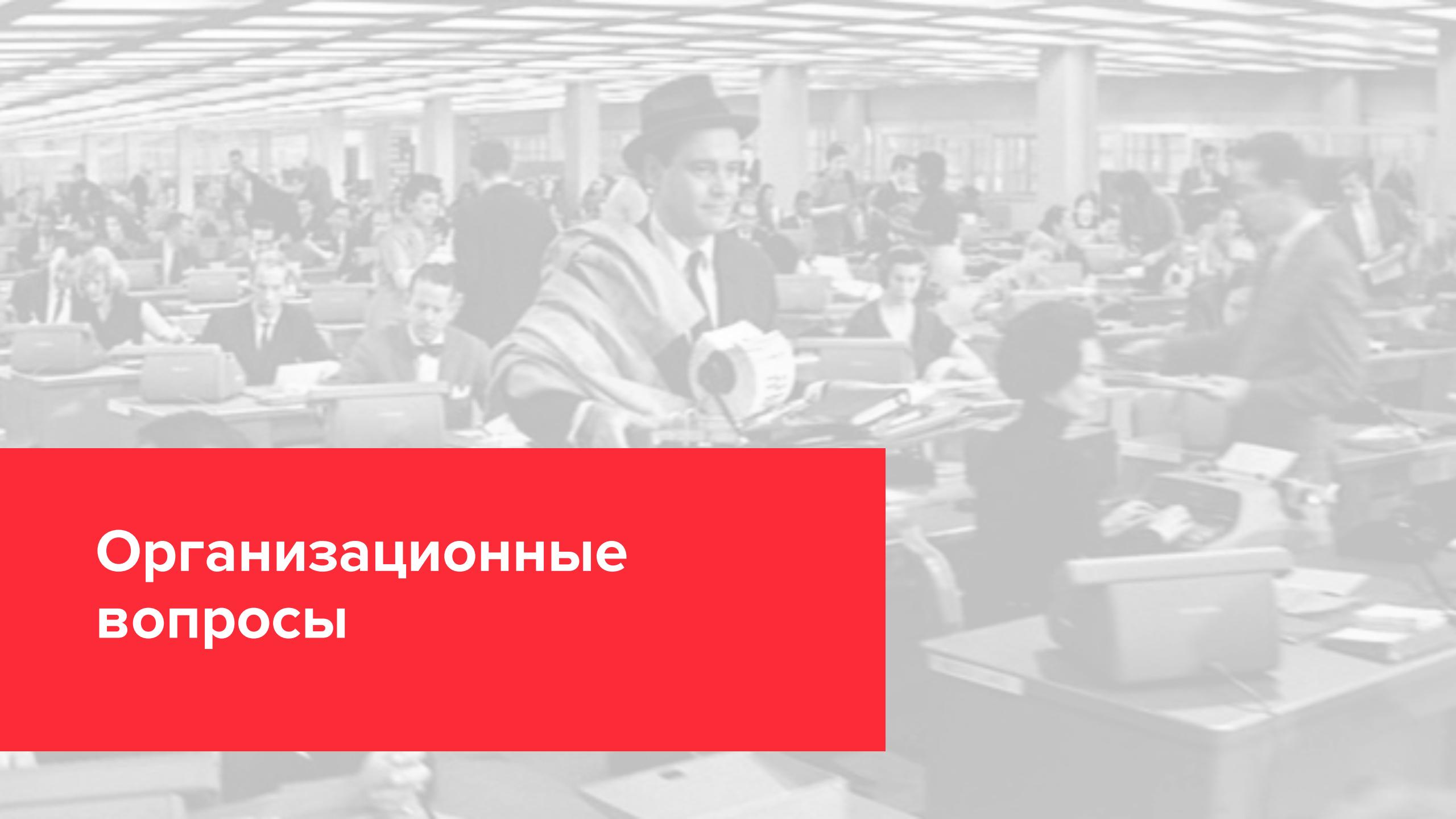
Сегментация

Даниил Лысухин

Программист-исследователь в команде машинного
зрения



Организационные вопросы





Оценки за курс

- 2 конкурса
- 50 баллов максимум каждый
- Возможны бонусы
 - Sharing is caring



Оценки за курс

- Оценки и баллы:
 - 90+: отлично
 - 75+: хорошо
 - 50+: удовлетворительно
 - ...ну вы поняли

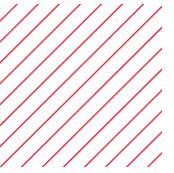


Контест

- Дедлайн по контесту обновлен: **5 Мая 23:59**

Конкурс

#	Team Name	Notebook	Team Members	Score ⓘ	Entries	Last
1	Aleksey Yaroshenko			8.58117	8	1d
2	Ivan			8.90319	5	2d
3	Mikhail Gurevich			9.36638	9	11h
4	Oleg Cheremisin			9.57313	5	1d
5	AntonyZak			9.58521	27	17h
6	Egor Kravchenko			9.96078	9	6d
7	dmitry_vaskin			9.98603	20	3d
8	Andre Pedchenko			10.24442	27	10h
9	Dmitry Dovgopoly			11.04508	5	3d
10	Pumba			11.29027	3	4d



Вопросы?



План

- Сегментация: что и зачем
- Особенности семантической сегментации
- Экономия памяти



Задача сегментации



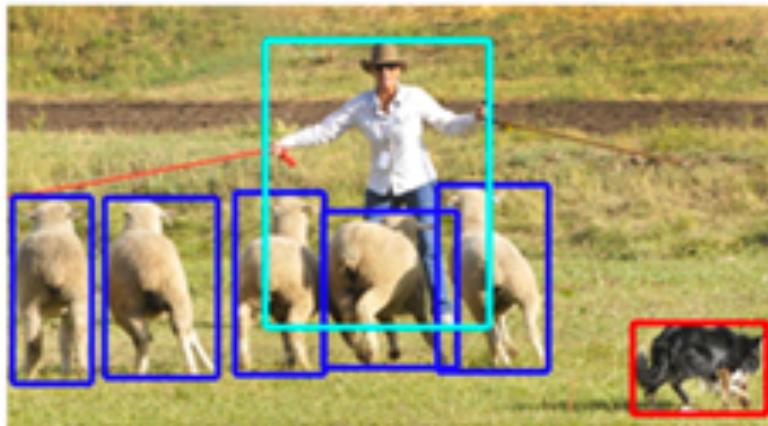
Локализация

- **Детектирование** объектов - частный случай задачи **локализации**
- **Локализация** = **классификация** + указание “где”

Локализация



(a) classification



(b) detection



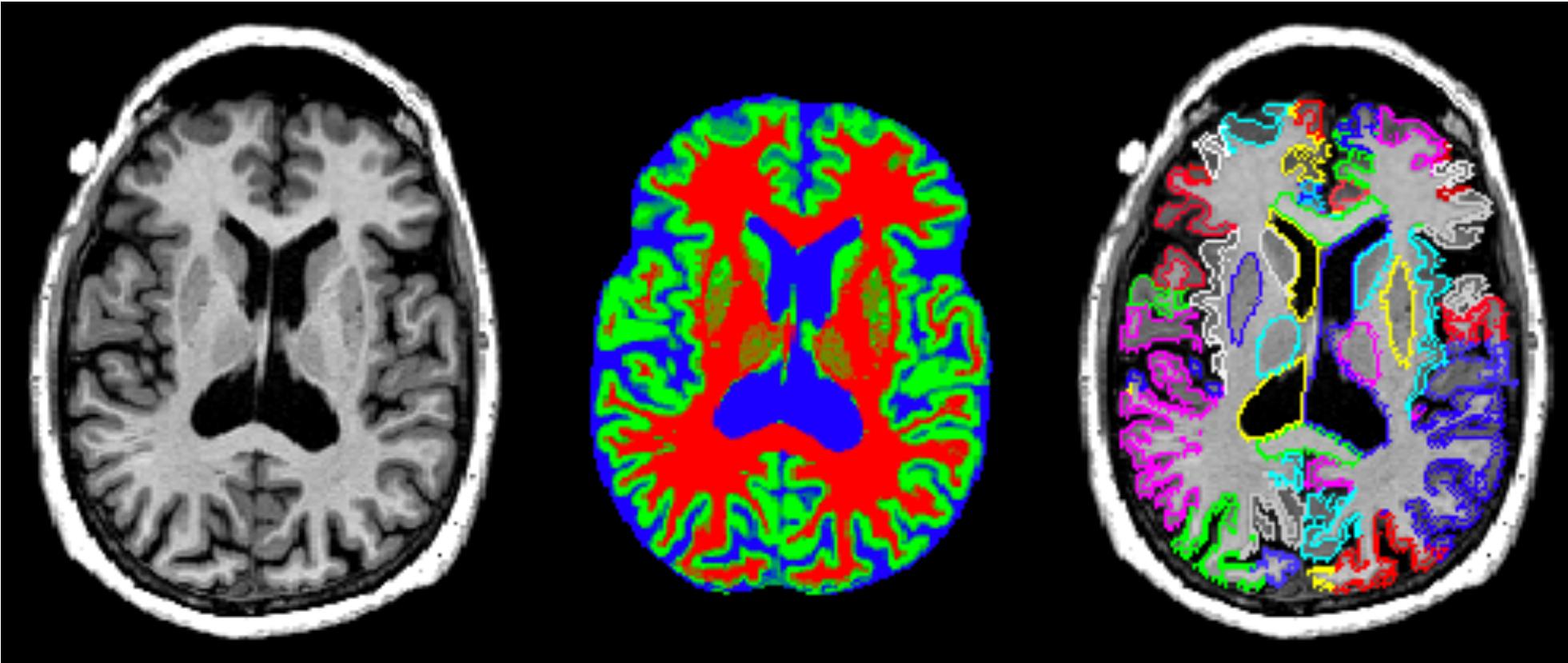
(c) segmentation



Сегментация

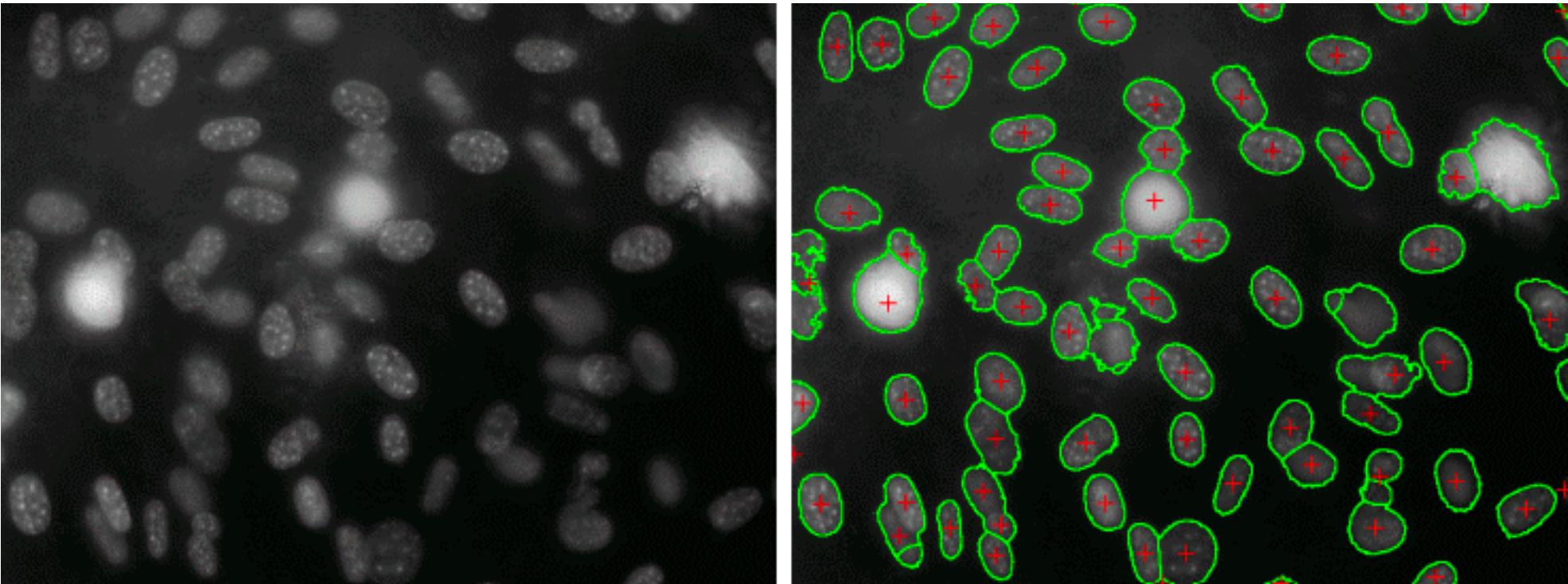
- Если предсказание классов на изображении происходит **попиксельно**, то говорят о **сегментации** изображения

Сегментация



<https://soundray.org/maper/>

Сегментация



<https://www.mathworks.com/matlabcentral/fileexchange/67313-cell-segmentation-seg-self-method>

Сегментация



https://www.researchgate.net/figure/Left-Original-satellite-image-Right-Semantic-segmentation-of-roads-buildings-and_fig7_326883028

