

Сегментация изображений

Виктор Китов

v.v.kitov@yandex.ru



Содержание

1 Введение

- Постановка задачи
- Меры качества
- Подходы решения задачи
- Повышение пространственного разрешения

2 Нейросетевые архитектуры

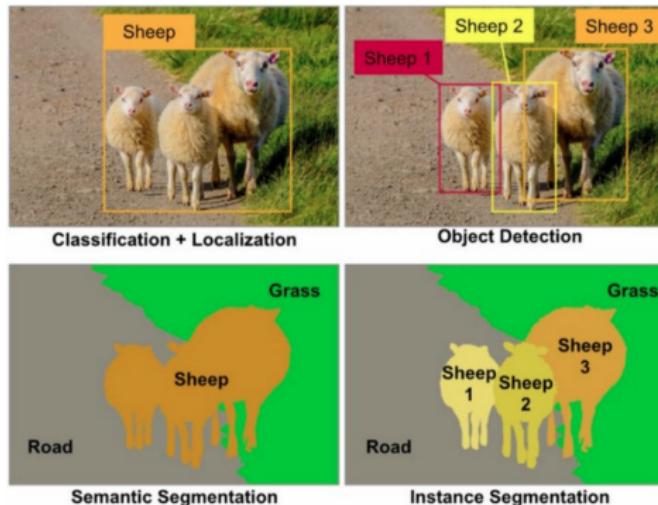
3 Instance и panoptic сегментация

1

Введение

- Постановка задачи
- Меры качества
- Подходы решения задачи
- Повышение пространственного разрешения

Типы задач



- Классификацию+локализацию можно выполнить, добавив к классификатору регрессионный вывод (x, y, h, w).
- Instance сегментацию можно решить, используя object detection+semantic segmentation.

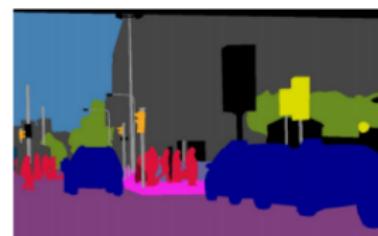
Panoptic segmentation: instance seg+сегментация фона



Image



Instance segmentation



Semantic segmentation



Panoptic segmentation

Применения

- Беспилотные системы (присутствует еще ограничение на скорость обработки)
 - self-driving cars: сегментировать людей, другие транспортные средства, знаки, дорожные препятствия
 - мониторинговые роботы: определить грузы на складе, их расположение и количество
- Изображения со спутника:
 - определить городские и сельскохозяйственные районы, дороги, транспортные средства, ход стройки
 - сегментировать поля с разными видами растений, их рост, найти лесные пожары, области загрязнений
- Медицина:
 - сегментировать кости, ткани, заболевания.
- Понимание сцены и событий на ней, замена фона.

Выход задачи сегментации¹



Input

Segmented

- 1: Person
 - 2: Bench
 - 3: Plant/Grass
 - 4: Cat

Semantic Labels

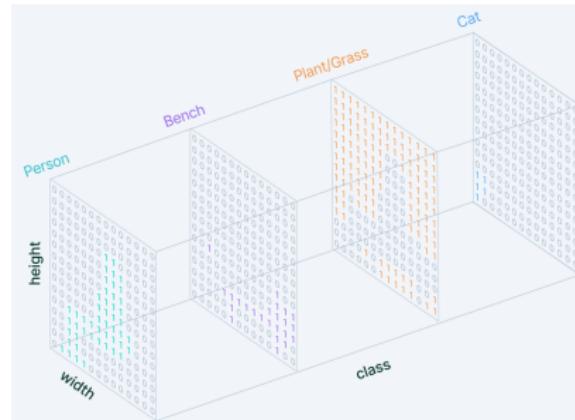
3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	1	1	3	3	3	3
3	3	3	3	3	3	3	1	1	1	3	3	3	3
3	3	3	3	3	3	3	1	1	1	3	3	3	3
3	3	3	3	3	3	3	1	1	1	1	3	3	3
3	3	3	3	3	3	3	1	1	1	1	3	3	3
3	3	2	3	3	3	1	1	1	1	1	2	3	3
3	3	1	1	1	1	1	1	1	1	1	1	2	2
3	3	1	1	1	1	1	1	1	1	1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

¹<https://www.v7labs.com/blog/semantic-segmentation-guide>

Выход задачи сегментации

Для каждого пикселя (i, j) : y - one-hot класс, \hat{y} - вероятности классов. Настройка модели (\uparrow логарифма правдоподобия $\Leftrightarrow \downarrow$ кросс-энтропии):

$$\sum_{n=1}^N \sum_{c=1}^C \mathbb{I}[y_n = c] \ln p_\theta(y = c|x_n) \rightarrow \max_{\theta}$$



1

Введение

- Постановка задачи
- **Меры качества**
- Подходы решения задачи
- Повышение пространственного разрешения

Меры качества: один класс

- Y - истинное выделение класса, \hat{Y} - предсказанное выделение.
- Точность (accuracy) - завышает качество, когда объект мал ($|(\neg Y) \cap (\neg \hat{Y})| \gg 1$)

$$\frac{|Y \cap \hat{Y}| + |(\neg Y) \cap (\neg \hat{Y})|}{W \cdot H}$$

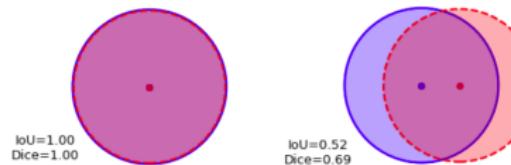
Меры качества: один класс³

- Мера intersection-over-union (IoU)=близость Жаккарда:

$$\text{IoU} = \text{Jaccard} = \frac{|\hat{Y} \cap Y|}{|\hat{Y} \cup Y|} = \frac{TP}{TP + FP + FN} \in [0, 1]$$

- Мера Dice=F-мера²

$$\text{Dice} = \frac{2 |\hat{Y} \cap Y|}{|\hat{Y}| + |Y|} = \frac{2 TP}{2 TP + FP + FN} = \frac{1}{\frac{1}{2} \frac{TP}{\hat{P}} + \frac{1}{2} \frac{TP}{P}} \in [0, 1]$$



IoU=1.00
Dice=1.00

IoU=0.52
Dice=0.69

²Докажите.

³<https://ilmonteux.github.io/2019/05/10/segmentation-metrics.html>

Меры качества: один класс

- Меры связаны монотонно⁴:

$$Dice = \frac{2IoU}{IoU + 1}$$

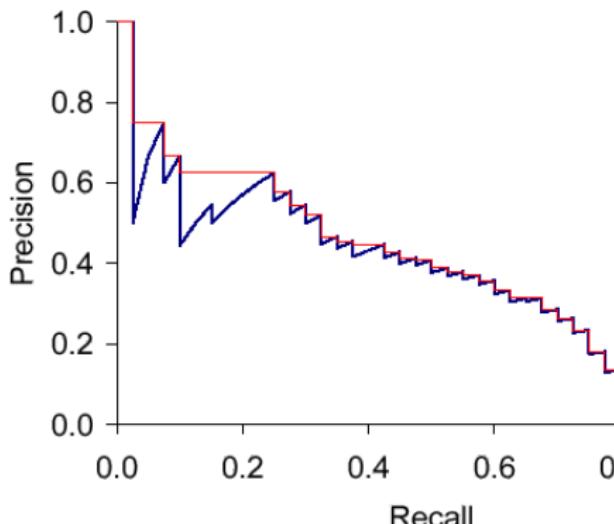
- По сравнению с IoU мера Dice меньше штрафует несоответствия для в целом верно классифицированных объектов (с большим TP):

$$Dice = \frac{TP + TP}{TP + TP + FP + FN}$$

⁴Докажите.

Меры качества: один класс

- Можно задать порог α , с которого начинается распознавание класса, считать $\text{Precision}(\alpha)$, $\text{Recall}(\alpha)$, построить $\text{Prec}(\text{Recall})$ посчитать плотность под графиком (реально он сглаживается).



