

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ М.В.ЛОМОНОСОВА

ЭССЕ ПО КУРСУ "ГЛУБОКОЕ ОБУЧЕНИЕ"

Сегментация изображений

Тыцкий В.И.
317 группа ВМК

2 июня 2021 г.

Аннотация

Сегментация изображений лежит в основе многочисленных задач компьютерного зрения. Эта задача встречается во многих сферах жизни: беспилотные автомобили, медицинская диагностика, киноиндустрия, фоторедактирование. В данном эссе дана постановка задачи сегментации, затронуты классические методы решения, а также рассмотрены современные подходы с использованием нейросетей.

Введение

Задача **семантической** сегментации – определение принадлежности каждого пикселя изображения к некоторому классу из заранее заданного множества.[3]

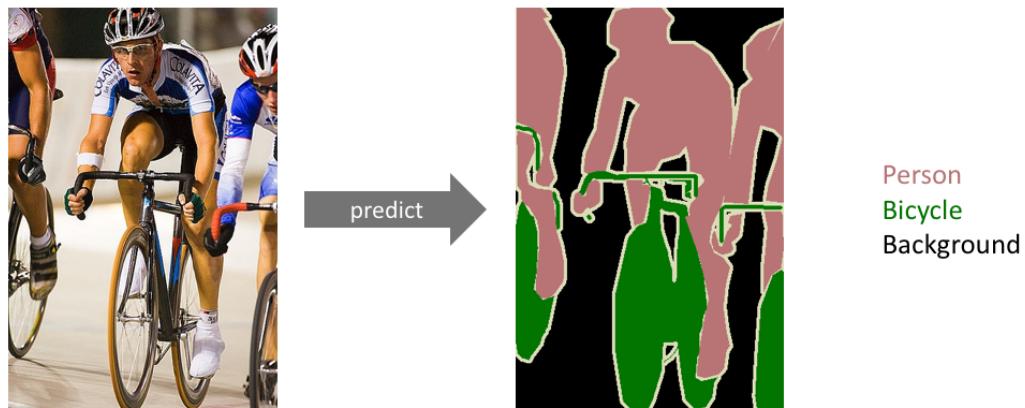


Рис. 1: Пример семантической сегментации [4]

На Рис. 1 Представлен пример желаемого результата для входного изображения.

Основные сферы применения

- **Беспилотное вождение.** Для беспилотных автомобилей (и других беспилотных видов транспорта) необходимо в реальном времени понимать происходящее вокруг, поэтому сегментация должна выполняться не только качественно, но и быстро.
- **Медицинские изображения.** Решение задачи сегментации может существенно ускорить и упростить диагностику заболеваний.
- **Извлечение отдельных объектов на изображении.** Широко применяется в киноиндустрии для создания спецэффектов. Помимо может широко применяться в профессиональной и любительской обработке фотографий.
- **Понимание изображений**

Типы сегментаций¹

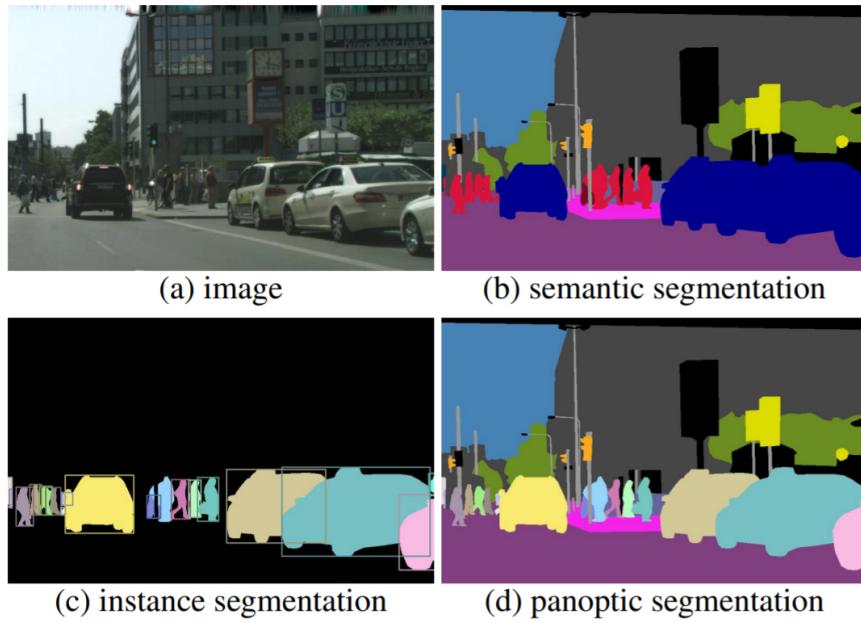


Рис. 2: Различные типы сегментации [2]

- **Semantic segmentation** – выделяет объекты в категории (человек, машина и др.). Каждый пиксель относится к некоторому классу.
- **Instance segmentation** – выделяет объекты в экземпляры категорий. Например, если на картинке два человека, то они будут определены как два разных человека. Причем не каждый пиксель может принадлежать классу.
- **Panoptic segmentation** – совмещает в себе instance и semantic сегментацию, то есть выделяются как категории объектов, так и экземпляры категории. [2]

Данные

Для решения задачи сегментации необходимы примеры для обучения каких-либо моделей. Основные датасеты (datasets):

- Common Objects in COntext—Coco Dataset (COCO)
- PASCAL Visual Object Classes (PASCAL VOC)
- The Cityscapes Dataset
- The Cambridge-driving Labeled Video Database—CamVid

¹У Дьяконова А.Г. [1] немного другие определения. Возможно что-то неправильно мной понято.