Сегментация



Сегментация



Виды сегментации

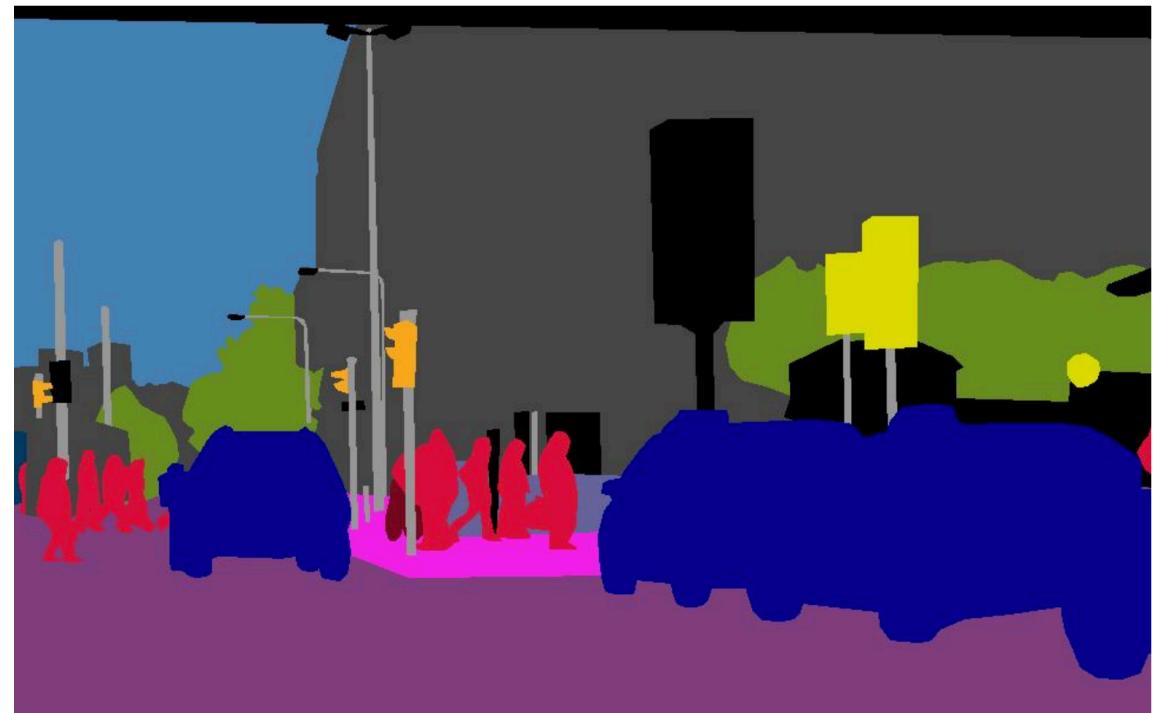




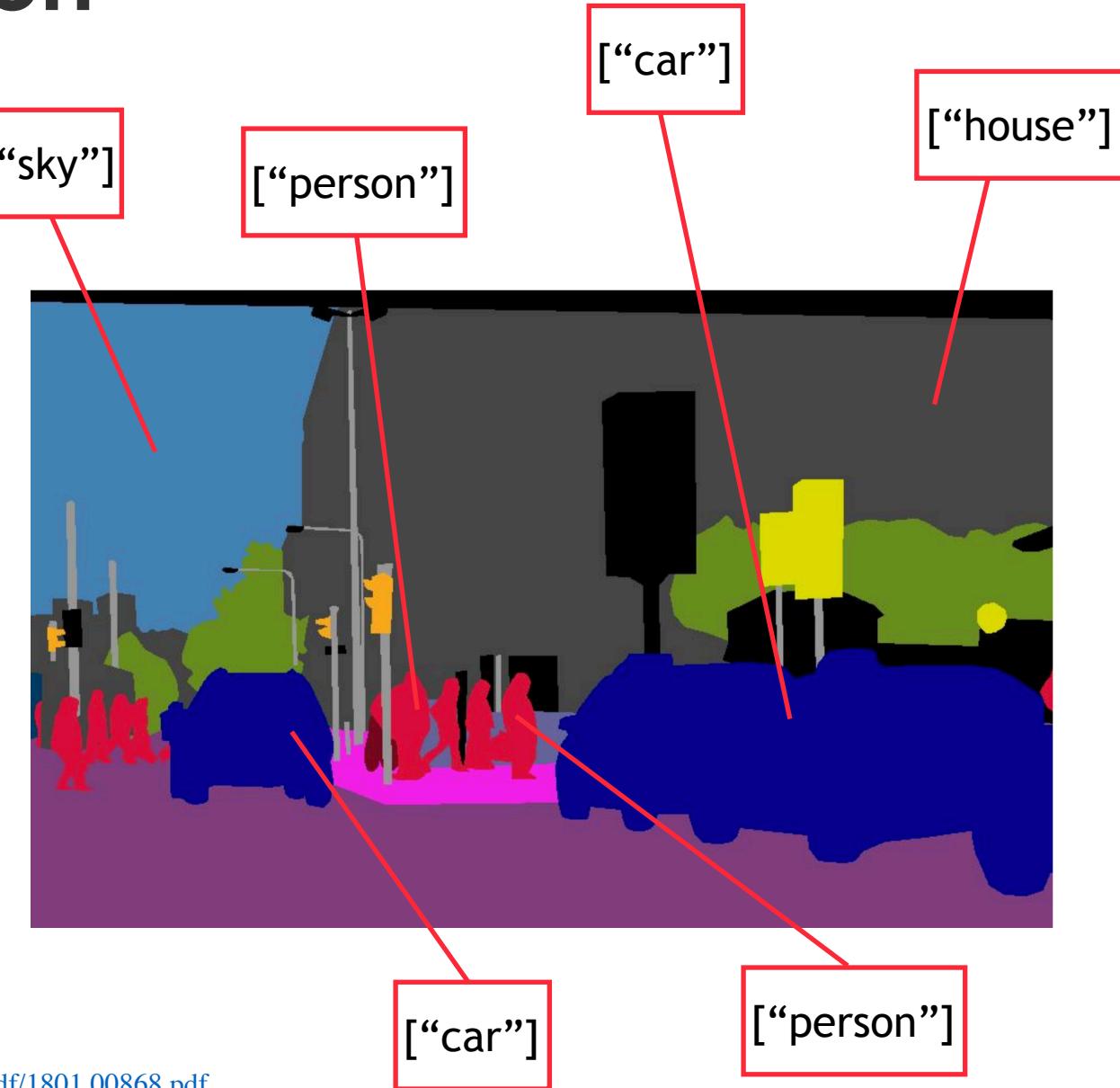
Виды сегментации

1. **Семантическая сегментация (Semantic)** - пикселям ставится в соответствие только их класс

Semantic segmentation



Semantic segmentation

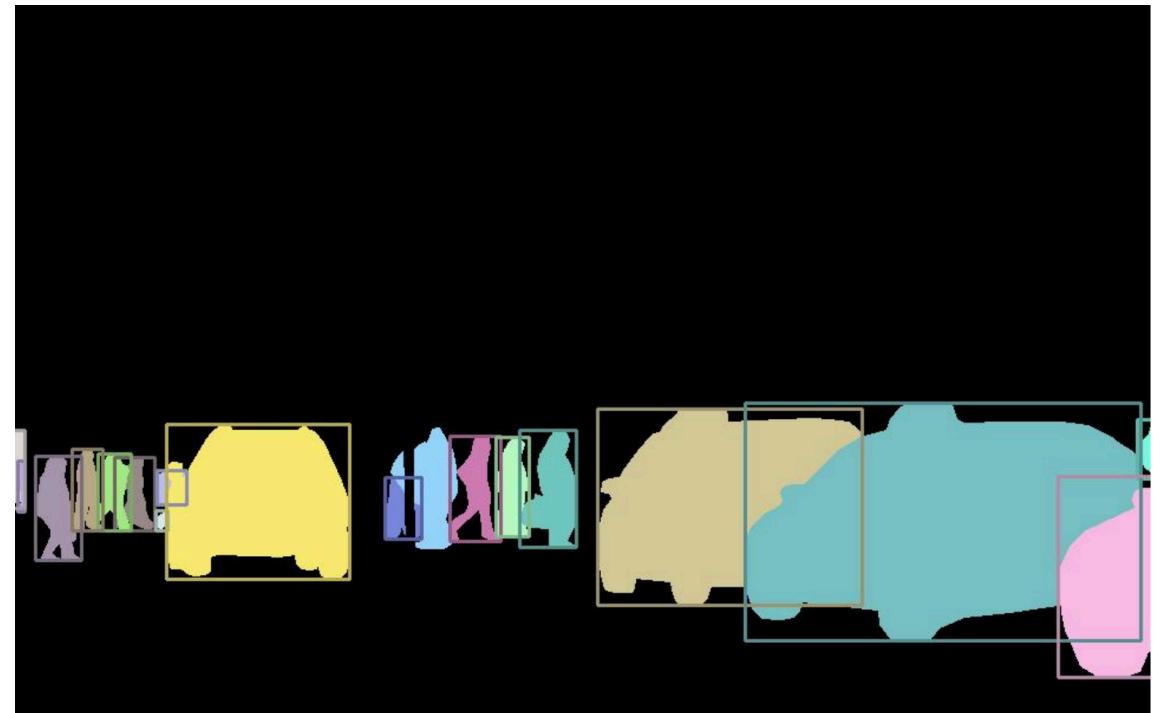




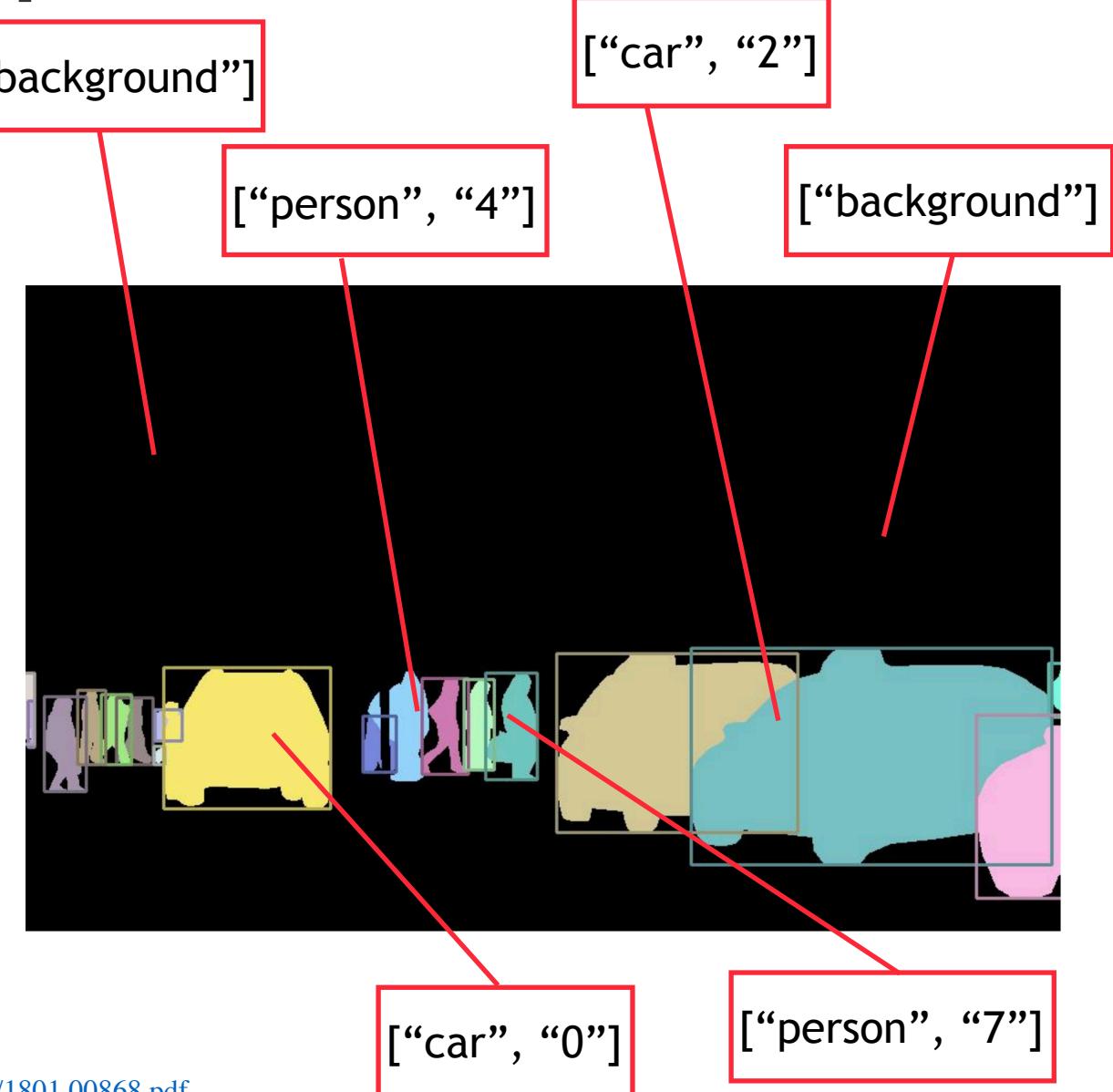
Виды сегментации

1. **Семантическая сегментация (Semantic)** - пикселям ставится в соответствие только их класс
2. **Сегментация объектов (Instance)** - пикселям ставится в соответствие класс + “порядковый номер” **объекта**, которому они принадлежат

Instance segmentation



Instance segmentation





Виды сегментации

1. **Семантическая сегментация (Semantic)** - пикселям ставится в соответствие только их класс
2. **Сегментация объектов (Instance)** - пикселям ставится в соответствие класс + “порядковый номер” объекта, которому они принадлежат
3. **“Полная” сегментация (Panoptic)** - “материалы” (stuff) как в (1), объекты (things) как в (2)

Panoptic segmentation

- Если класс может быть представлен отдельными экземплярами (человек, машина, самолет), то это “объект” (thing)
- Если нет, то это “материал” (stuff)
- Облака?.. Деревья?..