Оптимизация Dice напрямую

- Проблема cross-entropy loss: мало объекта и много фона.
 - вариант решения: взвешивание по редкости классов.
- Другой вариант - оптимизировать меры качества напрямую.
- В архитектуре V-net⁵:
 - выходы после SoftMax p_i , бинарная истинная разметка g_i , i -позиция.
- Модель оптимизируется по Dice:

$$D = \frac{2 \sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2} \quad \frac{\partial D}{\partial p_j} = 2 \left[\frac{g_j \left(\sum_i^N p_i^2 + \sum_i^N g_i^2 \right) - 2 p_j \left(\sum_i^N p_i g_i \right)}{\left(\sum_i^N p_i^2 + \sum_i^N g_i^2 \right)^2} \right]$$

⁵<https://arxiv.org/pdf/1606.04797.pdf>

Качество определения границ⁶

- Важный показатель-качество вокруг границ.
- Пусть $B_r(Y)$ - полоса ширины r вокруг границы маски Y ,
- Trimap IoU: IoU в окрестности истинных границ Y :

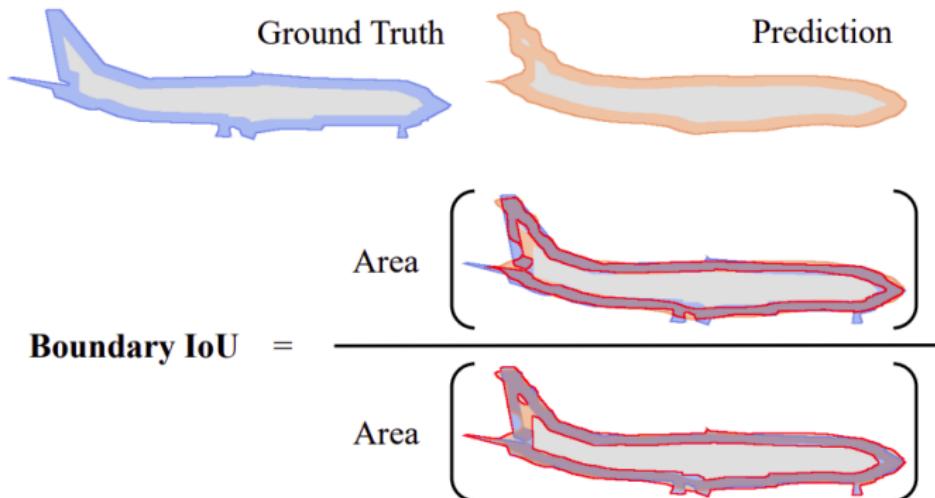
$$\text{Trimap IoU} = \frac{|B_r(Y) \cap \hat{Y} \cap Y|}{|(B_r(Y) \cap \hat{Y}) \cup (B_r(Y) \cap Y)|}$$

- недостатки: несимметрична, не смотрит ошибки границы \hat{Y} за пределами границ $B_r(Y)$
- Boundary IoU: симметрична, смотрим пересечение прогноза на границе прогноза с фактом на границе факта.

$$\text{Boundary IoU} = \frac{|(B_r(\hat{Y}) \cap \hat{Y}) \cap (B_r(Y) \cap Y)|}{|(B_r(\hat{Y}) \cap \hat{Y}) \cup (B_r(Y) \cap Y)|}$$

⁶<https://arxiv.org/pdf/2103.16562.pdf>

Boundary IoU



Меры качества: C классов⁷

- Матрица ошибок (confusion matrix) $M_{ij} = \#\{y = i \& \hat{y} = j\}$

$$G_i = \sum_j M_{ij} \text{ - \#пикселей класса } i \text{ (ground truth)}$$

$$P_j = \sum_i M_{ij} \text{ - \#прогнозов класса } j \text{ (predicted)}$$

- Overall pixel accuracy (OP) - микроусреднение на классах:

$$\frac{\sum_{i=1}^C M_{ii}}{\sum_{i=1}^C G_i} \text{ - доминируется классом 'фон'}$$

- Per-Class (PC) accuracy - макроусреднение на классах:

$$\frac{1}{C} \sum_{i=1}^C \frac{M_{ii}}{G_i} \text{ - выгоднее 'фон' относить к редким классам}$$

⁷<http://www.bmva.org/bmvc/2013/Papers/paper0032/paper0032.pdf>

Меры качества: C классов

- Индекс Жаккарда (Jaccard index) - макроусредненный intersection over union

$$\frac{1}{C} \sum_{i=1}^C \frac{M_{ii}}{G_i + P_i - M_{ii}} = \frac{1}{C} \sum_{i=1}^C \frac{|\hat{y} = i \text{ and } y = i|}{|\hat{y} = i \text{ or } y = i|}$$

- Macro-averaged Dice:

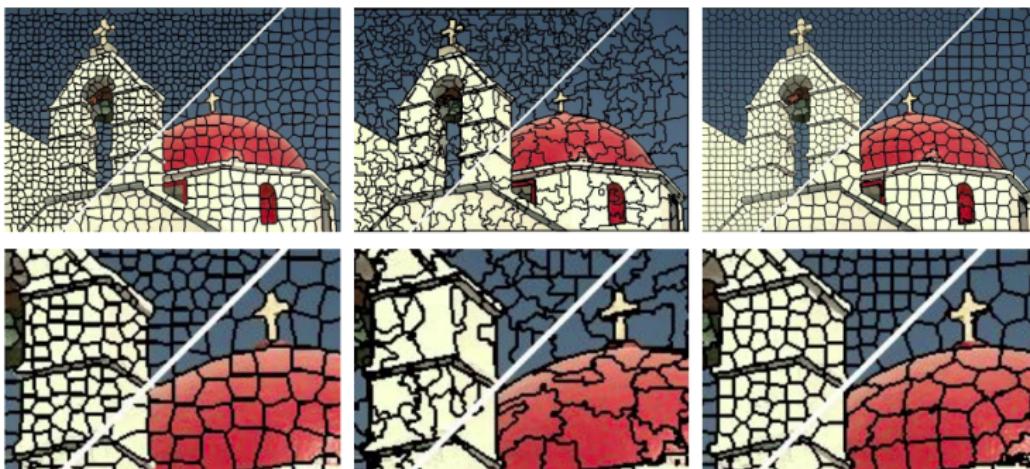
$$\frac{1}{C} \sum_{i=1}^C \frac{2M_{ii}}{G_i + P_i} = \frac{1}{C} \sum_{i=1}^C \frac{2 |\hat{y} = i \text{ and } y = i|}{|y = i| + |\hat{y} = i|}$$

1 Введение

- Постановка задачи
- Меры качества
- **Подходы решения задачи**
- Повышение пространственного разрешения

SLIC⁸: сегментация, основанная на суперпиксели

- Алгоритм SLIC позволяет разбить изображение на суперпиксели (близкие блоки с примерно похожими цветами), кластеризуя в $(x, y, color)$ пространстве.



⁸https://www.iro.umontreal.ca/~mignotte/IFT6150/Articles/SLIC_Superpixels.pdf

Алгоритм SLIC

1. Изображение переводится в цветовое пространство CIELAB.
 - в котором Евклидово расстояние \approx воспринимаемой цветовой разнице.
2. Инициализируются центры K кластеров по равномерной сетке $\{(x_0^k, y_0^k)\}_{k=1}^K$.
 - если $N = \#\text{пикселей}$, то $N/K = \#\text{пикселей в кластере}$, $S = \sqrt{N/K}$ - его сторона
3. Центры смещаются, чтобы обеспечить минимум перепада цветов вдоль вертикальной и горизонтальной оси в окрестности $3 \times 3 \Omega(x_0^k, y_0^k)$.
 - т.е.

$$\|I(x+1, y) - I(x-1, y)\|_2^2 + \|I(x+1, y) - I(x, y-1)\|_2^2 \rightarrow \min_{x, y \in \Omega(x_0^k, y_0^k)}$$
 - чтобы сместиться, если центроид попадает на границу или шумовой пиксель

Алгоритм SLIC

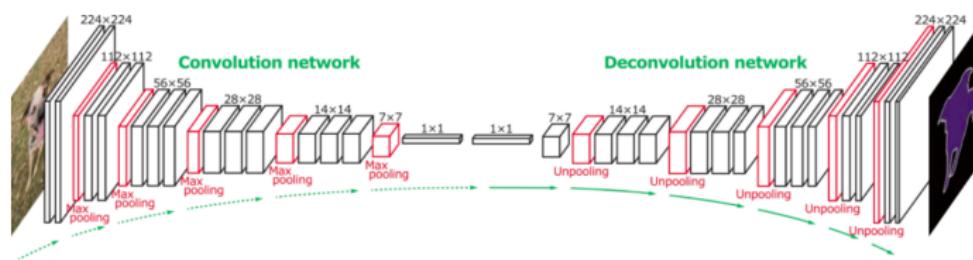
4. В цикле до сходимости:

- ① для каждого центроида производится распределение окружающих его пикселей между центроидами в (l, a, b, x, y) в окрестности $(\pm 2S, \pm 2S)$
- ② обновляются расположения центроидов кластеров

5. Постобработка: если обнаружены несвязные области, отнесенные к одному центроиду, они присоединяются к ближайшему соседнему кластеру.