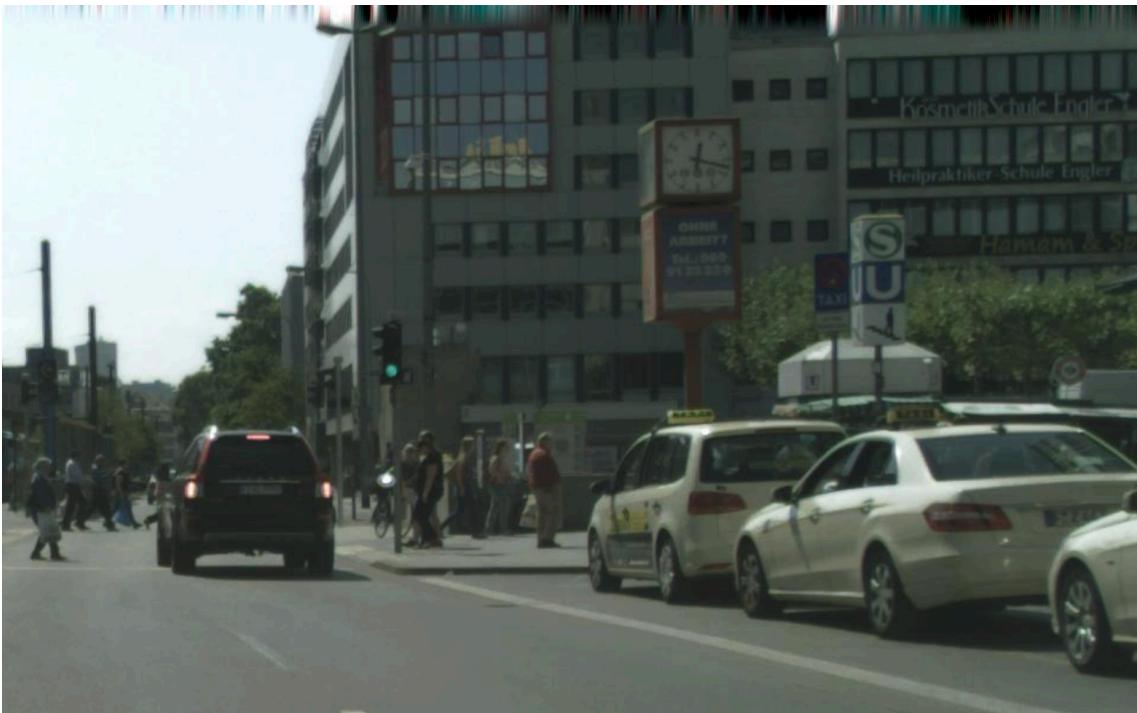
stuff: варианты перевода

Имя Существительное

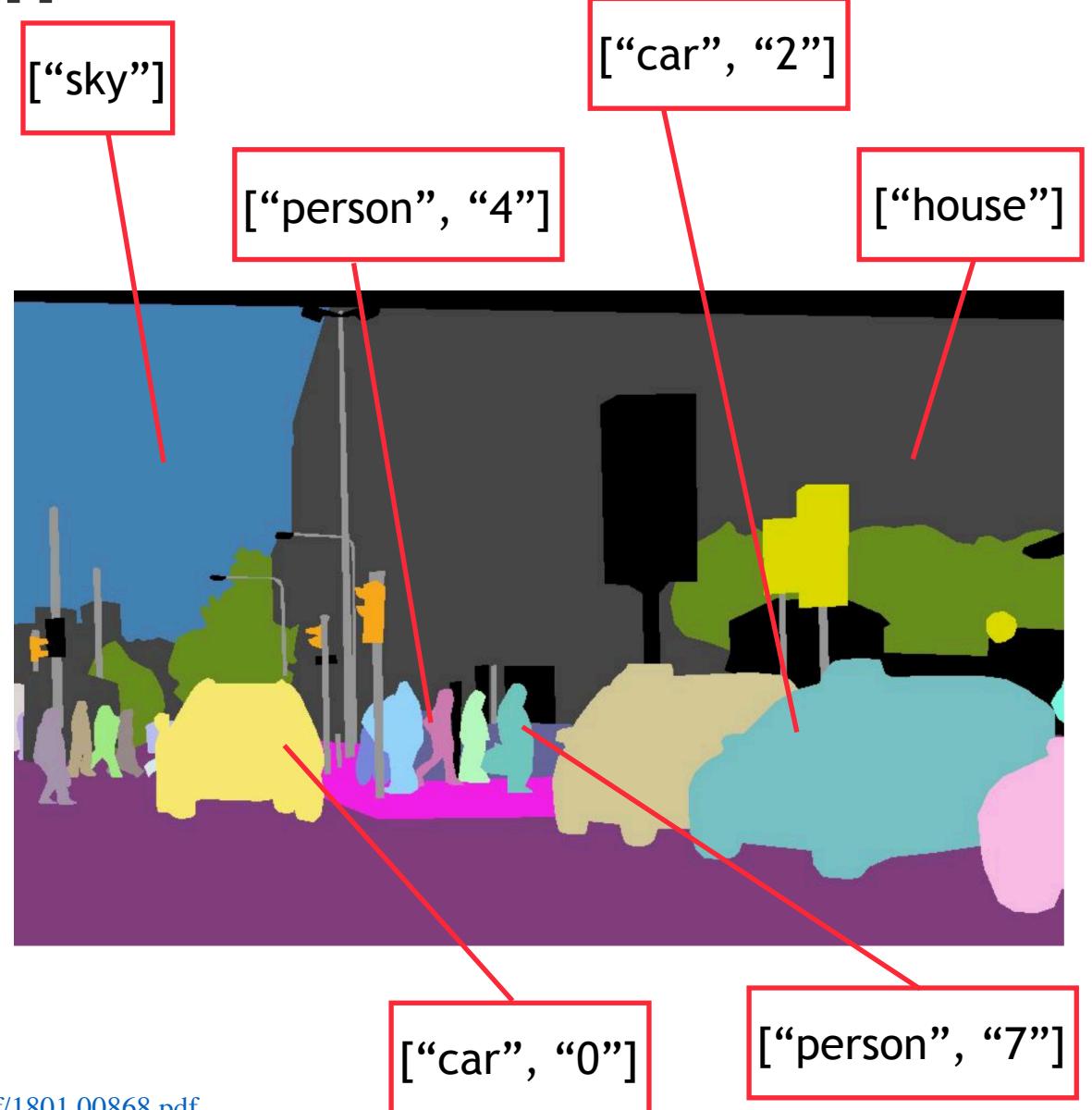
Частота ⓘ

материал	material, stuff, fabric, matter, stock	██████
вещи	things, stuff, belongings, goods, personal effects	██████
вещество	substance, matter, material, agent, stuff	██████
хлам	trash, rubbish, junk, stuff, lumber, rummage	██████
дрянь	stuff, trash, lousy, muck, yuck, rotter	██████
материя	matter, material, substance, fabric, stuff	██████
чепуха	nonsense, stuff, crap, applesauce, malarkey, baloney	██████
фигня	stuff, rubbish	██████

Panoptic segmentation



Panoptic segmentation





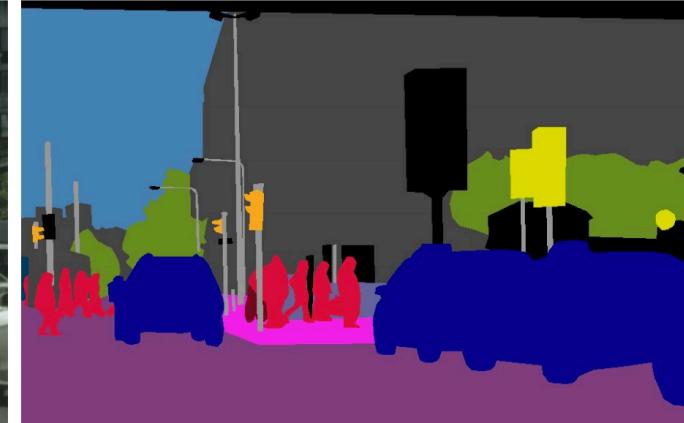
Виды сегментации

1. **Семантическая сегментация (Semantic)** - пикселям ставится в соответствие только их класс
2. **Сегментация объектов (Instance)** - пикселям ставится в соответствие класс + “порядковый номер” **объекта**, которому они принадлежат
3. **“Полная” сегментация (Panoptic)** - “материалы” (stuff) как в (1), объекты (things) как в (2)

Виды сегментации



(a) image



(b) semantic segmentation



(c) instance segmentation



(d) panoptic segmentation



Виды сегментации

- Вопросы?

Семантическая сегментация





Семантическая сегментация

- Семантическая сегментация ≈ пиксельная классификация



Семантическая сегментация

- Семантическая сегментация ≈ пиксельная классификация
- Что на входе модели?
- Что на выходе?
- Как выглядит таргет?
- Как выглядит лосс?

Семантическая сегментация



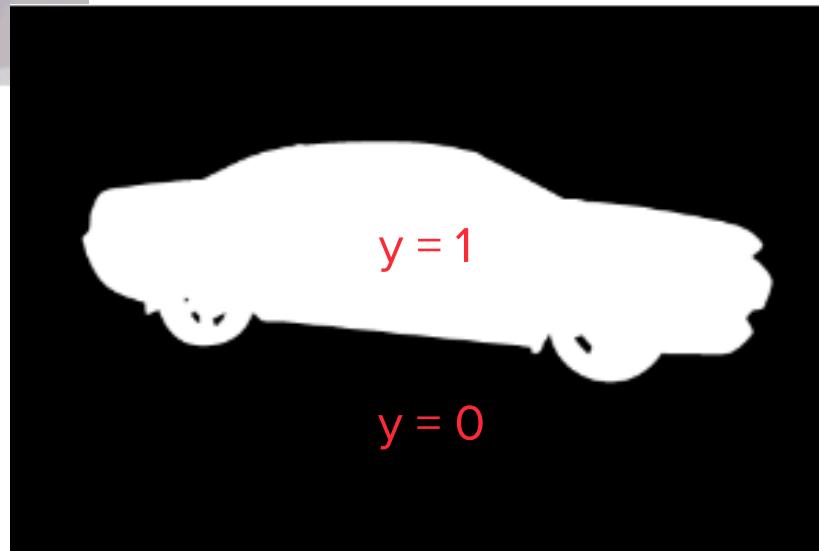
Исходная картинка
RGB

Семантическая сегментация



Исходная картинка
RGB

Бинарная маска
(таргет)

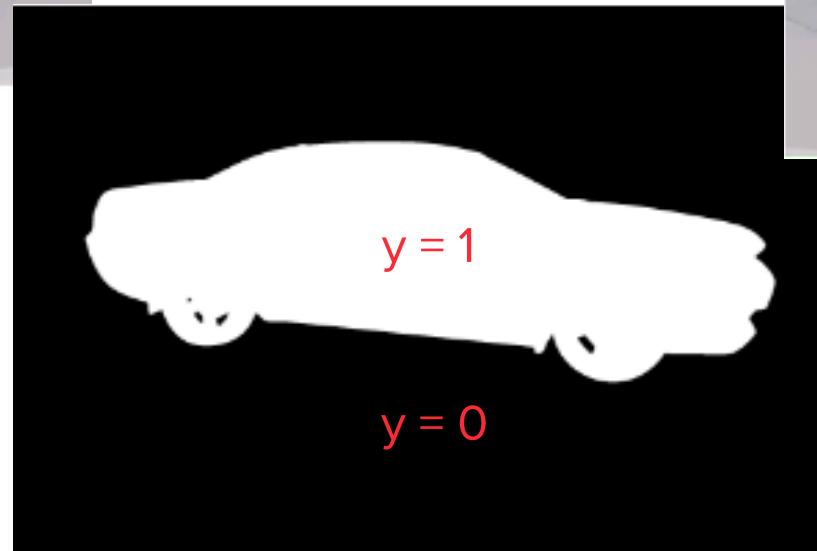


Семантическая сегментация

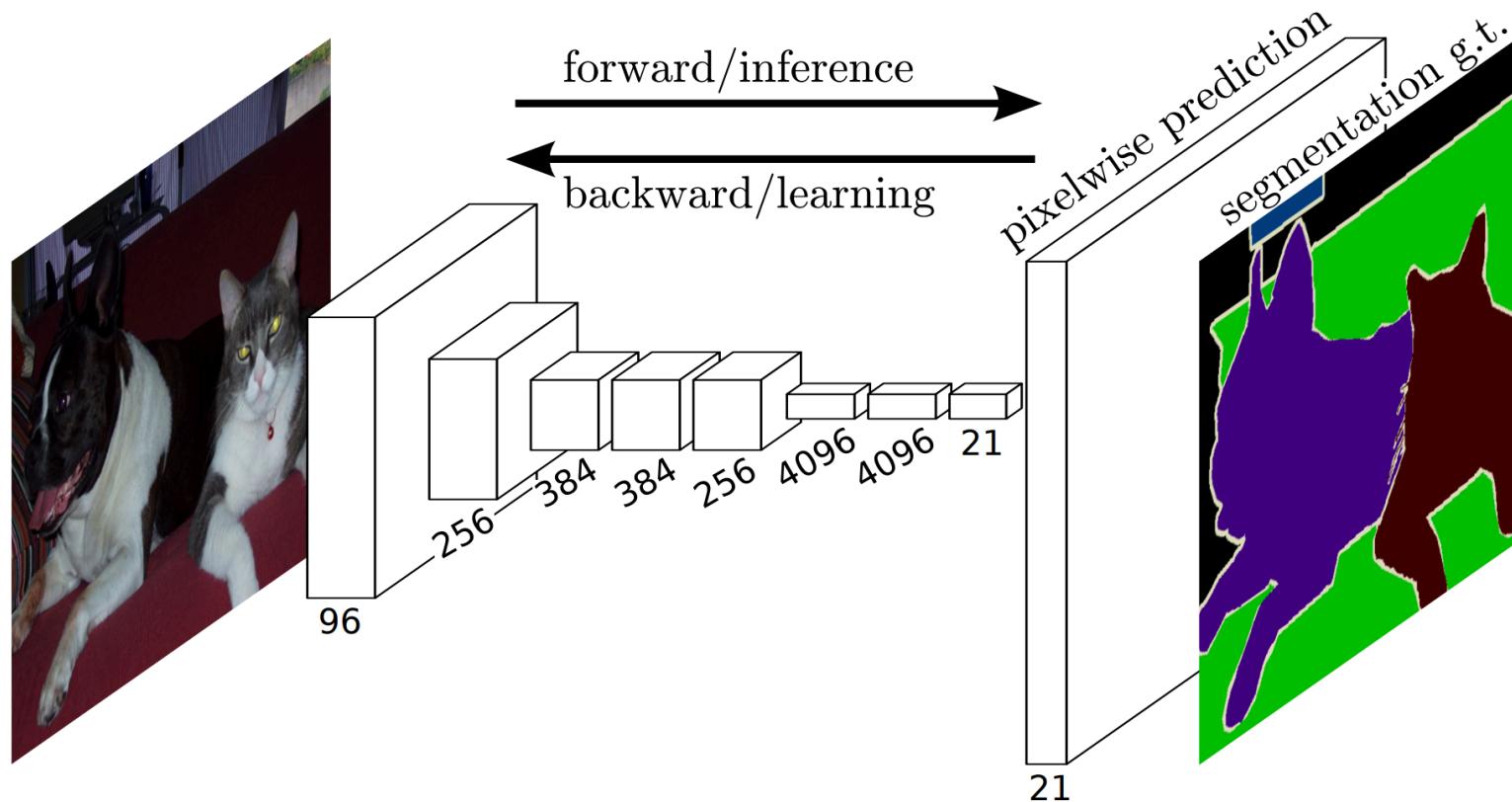


Исходная картинка
RGB

Бинарная маска
(таргет)



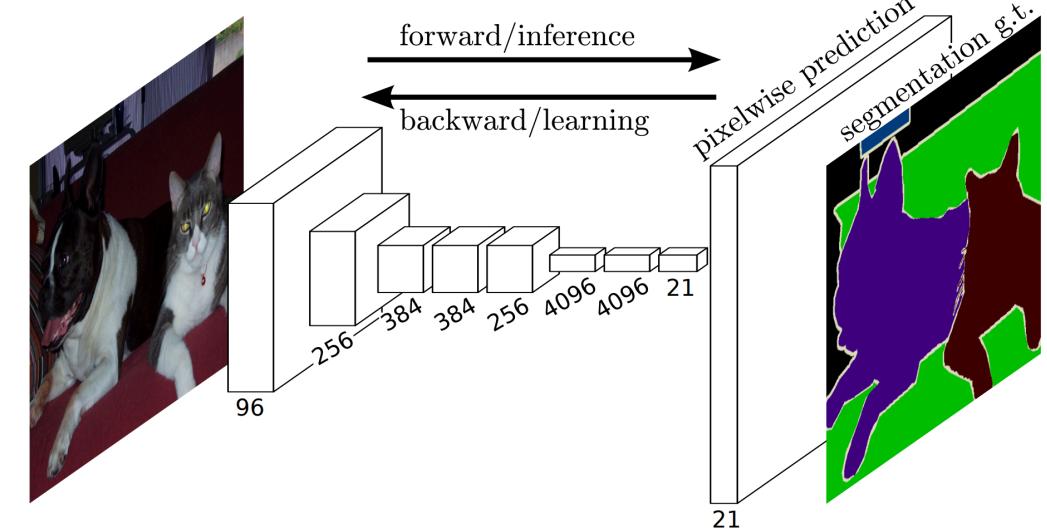
Fully-convolutional Network (FCN) (2014)



<https://arxiv.org/pdf/1411.4038.pdf>

Fully-convolutional Network (FCN) (2014)

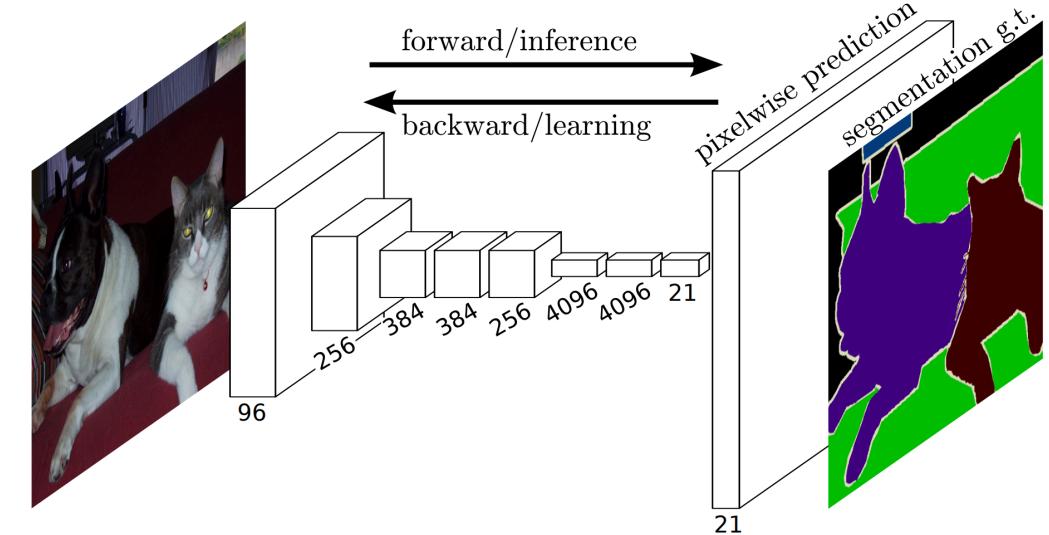
- В основе “обычная” сеть для классификации (напр. VGG)
- Убрали полносвязные слои в конце, заменили на Upsampling
- Размер выходного слоя = размеру входного изображения