Базовая нейросетевая архитектура сегментации

- Поскольку $\hat{Y} \in \mathbb{R}^{H \times W}$, используется следующая архитектура:



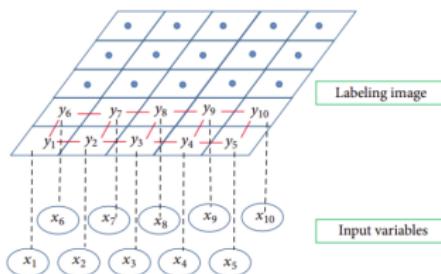
- Кодировщик: свёртки извлекают все более сложные признаки
 - разрешение \downarrow за счёт свёрток и пулингов
 - разумно инициализировать предобученными первыми слоями CNN
 - можно потом донастроить под сегментацию
- Декодировщик постепенно \uparrow разрешение.

Базовая нейросетевая архитектура сегментации

- Это полносвёрточная архитектура (fully convolutional architecture)
- Архитектура применима к изображениям произвольных размеров с любым соотношением сторон
 - но веса свёрток обучаются на (и далее ожидают) некоторое привычное разрешение
- Нужен способ \uparrow пространственное разрешение.

Условные случайные поля¹⁰

Условные случайные поля (conditional random fields⁹): метод надстройки стандартных моделей, позволяющий учесть взаимосвязи между соседними прогнозами $Y_{i,j}$ и $Y_{u,v}$.



⁹ Введение в условные случайные поля.

¹⁰ <https://downloads.hindawi.com/journals/mpe/2016/3846125.pdf>

Условные случайные поля

$$E(\mathbf{X}, \mathbf{Y}) = \sum_{i,j} D(X_{i,j}, Y_{i,j}) + \sum_{(i,j)} \sum_{(u,v) \in \mathcal{N}(i,j)} V(Y_{i,j}, Y_{u,v}) \rightarrow \min_{\mathbf{Y}}$$

- $(i,j), (u, v)$ -пространственные позиции, $\mathcal{N}(i,j)$ - окрестность позиций, связанных с (i,j)
- $E(\mathbf{X}, \mathbf{Y})$ - энергия, $D(X_{i,j}, Y_{i,j})$ - связь $X_{i,j}$ и $Y_{i,j}$ (модель классификации), $V(Y_{i,j}, Y_{u,v})$ - связь соседних меток.
- Энергия связана с распределением данных:

$$p(\mathbf{Y}|\mathbf{X}) \propto e^{-E(\mathbf{X}, \mathbf{Y})} \rightarrow \max_{\mathbf{Y}}$$

1 Введение

- Постановка задачи
- Меры качества
- Подходы решения задачи
- Повышение пространственного разрешения

Повышение пространственного разрешения (upsampling)

↑ пространственного разрешения (upsampling): расширение входа, затем обычная свертка.

- заполнение нулями (bed of nails):

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & a & 0 & b & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & c & 0 & d & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- перемасштабирование ближайшим соседом:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \rightarrow \begin{pmatrix} a & a & b & b \\ a & a & b & b \\ c & c & d & d \\ c & c & d & d \end{pmatrix}$$

- более точно: билинейная/бикубическая интерполяция

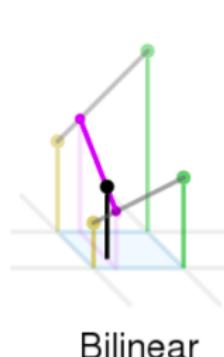
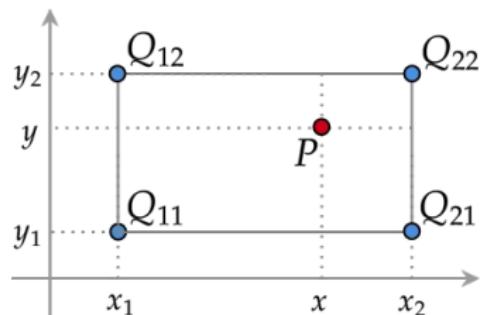
Билинейная интерполяция

$$P(y_1) = (1 - \alpha) Q_{11} + \alpha Q_{21}, \quad P(y_2) = (1 - \alpha) Q_{12} + \alpha Q_{22}$$

$$\alpha = \frac{x - x_1}{x_2 - x_1}$$

$$P = \beta P(y_1) + (1 - \beta) P(y_2)$$

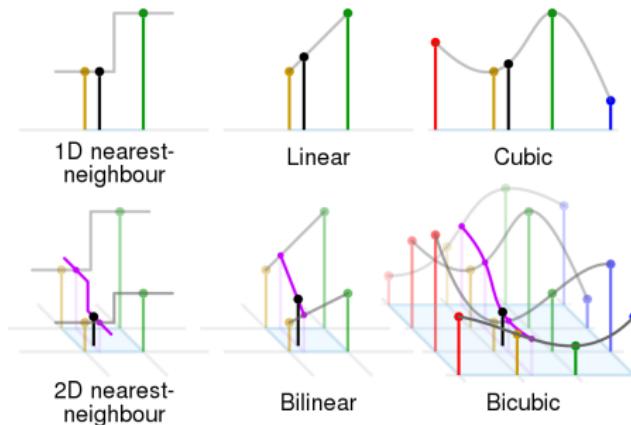
$$\beta = \frac{y - y_1}{y_2 - y_1}$$



Другие виды интерполяции

бикубическая интерполяция: $p(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j$

$p(x_i, y_j)$ -известны, производные на углах соседних квадратов равны



Транспонированная свёртка

Обычная свёртка: $h \times w \rightarrow H \times W$ и представима в виде матричного произведения:

$$\begin{pmatrix} A & B & C \\ D & E & F \\ G & H & I \end{pmatrix} \circledast \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} = \begin{pmatrix} \alpha A + \beta B + \gamma D + \delta E & \alpha B + \beta C + \gamma E + \delta F \\ \alpha D + \beta E + \gamma G + \delta H & \alpha E + \beta F + \gamma H + \delta I \end{pmatrix}$$

$$= \text{reshape} \left\{ \underbrace{\begin{pmatrix} \alpha & \beta & 0 & \delta & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha & \beta & 0 & \gamma & \delta & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha & \beta & 0 & \gamma & \delta & 0 \\ 0 & 0 & 0 & 0 & \alpha & \beta & 0 & \gamma & \delta \end{pmatrix}}_P \right\} \begin{pmatrix} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \\ I \end{pmatrix}$$

Транспонированная свёртка

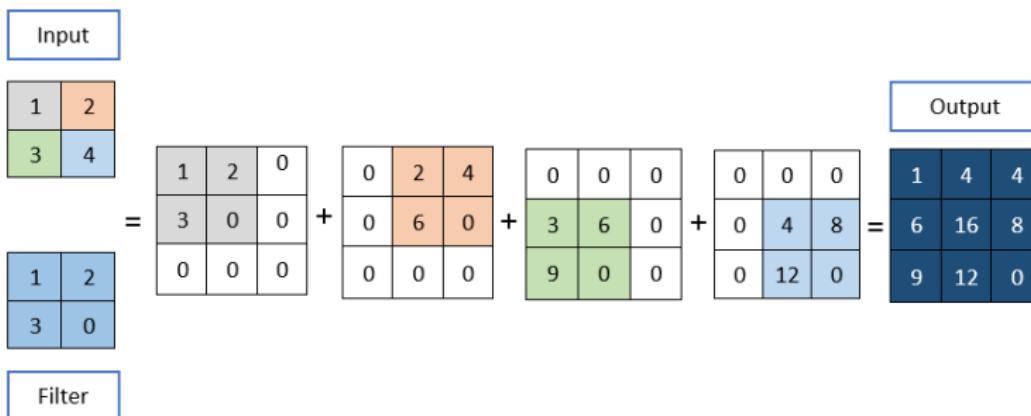
Транспонированная свёртка: $H \times W \rightarrow h \times w$,
результат-домножением на P^T :

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \circledast^T \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} = \begin{pmatrix} a\alpha & a\beta + b\alpha & b\beta \\ a\gamma + c\alpha & a\delta + b\gamma + c\beta + d\alpha & b\delta + d\beta \\ c\gamma & c\delta + d\gamma & d\delta \end{pmatrix}$$

$$= \text{reshape} \left\{ \underbrace{\begin{pmatrix} \alpha & 0 & 0 & 0 \\ \beta & \alpha & 0 & 0 \\ 0 & \beta & 0 & 0 \\ \gamma & 0 & \alpha & 0 \\ \delta & \gamma & \beta & \alpha \\ 0 & \delta & 0 & \beta \\ 0 & 0 & \gamma & 0 \\ 0 & 0 & \delta & \gamma \\ 0 & 0 & 0 & \delta \end{pmatrix}}_{P^T} \right\} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$$

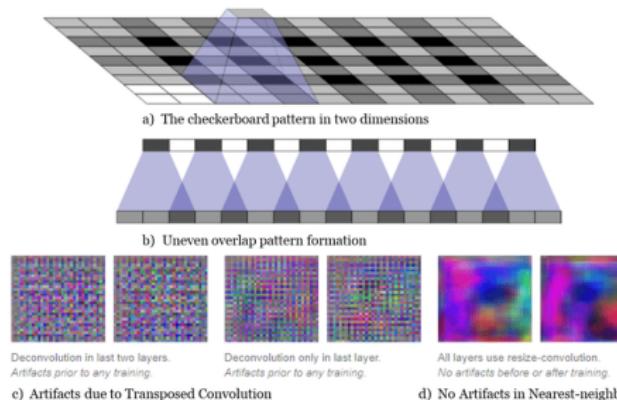
Интуиция

Транспонированная свертка суммирует "штампы" фильтра с весами входов:



Недостаток

Приводит к артефактам "шахматной доски" (checkerboard artifacts):



Есть способы борьбы с этим: $\text{stride}=\text{kernel size}$ либо применять несколько раз с перекрытием и смещением.