<https://arxiv.org/pdf/1411.4038.pdf>

Fully-convolutional Network (FCN) (2014)

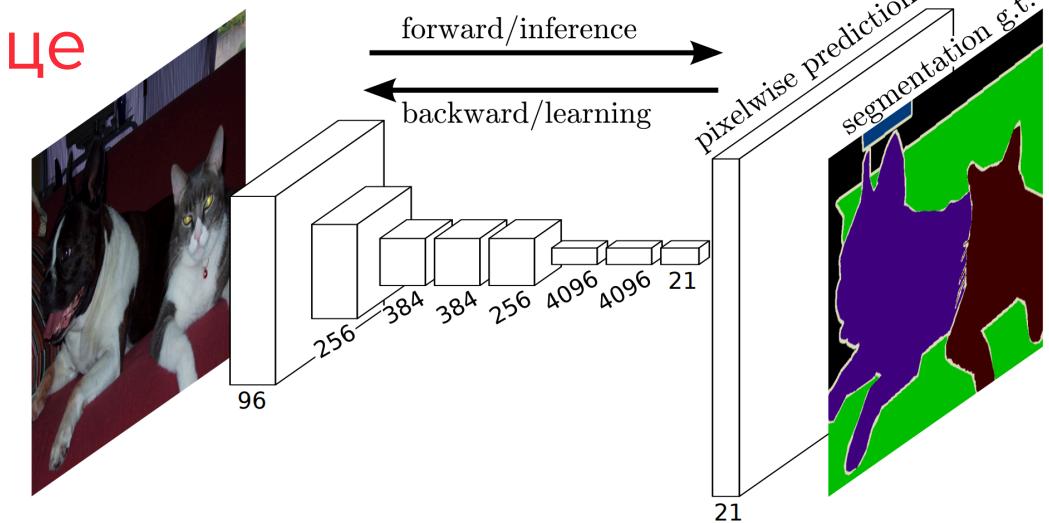
- В основе “обычная” сеть для классификации (напр. VGG)
- Убрали полносвязные слои в конце, заменили на Upsampling
- Размер выходного слоя = размеру входного изображения
- Недостатки?



<https://arxiv.org/pdf/1411.4038.pdf>

Fully-convolutional Network (FCN) (2014)

- В основе “обычная” сеть для классификации (напр. VGG)
- Убрали полносвязные слои в конце, заменили на Upsampling
- Размер выходного слоя = размеру входного изображения
- Недостатки?
 - Резкое увеличение размера в конце
 - Узкое горлышко для информации



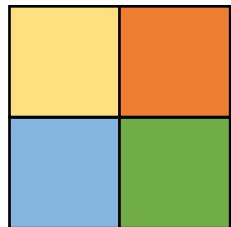
<https://arxiv.org/pdf/1411.4038.pdf>

- Как увеличить размер активационной карты?

- Как увеличить размер активационной карты?
 - Upsampling
 - Transposed Convolution

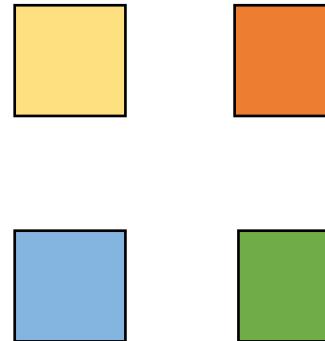
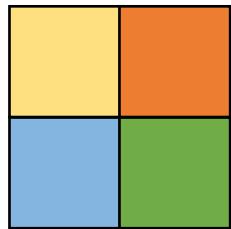


Upsampling





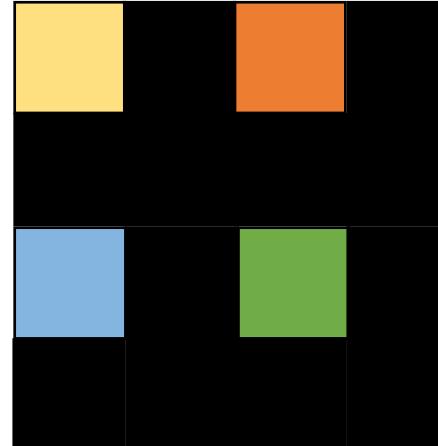
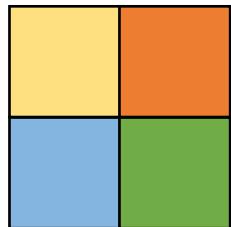
Upsampling





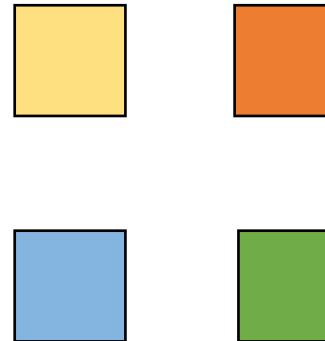
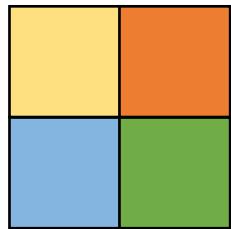
Upsampling

Без интерполяции

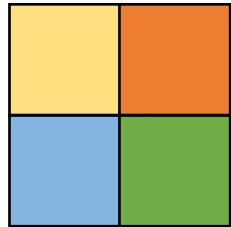




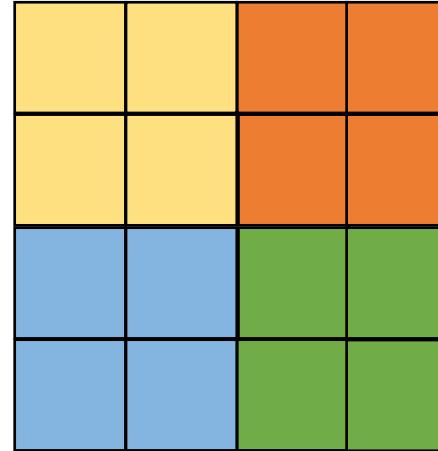
Upsampling



Upsampling

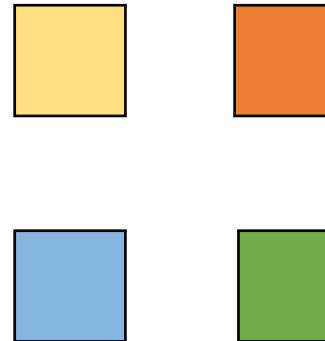
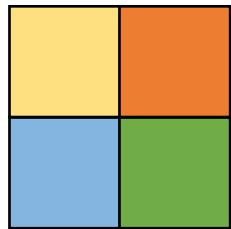


Интерполяция “ближайший сосед”
(nearest neighbour)

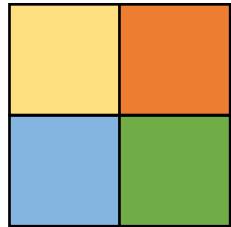




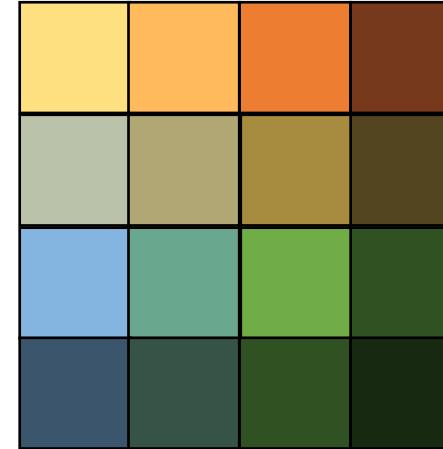
Upsampling



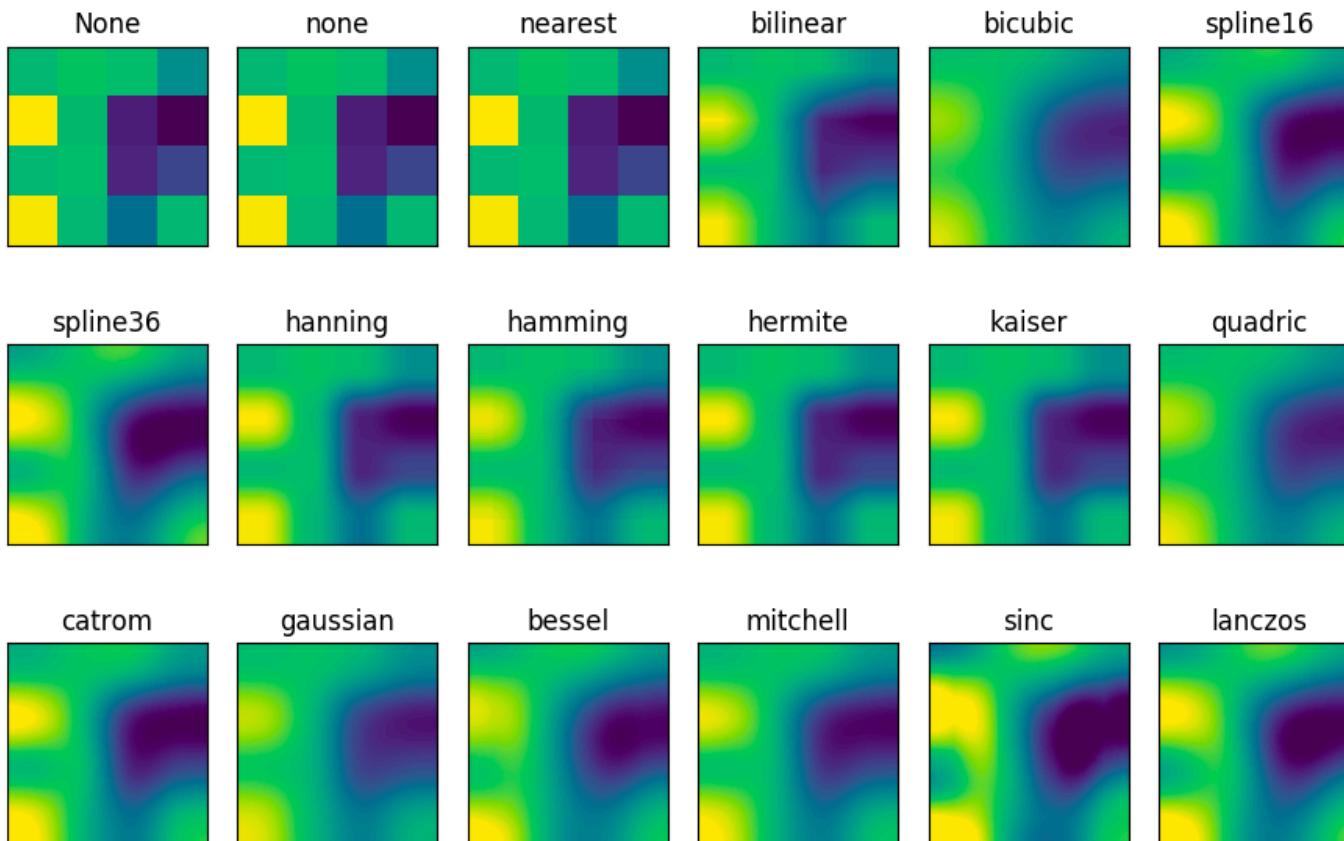
Upsampling



(Би)линейная интерполяция
(bilinear)



Upsampling



https://matplotlib.org/3.1.1/gallery/images_contours_and_fields/interpolation_methods.html

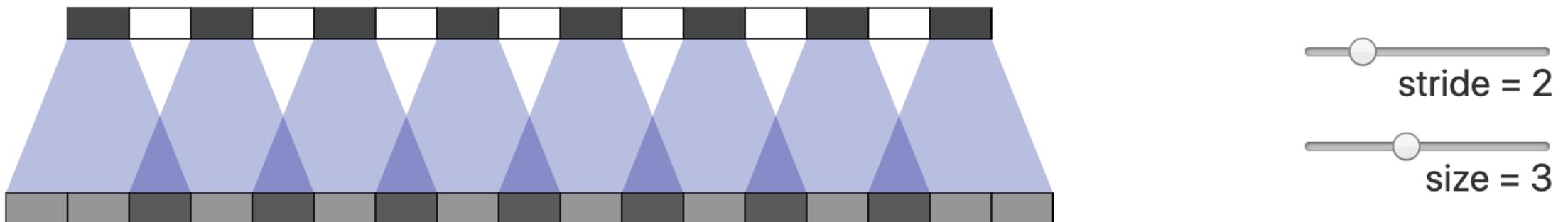
Transposed Convolution

- Исходные значения активаций = “веса” для коэффициентов ядра транспонированной свертки

Input	Kernel							Output			
$\begin{matrix} 0 & 1 \\ 2 & 3 \end{matrix}$	$\begin{matrix} 0 & 1 \\ 2 & 3 \end{matrix}$	$=$	$\begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix}$	$+$	$\begin{matrix} & 0 & 1 \\ & 2 & 3 \end{matrix}$	$+$	$\begin{matrix} & & \\ 0 & 2 & \\ 4 & 6 & \end{matrix}$	$+$	$\begin{matrix} & & \\ 0 & 3 & \\ 6 & 9 & \end{matrix}$	$=$	$\begin{matrix} 0 & 0 & 1 \\ 0 & 4 & 6 \\ 4 & 12 & 9 \end{matrix}$

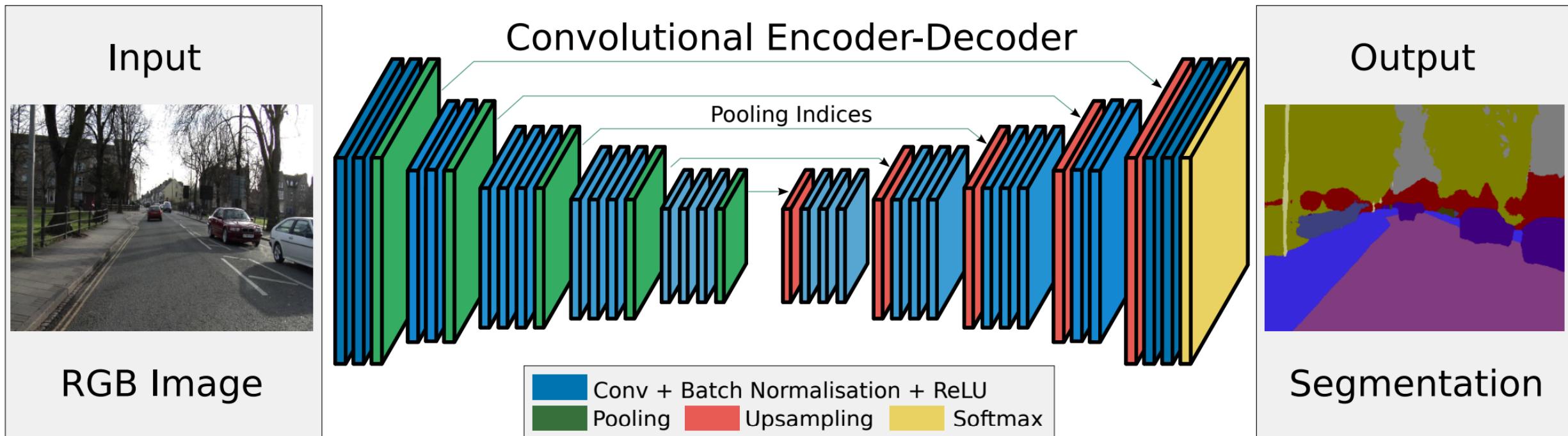
Transposed Convolution

- При различных наборах kernel_size / stride может возникать эффект “шахматной доски” из-за перекрытия рецептивных полей

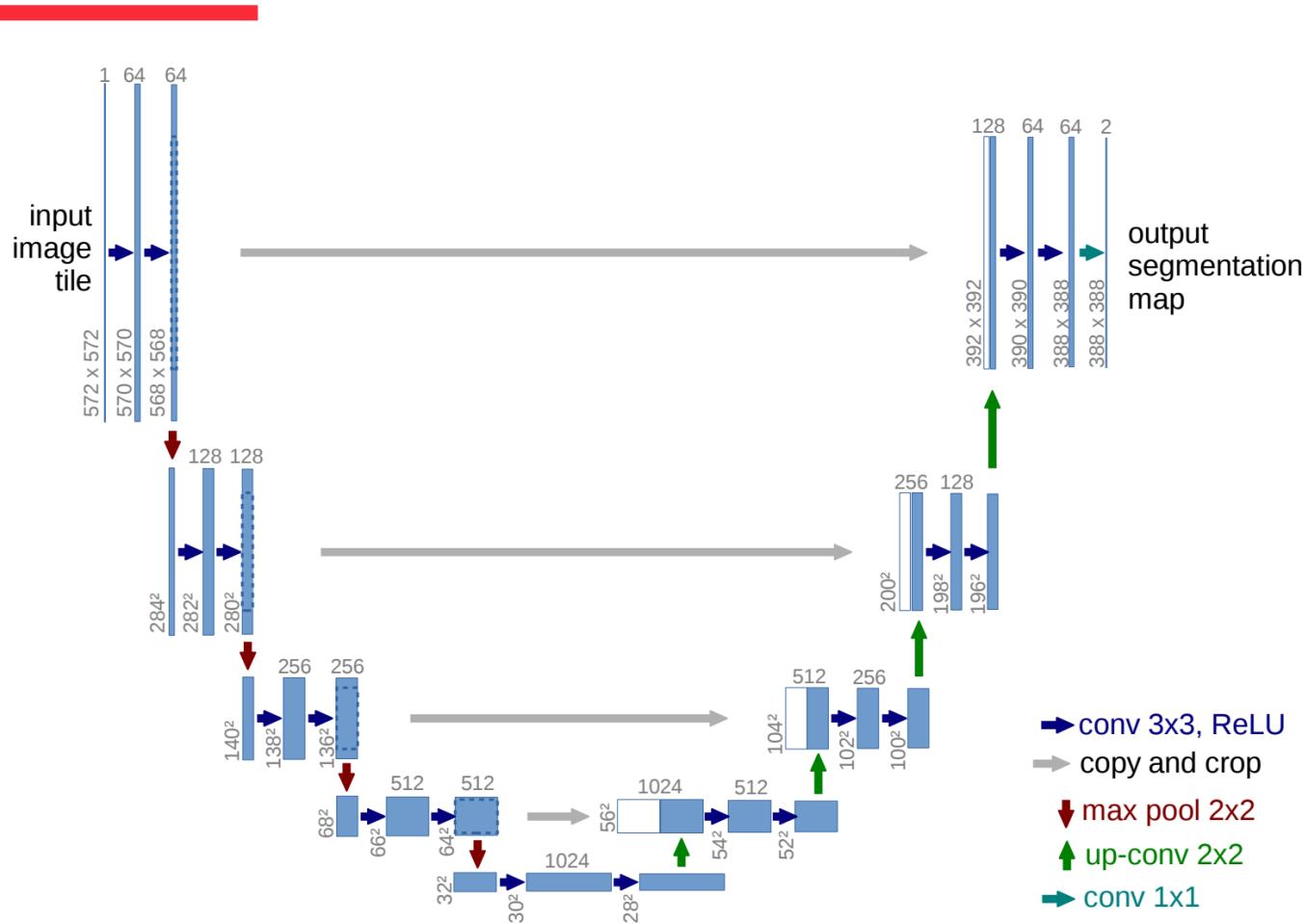


SegNet (2015)

- Сделали Upsampling постепенным

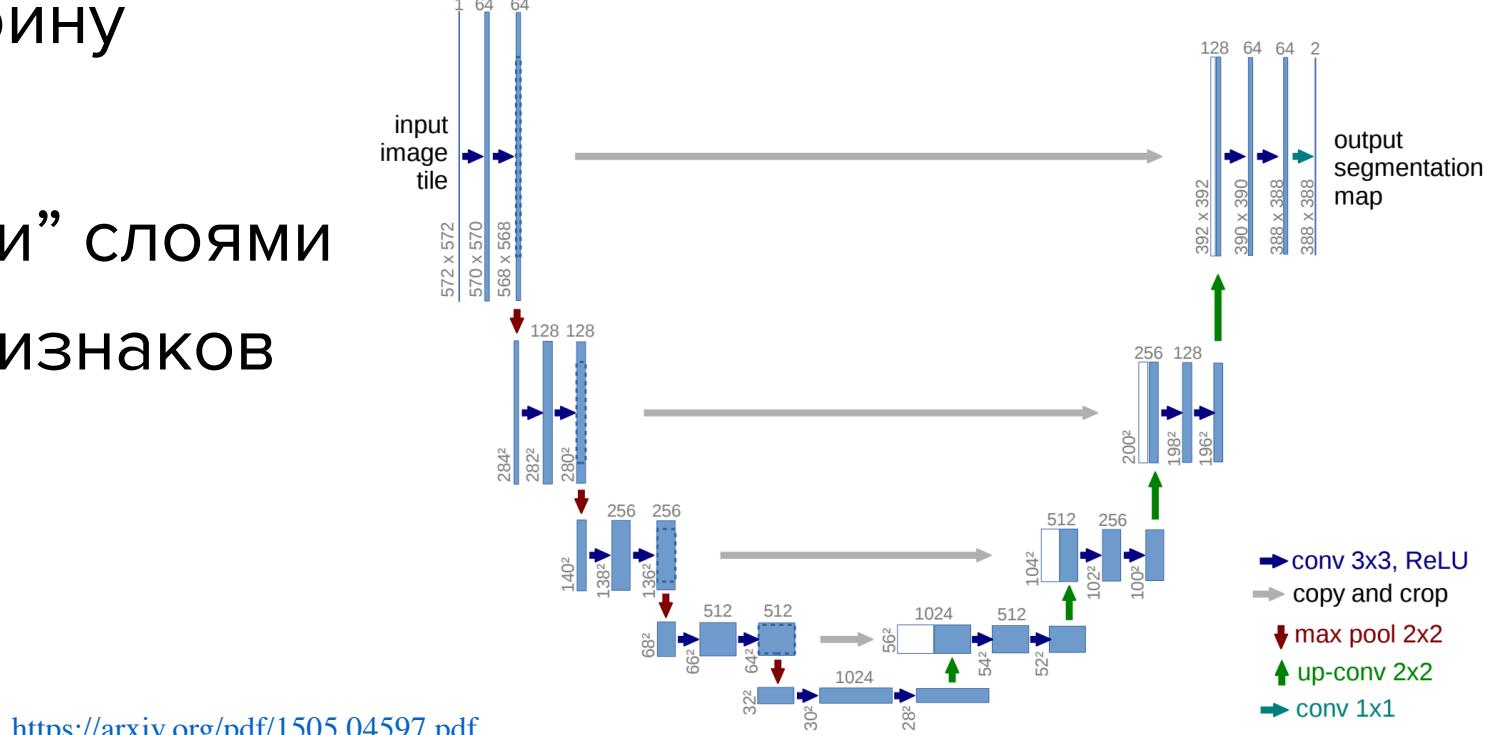


U-Net (2015)



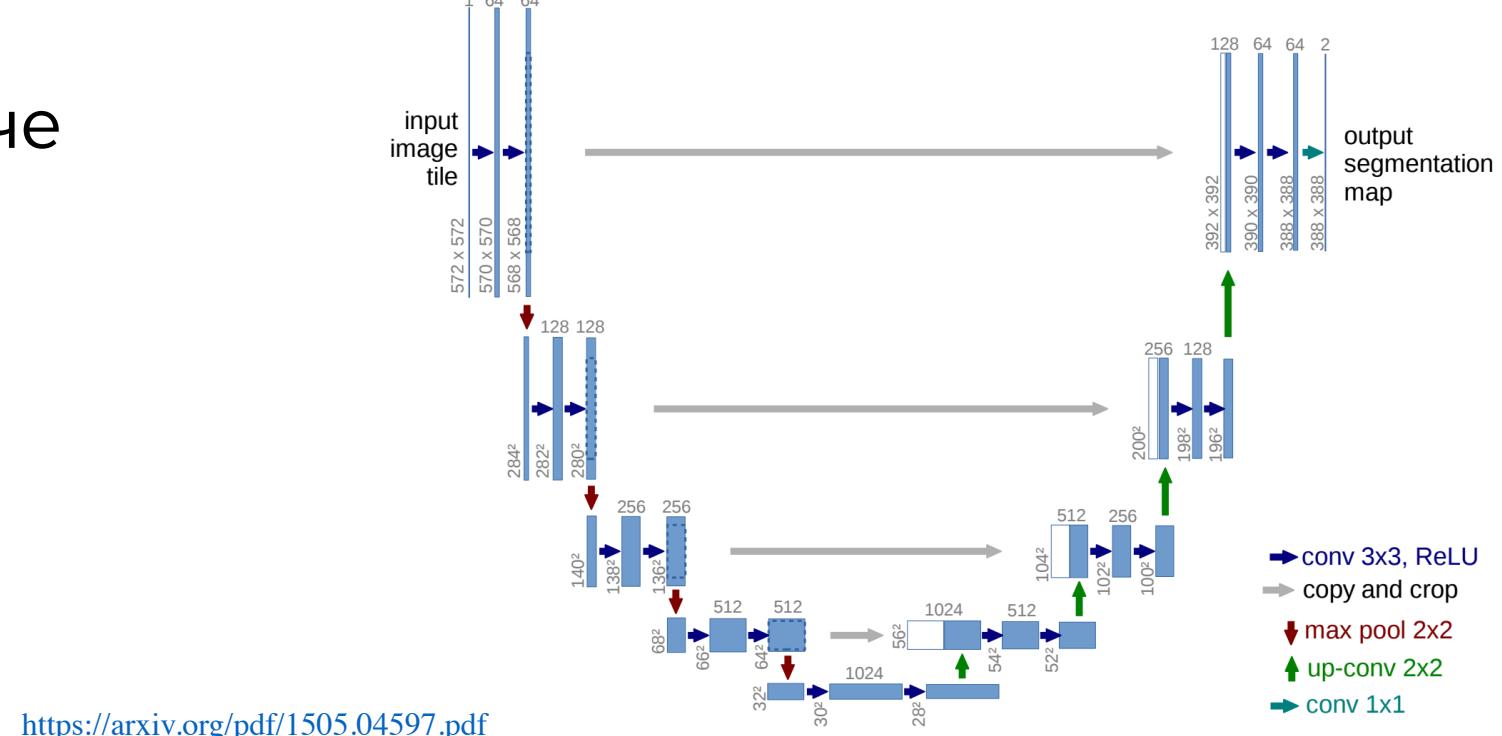
U-Net (2015)

- 2 половины = “encoder” & “decoder”
- Энкодер: тензор сжимается по пространственным размерностям и увеличивается в глубину
- Декодер: наоборот
- Между “симметричными” слоями происходит “склейка” признаков



U-Net (2015)

- Эффективнее обучение (30 изображений!)
- Четче границы
- Бейзлайн в любой сегментационной задаче



Feature Pyramid Networks (FPN) (2016)