Содержание

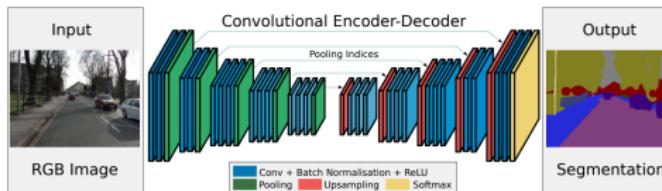
1 Введение

2 Нейросетевые архитектуры

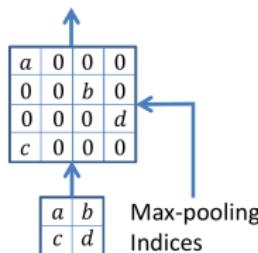
- Архитектура FCN
- Архитектура U-net
- Архитектурные расширения U-net
- Учёт глобальной информации
- Использование dilated свёрток

3 Instance и panoptic сегментация

SegNet¹¹



- Проблема: при пулинге теряем пространственную информацию.
- Решение: в декодировщике - max unpooling
 - в котором локация максимума сохраняется



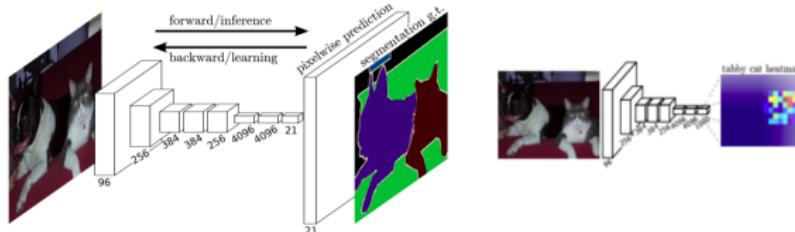
¹¹ <https://arxiv.org/pdf/1511.00561.pdf>

2 Нейросетевые архитектуры

- Архитектура FCN
- Архитектура U-net
- Архитектурные расширения U-net
- Учёт глобальной информации
- Использование dilated свёрток

Семейство архитектур FCN¹²

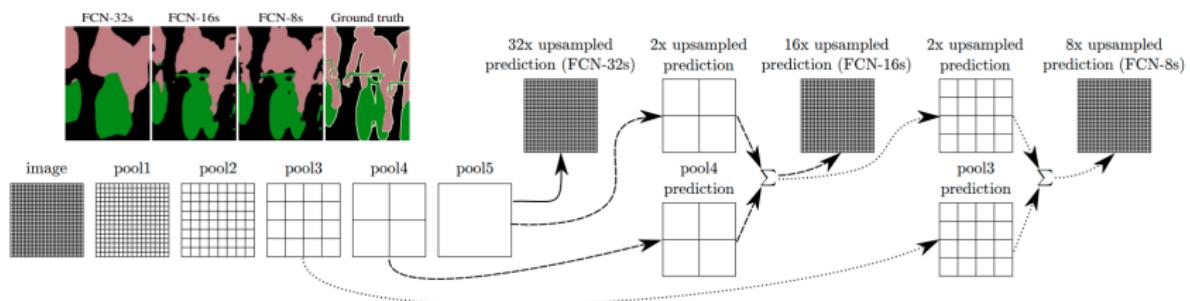
- Семейство архитектур FCN (fully convolutional networks)
- Для кодировщика использовались первые слои AlexNet, VGG, GoogleNet, затем все слои (кодировщик и декодировщик) дообучались под задачу.
- малоразмерное промежуточное представление => грубые неточные границы объектов



¹²<https://arxiv.org/pdf/1411.4038.pdf>

Архитектура FCN-8s

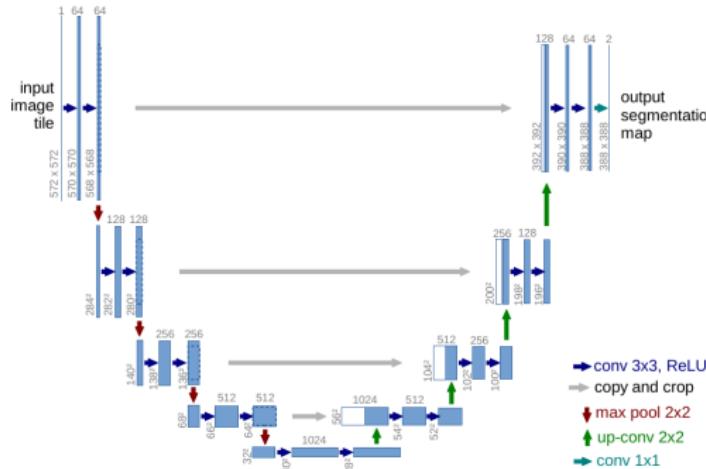
- Прибавление расширенных (upsampling) предыдущих слоев к текущему позволяет совместить:
 - низкоразмерную высокоразровневую информацию
 - высокоразмерную низкоразровневую информацию



2 Нейросетевые архитектуры

- Архитектура FCN
- Архитектура U-net
- Архитектурные расширения U-net
- Учёт глобальной информации
- Использование dilated свёрток

Архитектура U-net¹³



Горизонтальные числа = # [каналов]; вертикальные числа = пространственное разрешение.

Белые блоки - скопированный выход с пред. слоёв; up-conv - перемасштабирование & свертка.

¹³<https://arxiv.org/pdf/1505.04597.pdf>

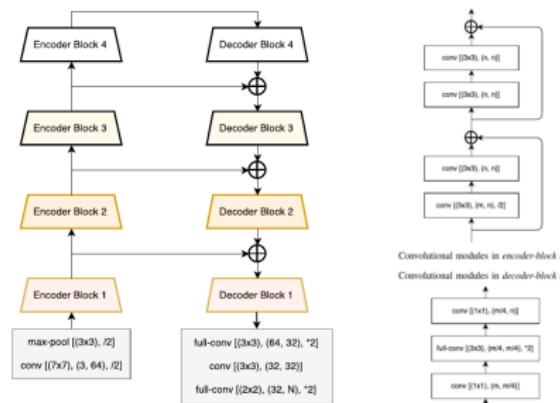
Обсуждение

Ключевые идеи U-net:

- сохраняем пространственную информацию на каждом слое
 - используется только свертка, пулинг, масштабирование.
 - не используется векторизация & полносвязные слои
- 1ая половина - кодировщик; 2ая половина - декодировщик.
- Кодировщик собирает все более широкую локальную информацию
 - и более абстрактные признаки
- Декодировщик восстанавливает локальную информацию из
 - более абстрактных признаков (зеленая стрелка)
 - более низкоуровневых признаков (серая стрелка)

LinkNet¹⁴

Как U-net, но агрегирует информацию суммированием представлений, а не конкатенацией вдоль каналов. ResNet блоки в кодировщике. В декодировщике. 1×1 свертка $\downarrow \#$ параметров и вычислений.

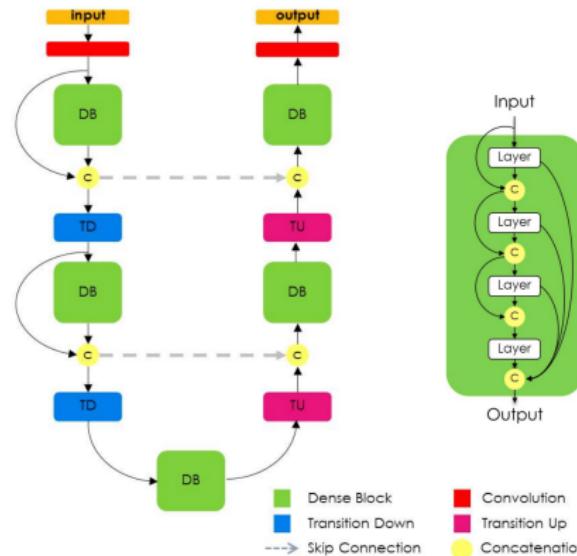


¹⁴<https://arxiv.org/pdf/1707.03718.pdf>

One Hundred Layers Tiramisu¹⁵

One Hundred Layers Tiramisu - U-net, состоящая из dense блоков:

Архитектура и dense block.

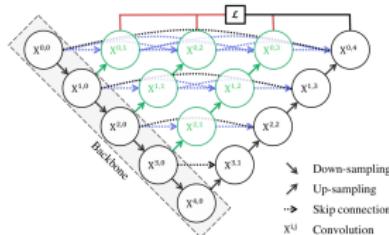


¹⁵<https://arxiv.org/pdf/1611.09326.pdf>

2 Нейросетевые архитектуры

- Архитектура FCN
- Архитектура U-net
- Архитектурные расширения U-net
- Учёт глобальной информации
- Использование dilated свёрток

U-net++¹⁶



- U-net часть выделена чёрным.
- За счёт зелёных промежуточных блоков высокоуровневые и низкоуровневые признаки приводятся в соответствие.
- Каждый ярус - dense блок (легче настройка, можно ↓ #каналов, т.к. информация не забывается)
- $X^{0,1}, X^{0,2}, X^{0,3}, X^{0,4}$ все выдают прогноз Y (ансамбль).

¹⁶<https://arxiv.org/pdf/1807.10165.pdf>

U-net++

- Итоговый прогноз:

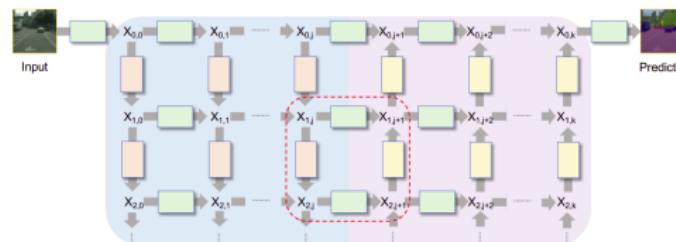
$$\hat{Y} = \frac{1}{4} (X^{0,1} + X^{0,2} + X^{0,3} + X^{0,4})$$

- Настройка: cross-entropy+dice:

$$\mathcal{L}(Y, \hat{Y}) = -\frac{1}{N} \sum_{b=1}^N \left(\frac{1}{2} \cdot Y_b \cdot \log \hat{Y}_b + \frac{2 \cdot Y_b \cdot \hat{Y}_b}{Y_b + \hat{Y}_b} \right)$$

GridNet¹⁷

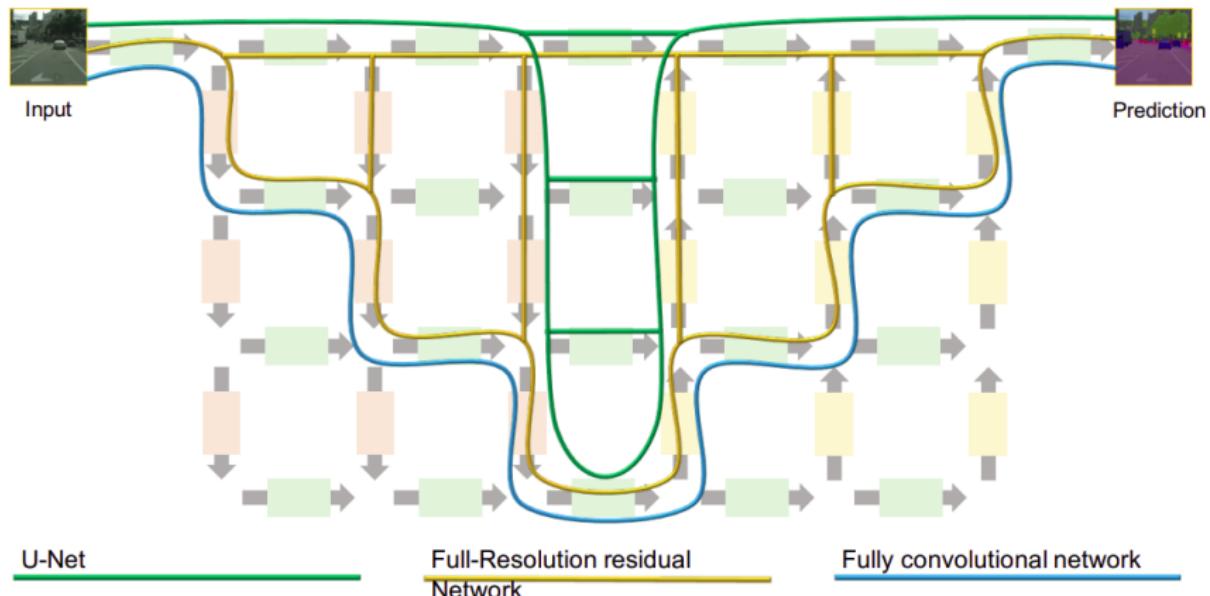
- Отличия GridNet от Unet++:
 - обучение и прогнозы - только по самому правому верхнему элементу.
 - $X_{0,0}, X_{0,1}, \dots, X_{0,k}$ - все агрегируют как низкоуровневые, так и высокоуровневые (одного порядка) признаки.
- Выходы и входы элементов агрегируются суммированием.
- Обучаем ансамбль, т.к. выход определяется информацией, полученной по разным путям от входа к выходу.



- Модель не показала улучшение качества.

¹⁷ <https://arxiv.org/pdf/1707.07958.pdf>

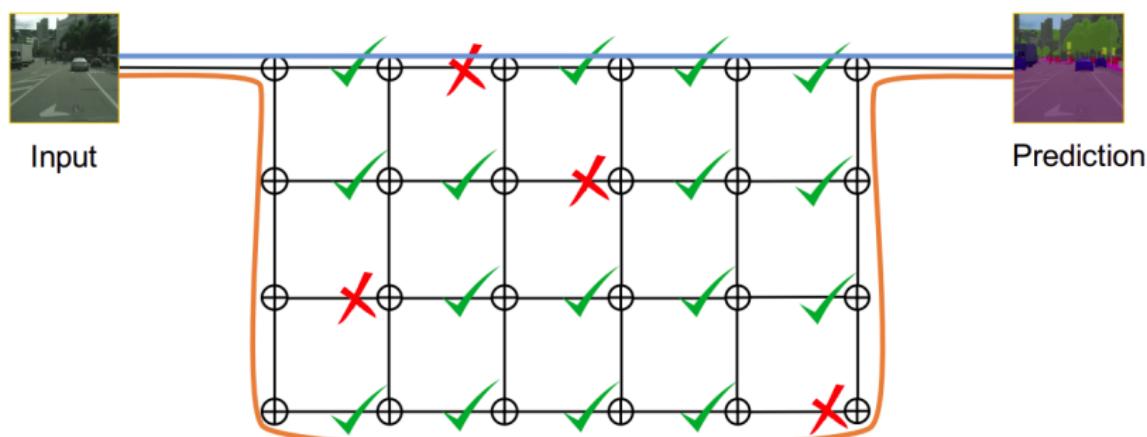
GridNet обобщает другие архитектуры



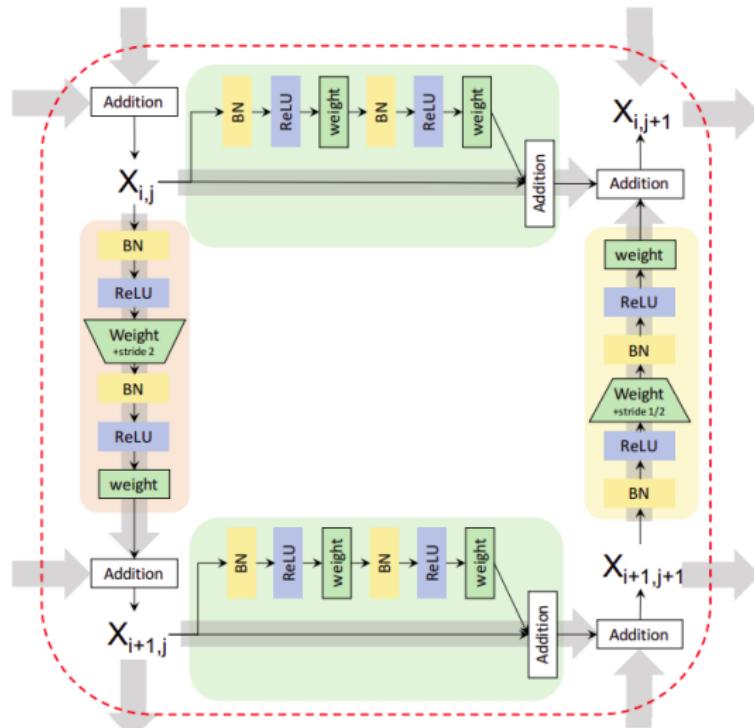
DropOut в GridNet

DropOut - отбрасывается поднабор горизонтальных связей.