- Добавили “принудительный” учет разных масштабов для предсказания

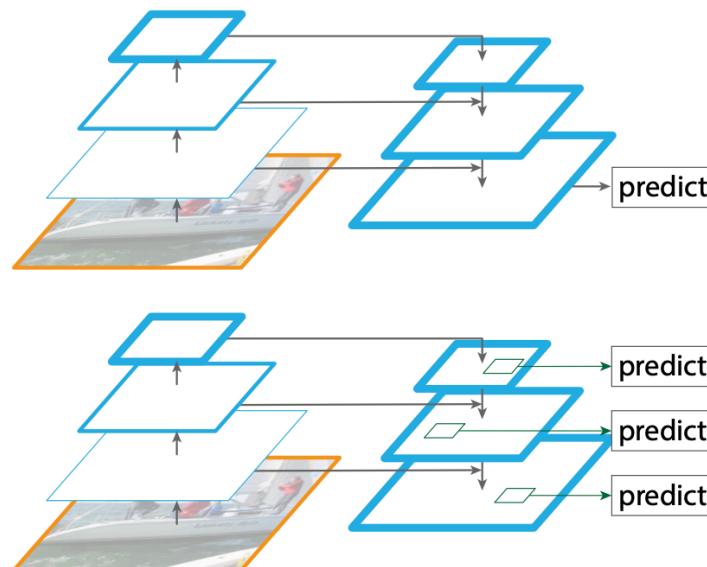
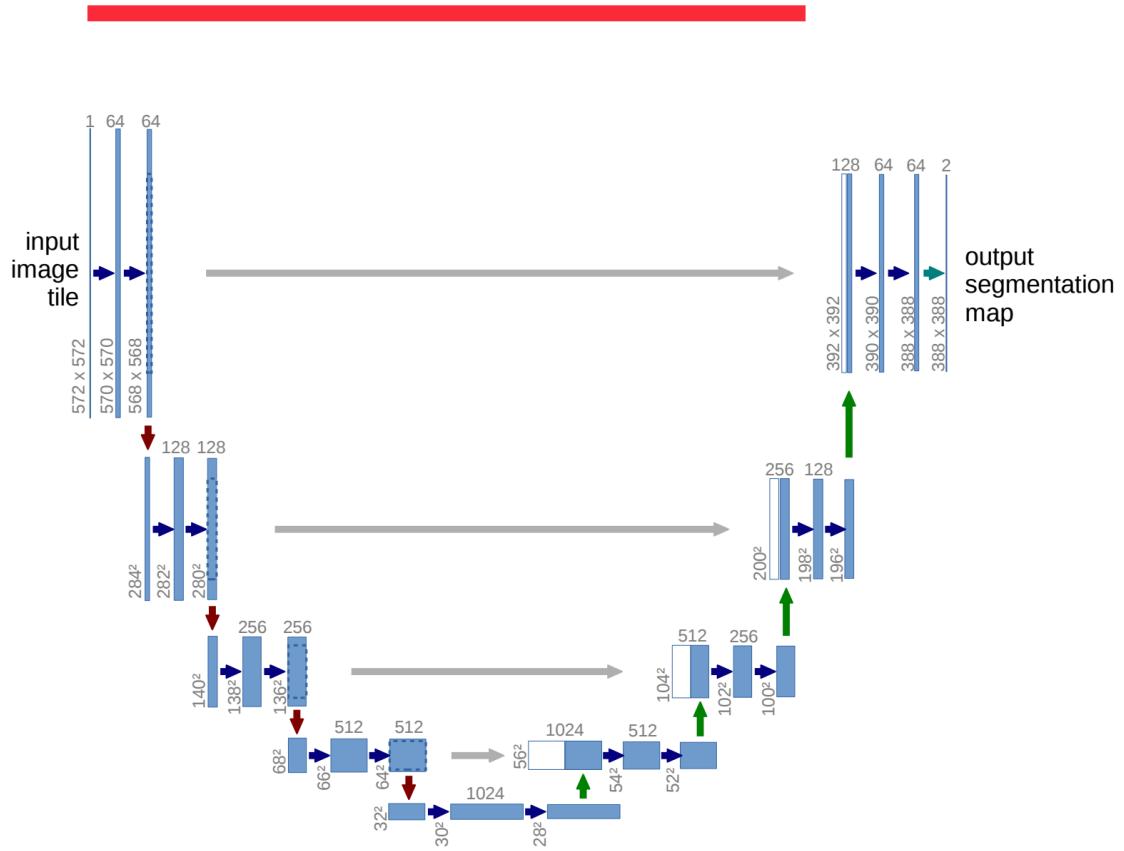
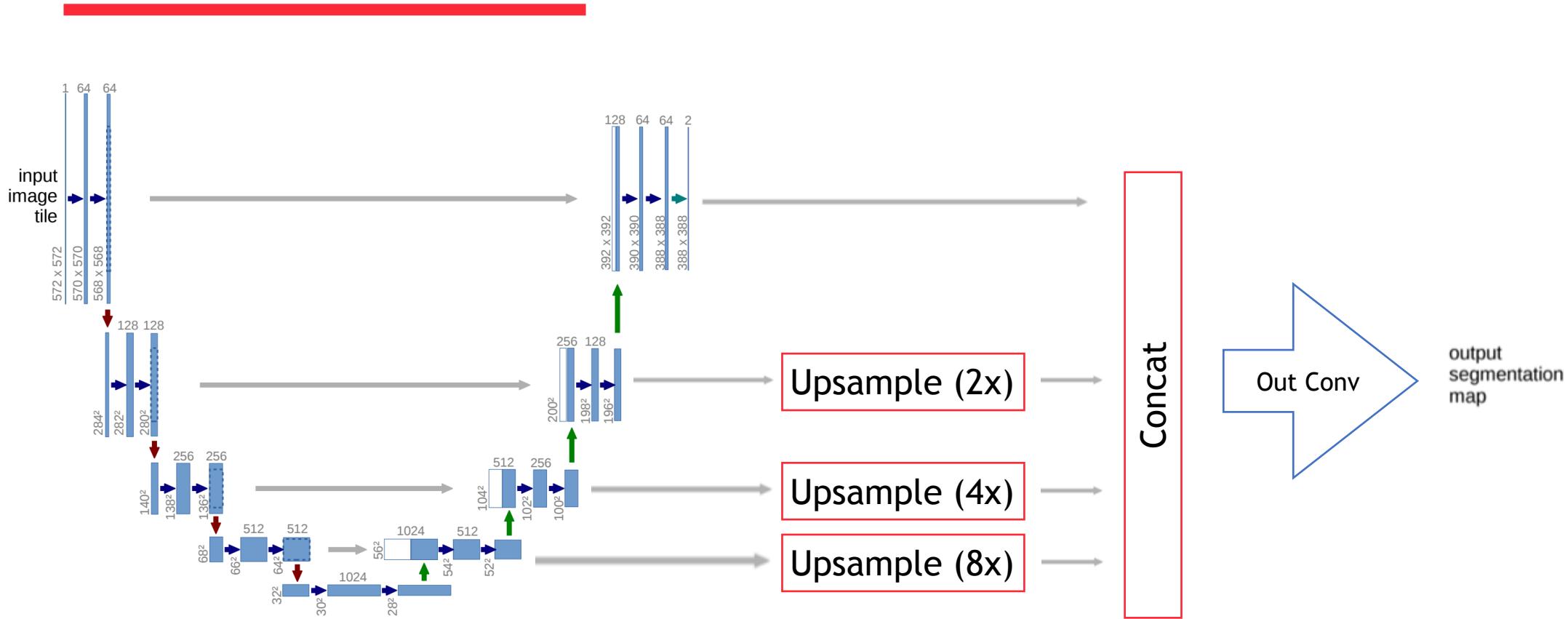


Figure 2. Top: a top-down architecture with skip connections, where predictions are made on the finest level (e.g., [28]). Bottom: our model that has a similar structure but leverages it as a *feature pyramid*, with predictions made independently at all levels.

U-Net



U-Net + FPN = <3

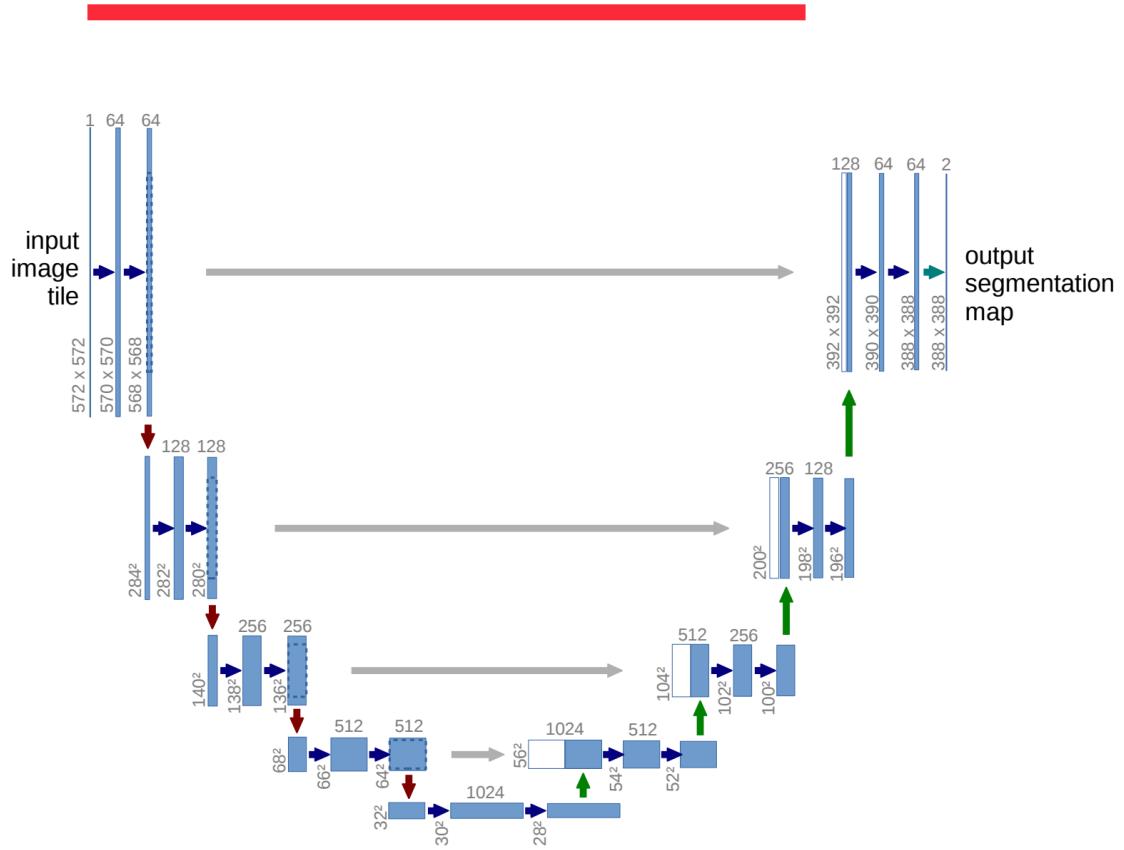


Сегментация в 2к20

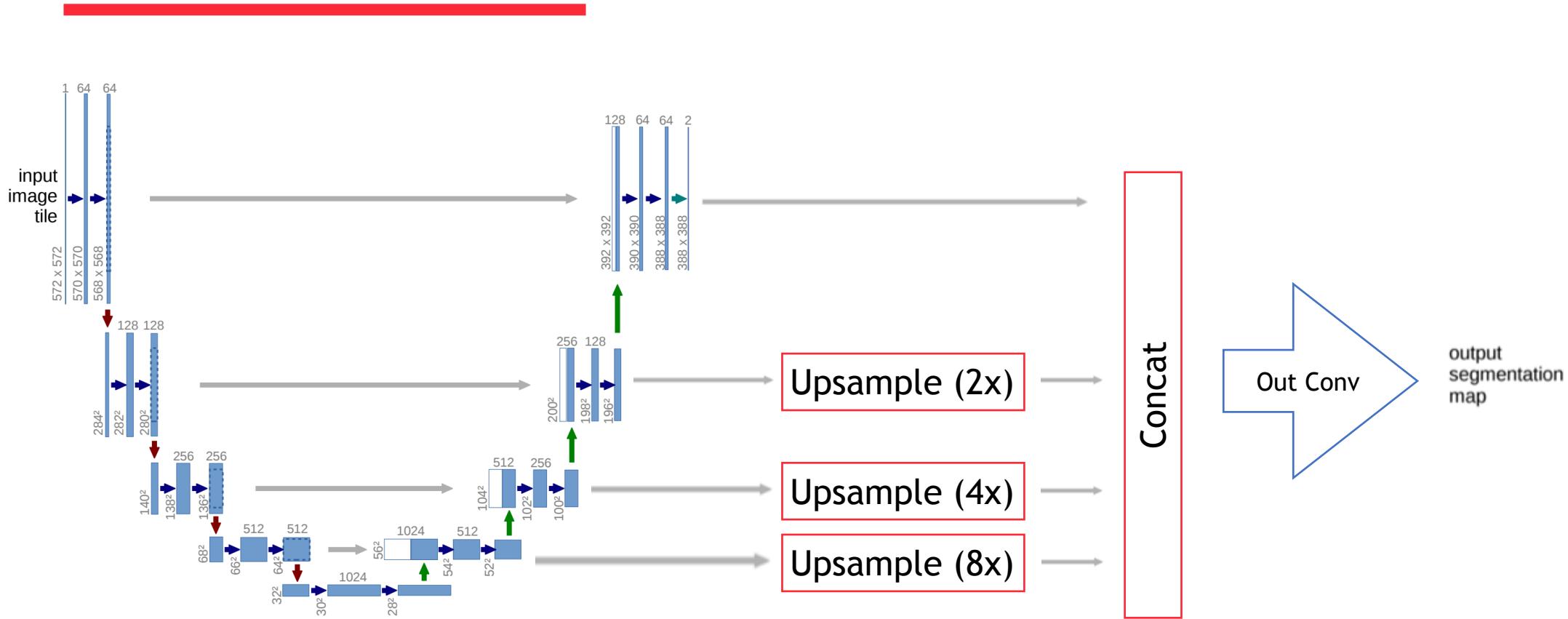
- https://github.com/qubvel/segmentation_models.pytorch

```
import segmentation_models_pytorch as smp  
  
model = smp.Unet('resnet34', encoder_weights='imagenet')
```

U-Net



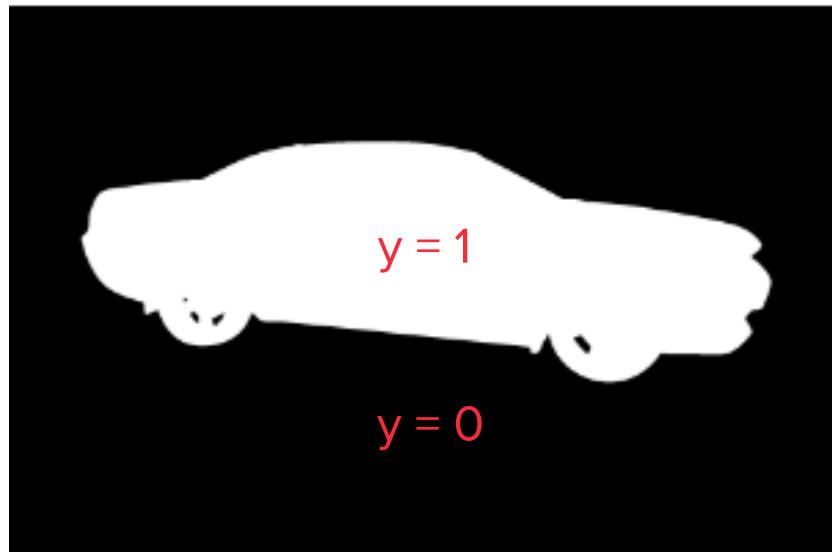
U-Net + FPN = <3



Losses: Cross-Entropy

- Семантическая сегментация ≈ пиксельная классификация
- Стартовый лосс: cross-entropy с усреднением по всем пикселями маски

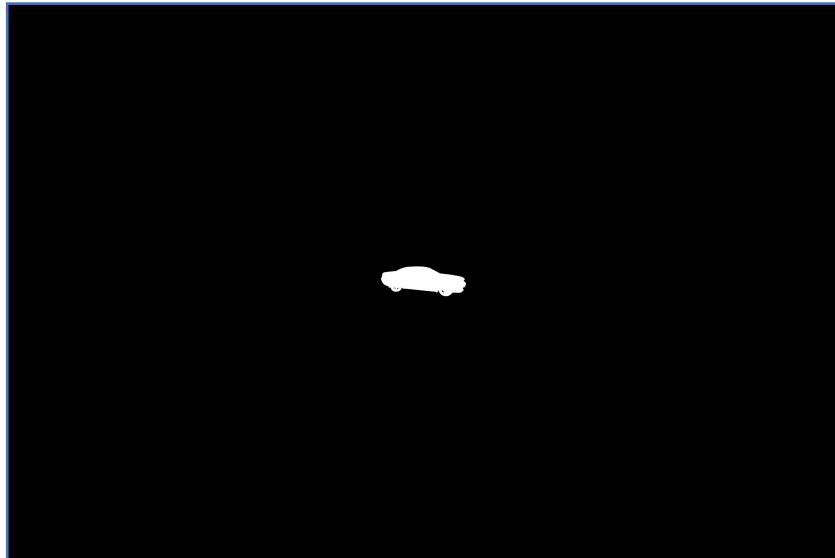
$$\text{CE}(p, \hat{p}) = -(p \log(\hat{p}) + (1 - p) \log(1 - \hat{p}))$$



Losses: Cross-Entropy

- Семантическая сегментация ≈ пиксельная классификация
- Стартовый лосс: cross-entropy с усреднением по всем пикселями маски
- Что будет, если объекты маленькие?