- модель учится использовать информацию с разных ярусов



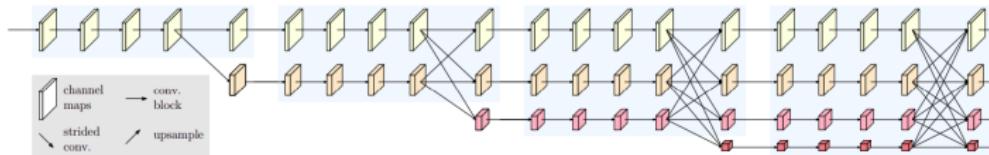
GridNet - каждый элемент



High-Resolution Representations

High-Resolution Representations for Labeling Pixels and Regions¹⁸.

- Зачем объединять только 2 представления - низкоуровневое и высокоуровневое?
- Можно параллельно поддерживать представления в нескольких разрешениях и дать сети самой выбирать, как и когда их объединять.



- Модель показала улучшение качества.

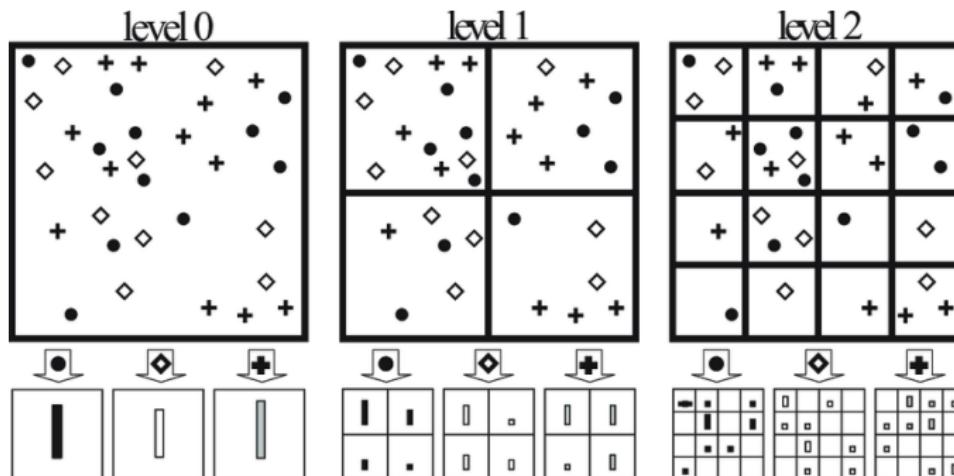
¹⁸<https://arxiv.org/pdf/1904.04514.pdf>

2 Нейросетевые архитектуры

- Архитектура FCN
- Архитектура U-net
- Архитектурные расширения U-net
- Учёт глобальной информации
- Использование dilated свёрток

PSPNet¹⁹

- По сравнению с FCN PSPNet учитывает глобальный контекст с помощью SpatialPyraidePooling.



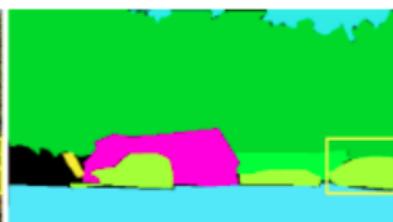
¹⁹<https://arxiv.org/pdf/1612.01105.pdf>

PSPNet

В примере FCN классифицирует лодку машиной, а PSPNet-лодкой, т.к. в окрестности воды.



(a) Image



(b) Ground Truth

sky
tree
grass
earth
plant
car
boat



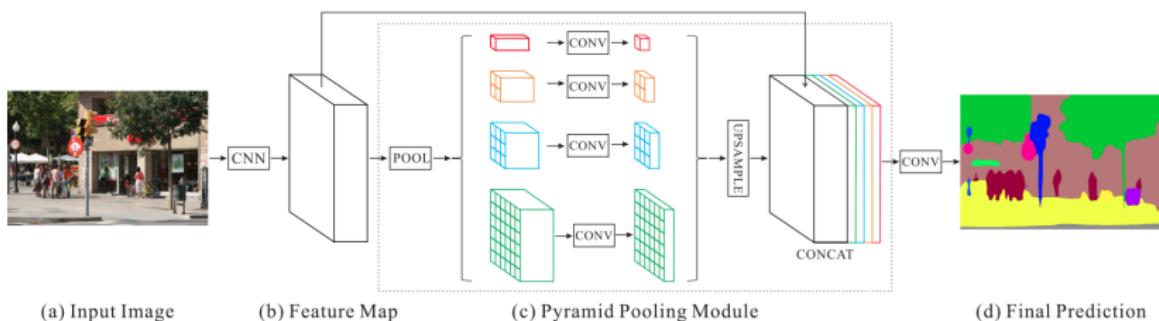
(c) FCN



(d) PSPNet

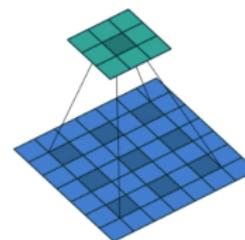
PSPNet

Кодировщик: предобученное начало ResNet. Пирамидальный пулинг для агрегации общей информации, затем расширение (upsampling) для совместимости с высокоразмерным представлением.

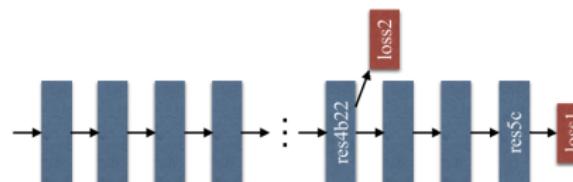


PSPNet

- CNN кодировщик - ResNet со свёртками с dilation:



- Во время обучения используется дополнительная ф-ция потерь в середине (принцип deep supervision):



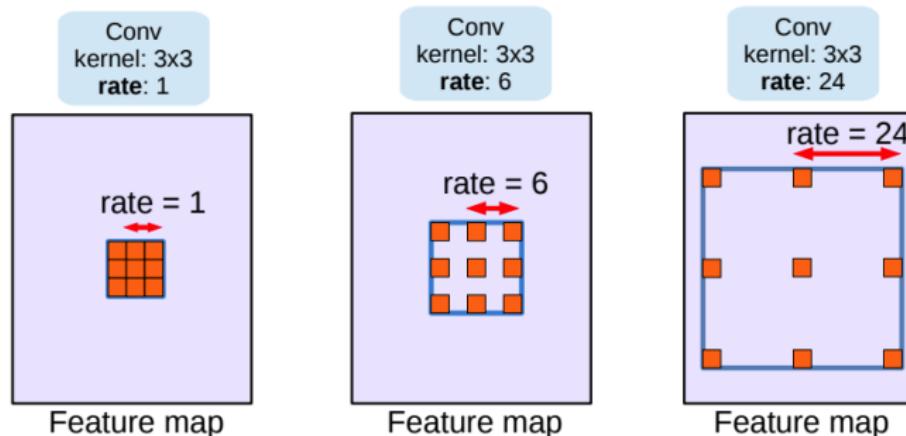
2 Нейросетевые архитектуры

- Архитектура FCN
- Архитектура U-net
- Архитектурные расширения U-net
- Учёт глобальной информации
- Использование dilated свёрток

DeepLab V3 (2017)²⁰

Используются dilated convolutions (называемые Atrous convolutions).

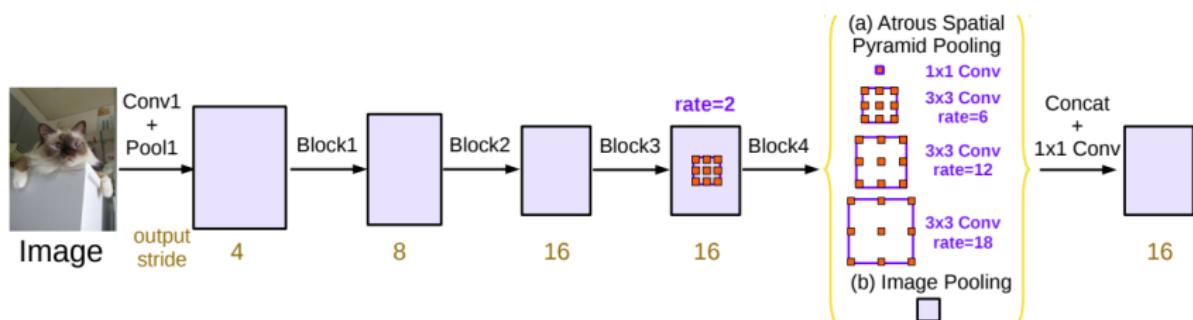
- позволяют собирать информацию по более широкой окрестности при том же #вычислений и #параметров.



²⁰<https://arxiv.org/pdf/1706.05587v3.pdf>

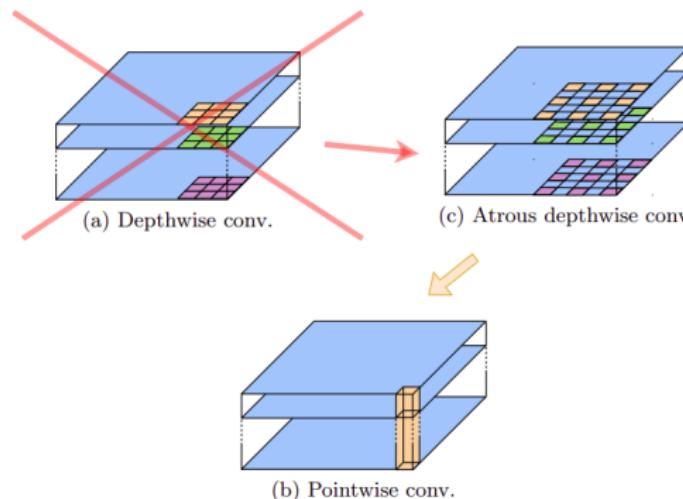
DeepLab V3: блок объединения результатов

- Конкatenируются рез-ты dilated conv с разным dilation
 - +GlobalAvgPooling->conv 1x1->растягивается по размеру feature map
- Потом применяется conv 1x1 (\downarrow размерности)



DeepLab V3+ (2018)²¹

- Основан на архитектуре Xception
- Но для ↑ области видимости использованы dilated depthwise separable свёртки.



²¹<https://arxiv.org/pdf/1802.02611v3.pdf>

DeepLab V3+: дополнительные связи

Для \uparrow точности комбинируются низкоуровневые и высокуюровневые признаки в кодировщике и декодировщике:

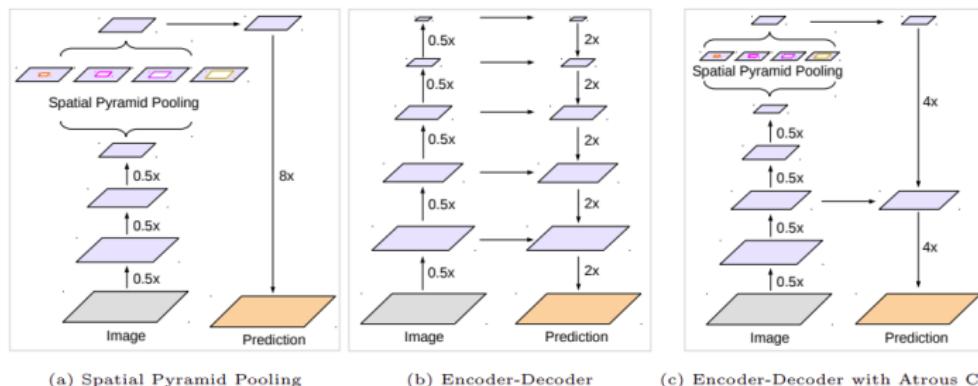
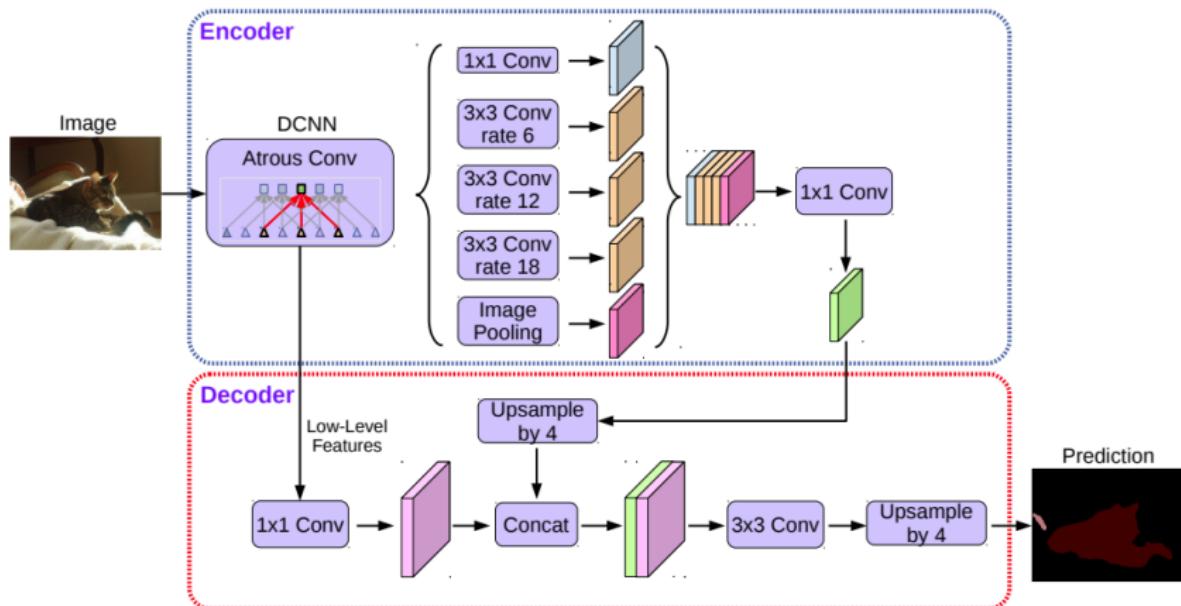


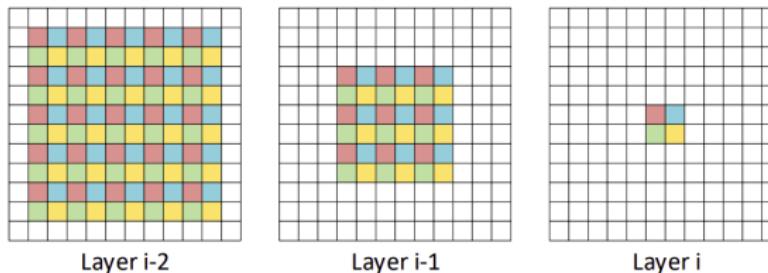
Fig. 1. We improve DeepLabv3, which employs the spatial pyramid pooling module (a), with the encoder-decoder structure (b). The proposed model, DeepLabv3+, contains rich semantic information from the encoder module, while the detailed object boundaries are recovered by the simple yet effective decoder module. The encoder module allows us to extract features at an arbitrary resolution by applying atrous convolution.

DeepLab V3+



Особенности работы с dilated свёртками

Выход dilated свёрток зависит от разных пикселей:



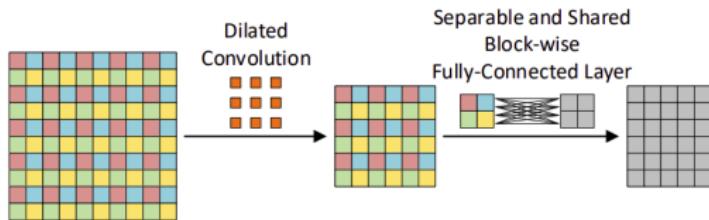
- Получаем пространственную нестабильность прогнозов, т.к. они определяются разными блоками пикселей.
- Smoothed Dilated Convolutions for Improved Dense Prediction²²: предлагается 2 подхода для сглаживания выходов.
 - подходит \uparrow точность сегментации DeepLab V2.