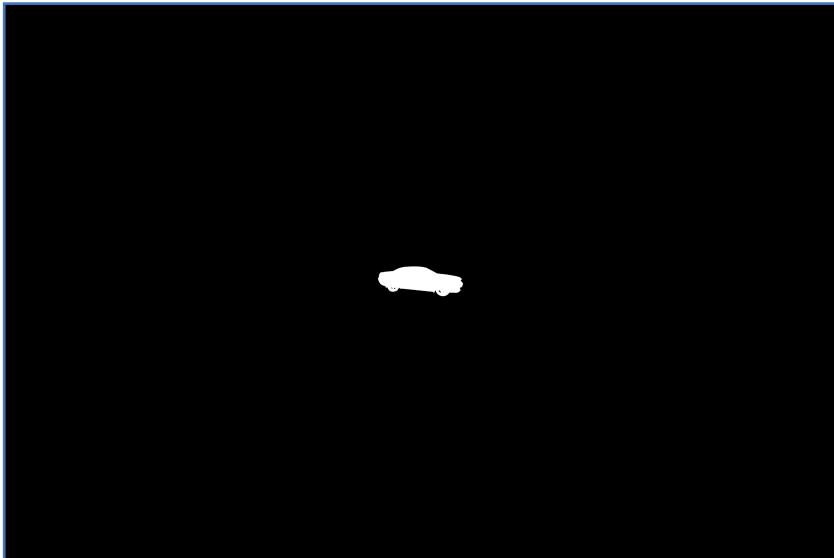
$$\text{CE}(p, \hat{p}) = -(p \log(\hat{p}) + (1 - p) \log(1 - \hat{p}))$$



Losses: Cross-Entropy

- Семантическая сегментация ≈ пиксельная классификация
- Стартовый лосс: cross-entropy с усреднением по всем пикселями маски
- Что будет, если объекты маленькие?

$$\text{CE}(p, \hat{p}) = -(p \log(\hat{p}) + (1 - p) \log(1 - \hat{p}))$$

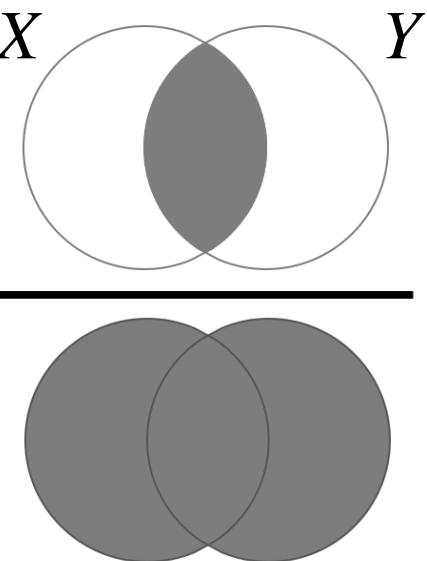


Надо учесть дисбаланс!

- Веса для классов
- Focal Loss

Losses: Jaccard, Dice

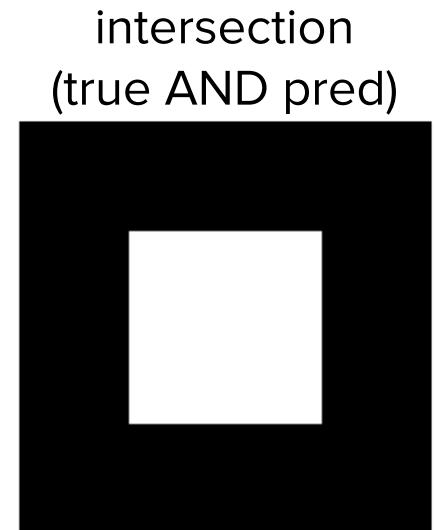
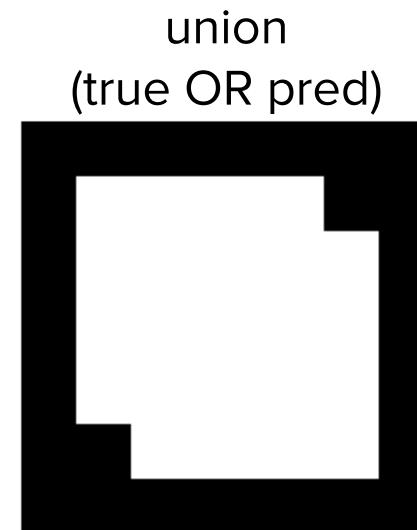
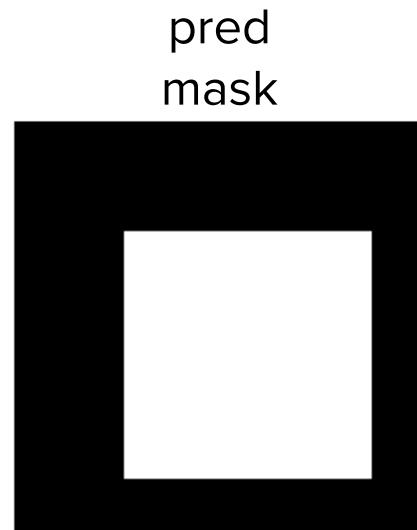
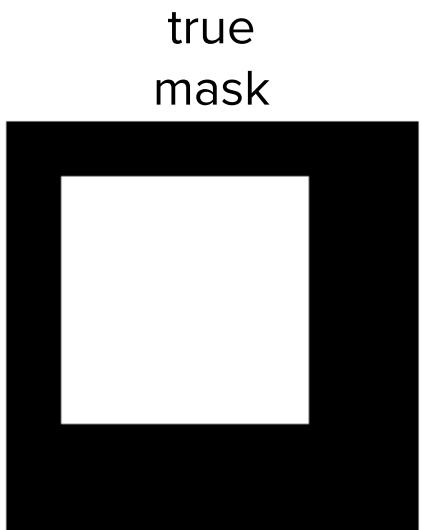
- Вспомним задачу детектирования: метрика Intersection-over-Union (IoU), она же Jaccard Index:

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


$$Jaccard(X, Y) = \frac{|X \wedge Y|}{|X \vee Y|}$$

Losses: Jaccard, Dice

- Intersection и Union в задаче сегментации:

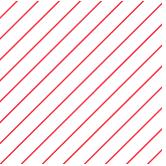


$$Jaccard(X, Y) = \frac{|X \wedge Y|}{|X \vee Y|}$$



Losses: Jaccard, Dice

- Величина $Jaccard(X, Y)$ недифференцируема = нельзя использовать при обучении



Losses: Jaccard, Dice

- Величина Jaccard(X, Y) недифференцируема = нельзя использовать при обучении
- ...Но можно ее аппроксимировать

Losses: Jaccard, Dice

- Величина Jaccard(X, Y) недифференцируема = нельзя использовать при обучении
- ...Но можно ее аппроксимировать
 - Предсказанная маска - пиксели от 0 до 1 (сигмоида)
 - Ground-true маска - бинарная (только 0 и 1)
- $$JaccardLoss = - \log \frac{X * Y}{X + Y - X * Y}$$
- $$DiceLoss = - \log 2 * \frac{X * Y}{X + Y}$$



Losses

- Часто используют комбинированный лосс (BCE + Dice, ...)
- См. еще один суррогатный лосс:
 - Lovasz-Softmax Loss: <https://arxiv.org/abs/1705.08790>



Семантическая сегментация

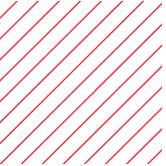
- Вопросы?

Проблемы с памятью



Проблемы с памятью

- Модели сегментации прожорливы
 - Кроме энкодера есть декодер
 - В некоторых случаях на входе картинки в высоком разрешении
 - Размер батча становится ограниченным - памяти GPU не хватает



Проблемы с памятью

- Модели сегментации прожорливы
 - Кроме энкодера есть декодер
 - В некоторых случаях на входе картинки в высоком разрешении
- Размер батча становится ограниченным - памяти GPU не хватает
 - Что плохого?



Проблемы нормализации

- BatchNorm плохо работает при небольших размерах батча



Проблемы нормализации

- BatchNorm плохо работает при небольших размерах батча
 - Искать альтернативы BatchNorm (InstanceNorm, ...)

Проблемы нормализации

- BatchNorm плохо работает при небольших размерах батча
 - Искать альтернативы BatchNorm (InstanceNorm, ...)
 - Увеличивать размер батча 

BatchNorm

- Вспомним слой BatchNorm:

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_1 \dots m\}$;

Parameters to be learned: γ, β

Output: $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad // \text{mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad // \text{mini-batch variance}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{normalize}$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) \quad // \text{scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

BatchNorm

- Вспомним слой BatchNorm:

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_1 \dots m\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad // \text{mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad // \text{mini-batch variance}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{normalize}$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) \quad // \text{scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

$$\frac{\partial \ell}{\partial \hat{x}_i} = \frac{\partial \ell}{\partial y_i} \cdot \gamma$$

$$\frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} = \sum_{i=1}^m \frac{\partial \ell}{\partial \hat{x}_i} \cdot (x_i - \mu_{\mathcal{B}}) \cdot \frac{-1}{2} (\sigma_{\mathcal{B}}^2 + \epsilon)^{-3/2}$$

$$\frac{\partial \ell}{\partial \mu_{\mathcal{B}}} = \left(\sum_{i=1}^m \frac{\partial \ell}{\partial \hat{x}_i} \cdot \frac{-1}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \right) + \frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} \cdot \frac{\sum_{i=1}^m -2(x_i - \mu_{\mathcal{B}})}{m}$$

$$\frac{\partial \ell}{\partial x_i} = \frac{\partial \ell}{\partial \hat{x}_i} \cdot \frac{1}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} + \frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} \cdot \frac{2(x_i - \mu_{\mathcal{B}})}{m} + \frac{\partial \ell}{\partial \mu_{\mathcal{B}}} \cdot \frac{1}{m}$$

$$\frac{\partial \ell}{\partial \gamma} = \sum_{i=1}^m \frac{\partial \ell}{\partial y_i} \cdot \hat{x}_i$$

$$\frac{\partial \ell}{\partial \beta} = \sum_{i=1}^m \frac{\partial \ell}{\partial y_i}$$

BatchNorm

- Вспомним слой BatchNorm:

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_1 \dots m\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad // \text{mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad // \text{mini-batch variance}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{normalize}$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) \quad // \text{scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

$$\frac{\partial \ell}{\partial \hat{x}_i} = \frac{\partial \ell}{\partial y_i} \cdot \gamma$$

$$\frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} = \sum_{i=1}^m \frac{\partial \ell}{\partial \hat{x}_i} \cdot (x_i - \mu_{\mathcal{B}}) \cdot \frac{-1}{2} (\sigma_{\mathcal{B}}^2 + \epsilon)^{-3/2}$$

$$\frac{\partial \ell}{\partial \mu_{\mathcal{B}}} = \left(\sum_{i=1}^m \frac{\partial \ell}{\partial \hat{x}_i} \cdot \frac{-1}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \right) + \frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} \cdot \frac{\sum_{i=1}^m -2(x_i - \mu_{\mathcal{B}})}{m}$$

$$\frac{\partial \ell}{\partial x_i} = \frac{\partial \ell}{\partial \hat{x}_i} \cdot \frac{1}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} + \frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} \cdot \frac{2(x_i - \mu_{\mathcal{B}})}{m} + \frac{\partial \ell}{\partial \mu_{\mathcal{B}}} \cdot \frac{1}{m}$$

$$\frac{\partial \ell}{\partial \gamma} = \sum_{i=1}^m \frac{\partial \ell}{\partial y_i} \cdot \hat{x}_i$$

$$\frac{\partial \ell}{\partial \beta} = \sum_{i=1}^m \frac{\partial \ell}{\partial y_i}$$

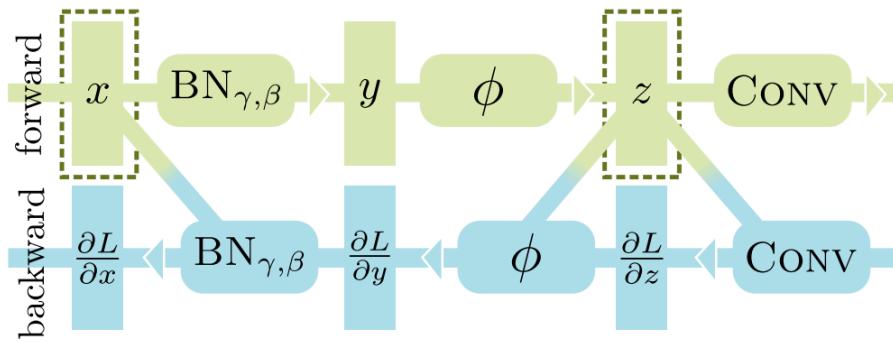
- Для вычисления градиентов слоя требуется входной тензор x
- При forward-pass он сохраняется в буфер
- При backward-pass он достается из буфера и подставляется в формулы



BatchNorm и костыли

- “Обычный” BatchNorm при forward-pass сохраняет входной тензор для вычисления градиентов при backward-pass
- Рассмотрим связку **BatchNorm + Activation + Convolution**

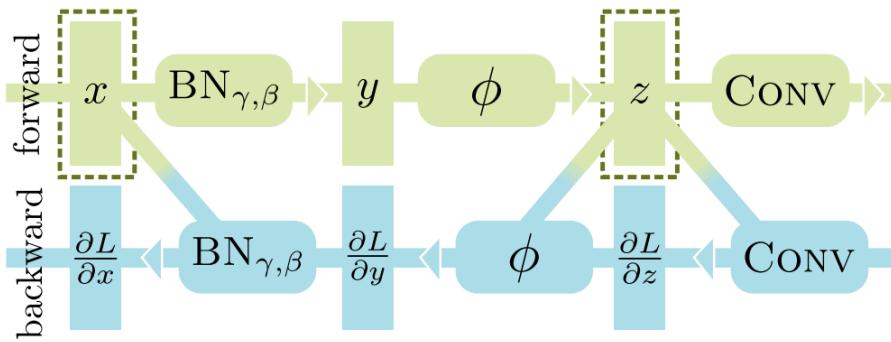
BatchNorm



(a) Standard building block (memory-inefficient)

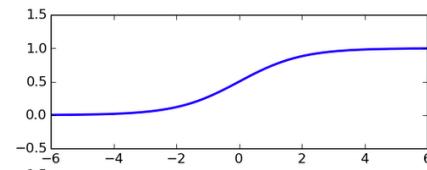
- Во время прямого прохода сохраняются переменные **x** и **z** (для вычисления градиентов при обратном проходе)
- **А как же y ?**

BatchNorm



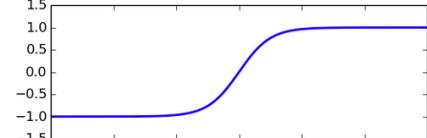
(a) Standard building block (memory-inefficient)

- Во время прямого прохода сохраняются переменные **х** и **z** (для вычисления градиентов при обратном проходе)
- **А как же y?**



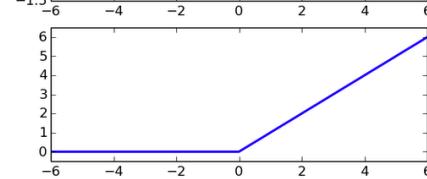
Sigmoid

$$\phi(z) = \frac{1}{1 + e^{-z}}$$



Hyperbolic Tangent

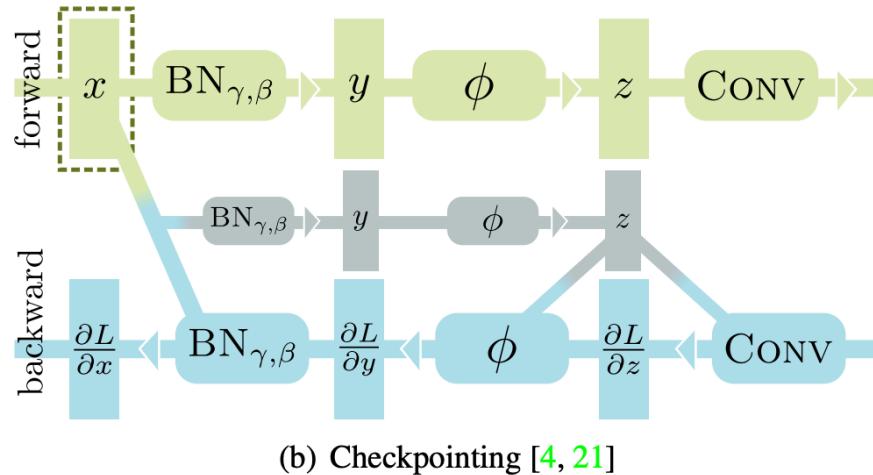
$$\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$



Rectified Linear

$$\phi(z) = \begin{cases} 0 & \text{if } z < 0 \\ z & \text{if } z \geq 0 \end{cases}$$