²²<https://arxiv.org/pdf/1808.08931.pdf>

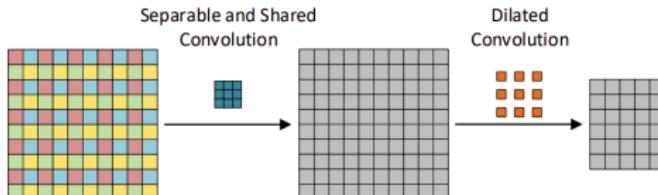
Smoothed Dilated Convolutions²³

Предлагаются 2 подхода (иллюстрация для dilation=2):

- ① патч 2x2 → патч 2x2, используя полно связный слой:
скользим им с одинаковыми весами по каждой карте
признаков в отдельности со stride=2:



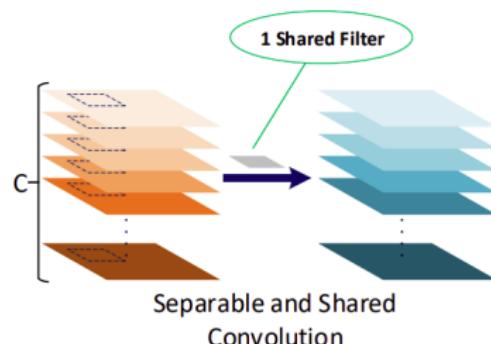
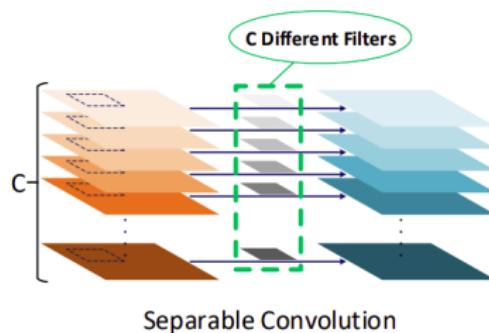
- ② либо применить separable & shared conv перед dilated conv:



²³<https://arxiv.org/pdf/1808.08931.pdf>

Smoothed Dilated Convolutions

- Т.е. 2й подход: **separable & shared conv** - свёртка, действующая с одинаковыми весами на каждый канал в отдельности:

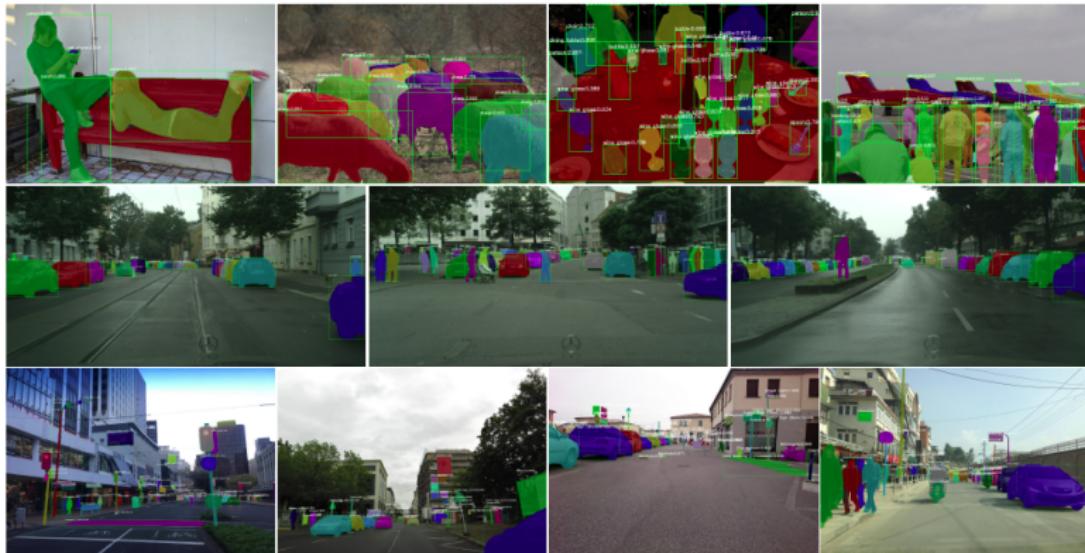


Содержание

- 1 Введение
- 2 Нейросетевые архитектуры
- 3 Instance и panoptic сегментация

Path Aggregation Network for Instance Segmentation²⁴

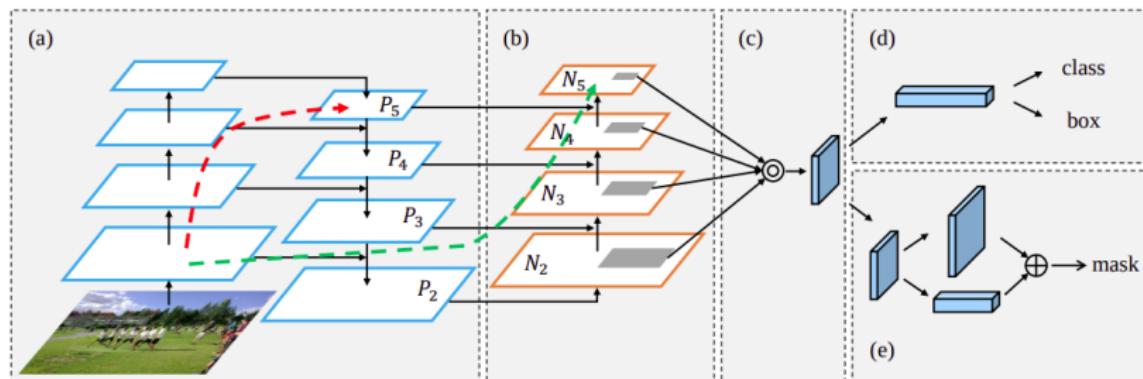
Path Aggregation Network (PANet) реализует instance-сегментацию.



²⁴<https://arxiv.org/pdf/1803.01534.pdf>

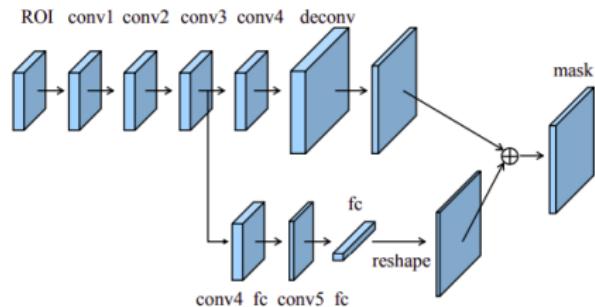
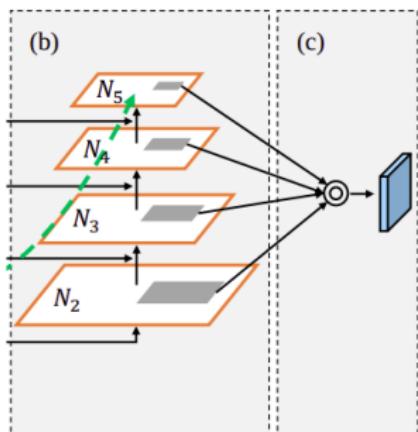
Path Aggregation Network for Instance Segmentation

Для каждого обнаруженного объекта предсказывается рамка, класс и маска выделения.



Path Aggregation Network for Instance Segmentation

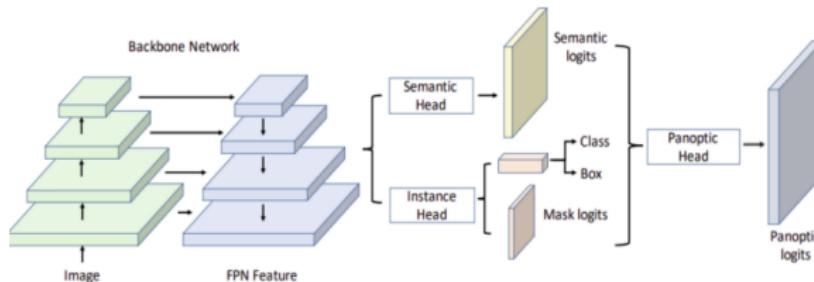
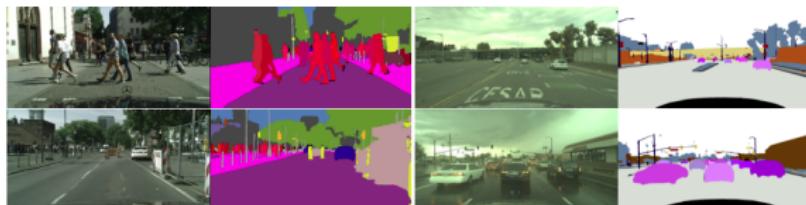
- Adaptive feature pooling использует max-pooling для каждой позиции по каналам.
- Маска выделения (mask prediction) предсказывается агрегацией выходов
 - сверточного слоя (учитываем локальную информацию)
 - полносвязного слоя (учитываем позицию)



Mask prediction branch with fully-connected fusion.

USPNet²⁵

- USPNet реализует panoptic segmentation.
- Блок Panoptic head: агрегация семантической и instance сегментации:



²⁵<https://arxiv.org/pdf/1901.03784.pdf>

Заключение

- Семантическая сегментация реализуется полносвёрточными архитектурами.
- Ключевая проблема: сохранение & учёт информации
 - с высокоуровневых признаков (семантика)
 - низкоуровневых признаков (\uparrow точности выделения границ)
- Решение проблемы:
 - суммирование или конкатенация промежуточных признаковых представлений
 - добавление канала GlobalPooling либо SpatialPyramidPooling.
- Метрики качества: accuracy, IoU, dice
 - их можно не только отслеживать, но и оптимизировать по ним
- Instance-сегментация: свой класс для каждого объекта.
- Panoptic-сегментация: Instance-сегментация + сегментация фона.