BatchNorm + Checkpoints



- При использовании т.н. чекпоинтов промежуточное значение **z** также не сохраняется, а вычисляется повторно при backward-pass
- Сэкономили память, но добавили вычислений

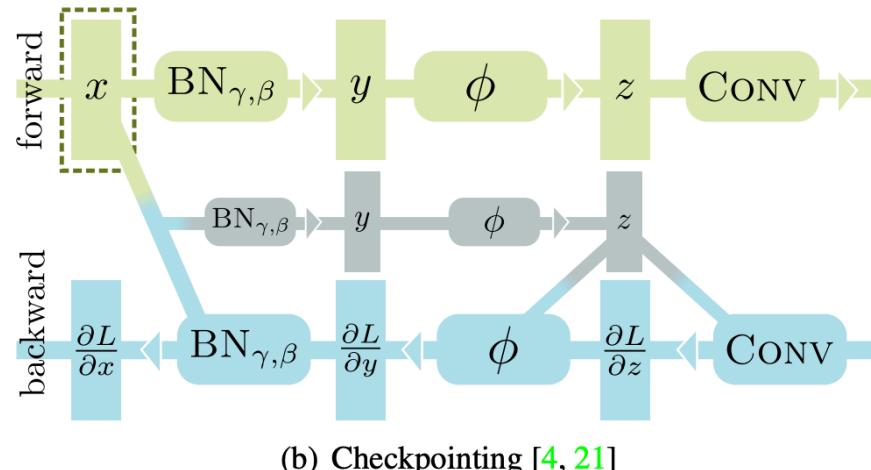


BatchNorm + Checkpoints

- Использование чекпоинтов осложнено тем, что для вычисления градиентов на более поздних слоях требуется “вызов” вычислений на более ранних (“нелокальность”)

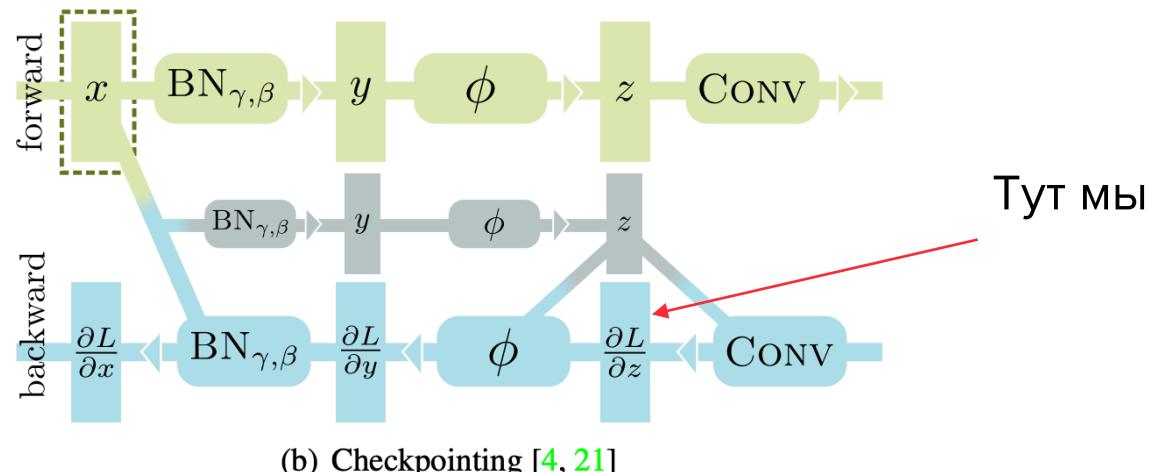
BatchNorm + Checkpoints

- Использование чекпоинтов осложнено тем, что для вычисления градиентов на более поздних слоях требуется “вызов” вычислений на более ранних (“нелокальность”)



BatchNorm + Checkpoints

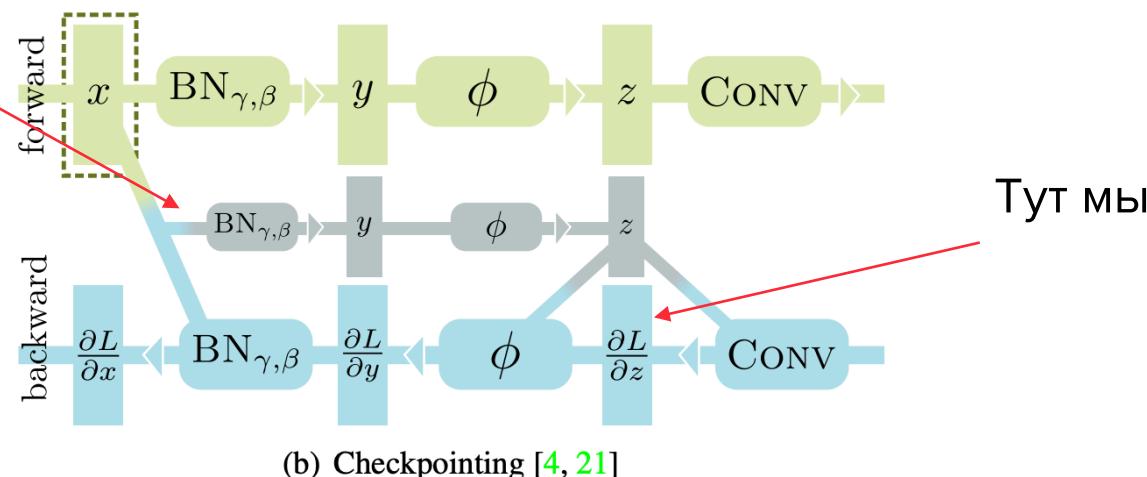
- Использование чекпоинтов осложнено тем, что для вычисления градиентов на более поздних слоях требуется “вызов” вычислений на более ранних (“нелокальность”)



BatchNorm + Checkpoints

- Использование чекпоинтов осложнено тем, что для вычисления градиентов на более поздних слоях требуется “вызов” вычислений на более ранних (“нелокальность”)

А тут необходимые
нам вычисления



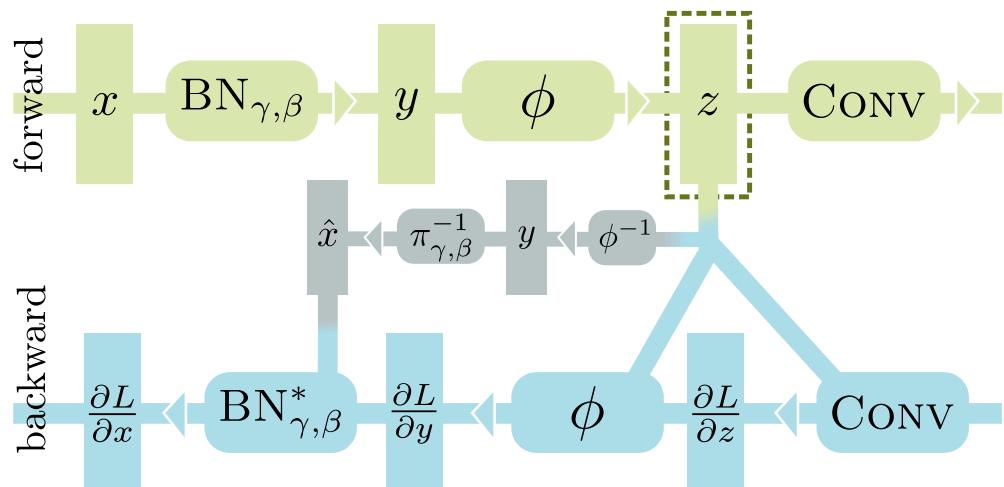


In-Place Activated BatchNorm

- Если функция активации (Φ) обратима, то достаточно хранить z
 - Вычисления в этом случае “локальны”

In-Place Activated BatchNorm

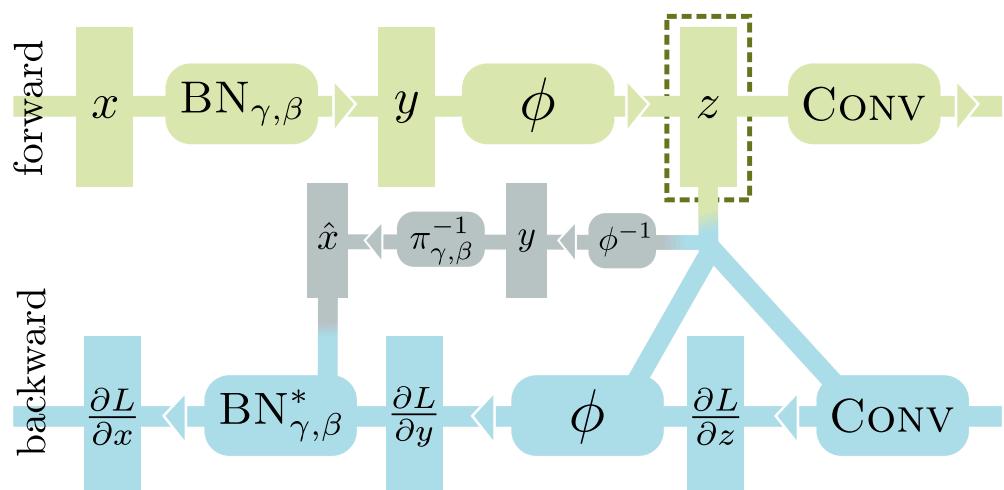
- Если функция активации (Φ) обратима, то достаточно хранить z
 - Вычисления в этом случае “локальны”



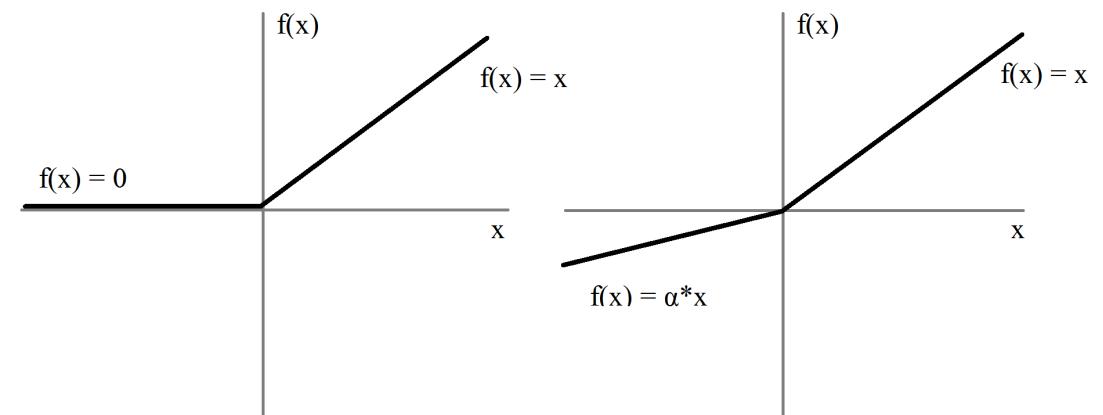
(d) In-Place Activated Batch Normalization I (proposed method)

In-Place Activated BatchNorm

- Если функция активации (Φ) обратима, то достаточно хранить z
 - Вычисления в этом случае “локальны”

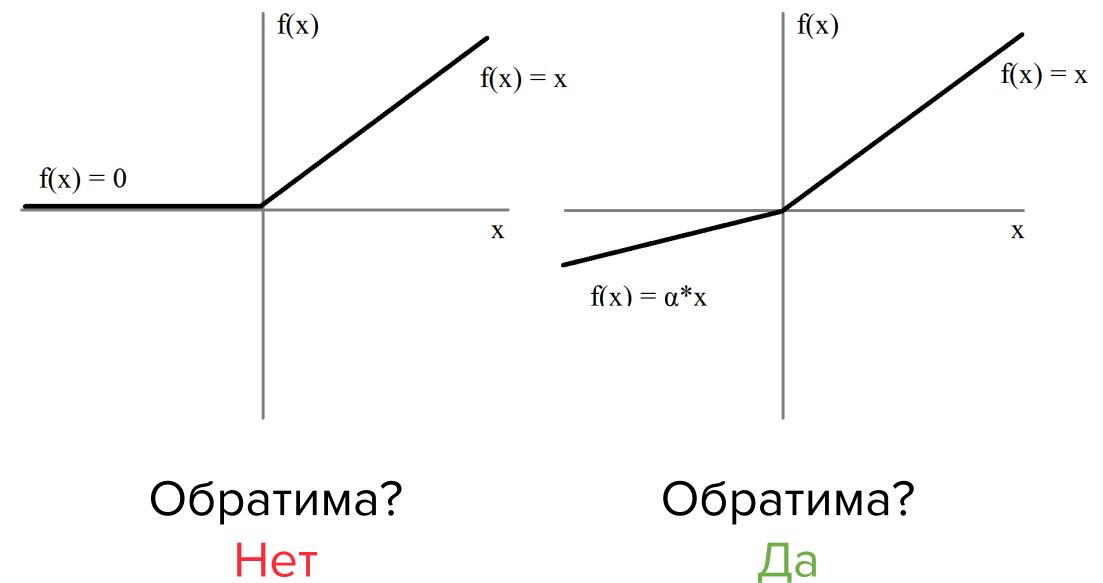
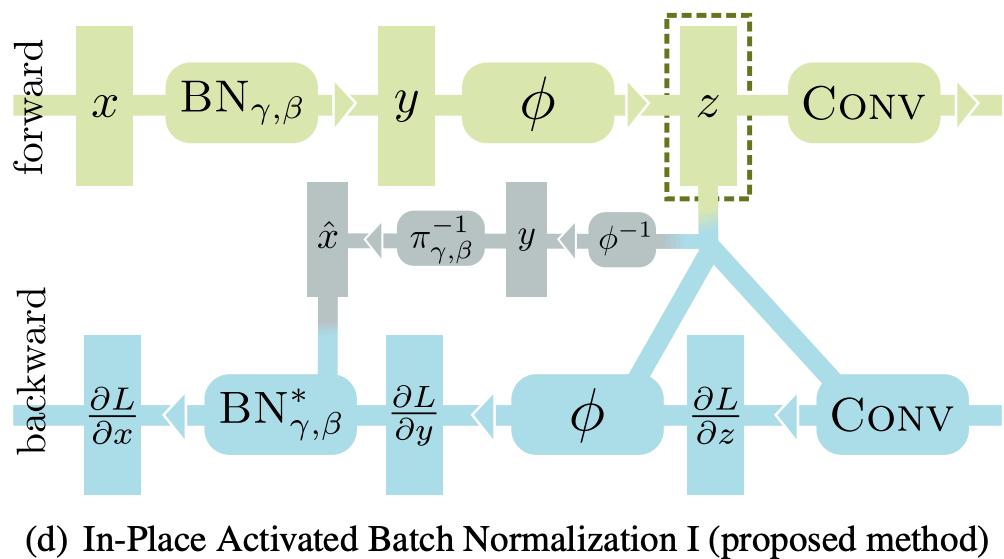


(d) In-Place Activated Batch Normalization I (proposed method)



In-Place Activated BatchNorm

- Если функция активации (Φ) обратима, то достаточно хранить z
 - Вычисления в этом случае “локальны”





In-Place Activated BatchNorm

- Утверждается, что можно сэкономить до 50% памяти
- Легко встроить (https://github.com/mapillary/inplace_abn)



In-Place ABN

- Вопросы?



Вместо итогов

- Сегментация - одна из наиболее востребованных задач в CV
- U-Net - хороший старт для семантической сегментации
- Для хорошей работы тяжелых моделей нужны оптимизации



В следующий раз

- Рекуррентные сети и CV