Метрики

Чтобы ввести, с помощью которых будет измеряться качество/ошибка необходимо формализовать определение сегментации:

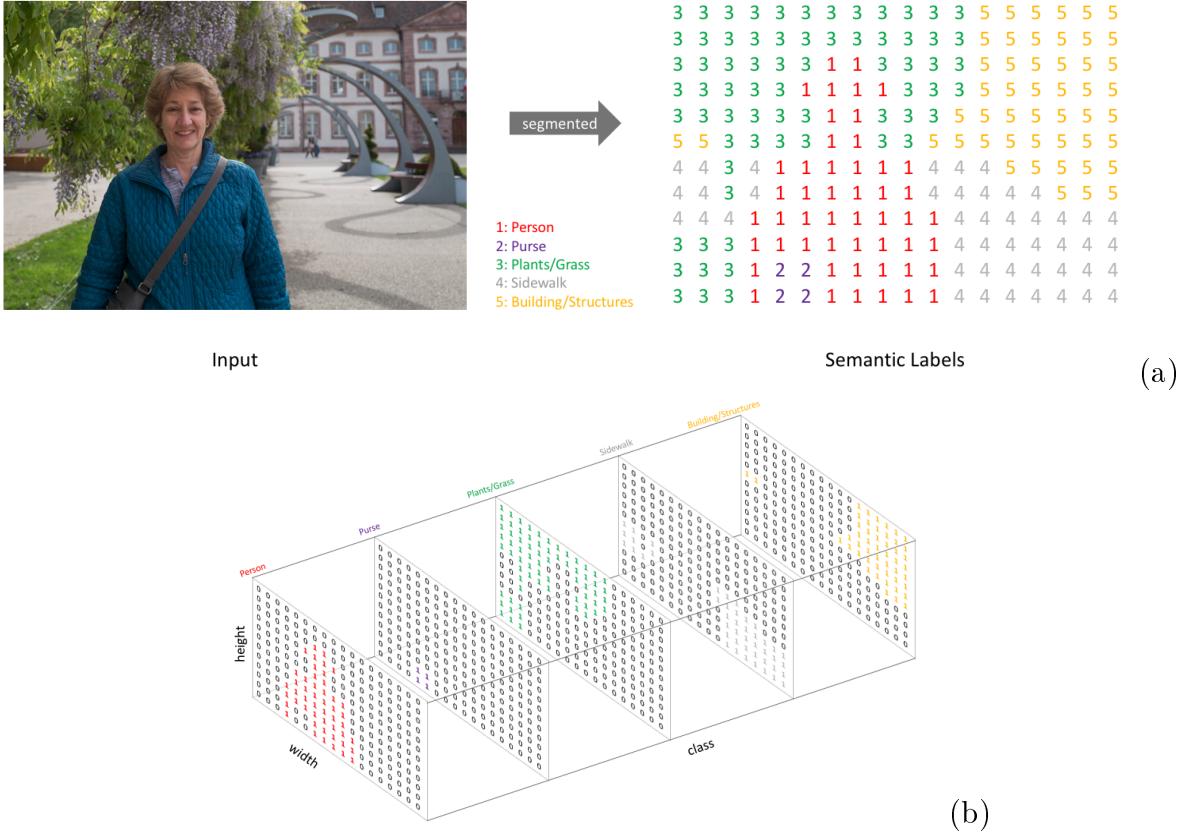


Рис. 3: Представление семантической сегментации [3]

- (a) Первое определение (F_1) семантической сегментации.
- (b) Второе определение (F_2) семантической сегментации.

Пусть W - ширина входного изображения, H - высота, C - количество каналов (в RGB равно 3), $K = \{0, \dots, n-1\}$ - метки классов. Изображение можно представить как тензор $M^{H \times W \times C}$, где $M \in \mathbb{R}$. Обычно $M = [0, 1]$ или $M = \{0, \dots, 255\}$.

Семантическую сегментацию можно определить двумя способами:

1. $F_1 : M^{H \times W \times C} \rightarrow K^{H \times W}$
2. $F_2 : M^{H \times W \times C} \rightarrow \{0, 1\}^{H \times W \times |K|}$

Во втором случае только один элемент равен 1 вдоль третьей оси. Нетрудно показать, что $F_1(X)$ это argmax по глубине от тензора $F_2(X)$. В дальнейшем будет использоваться именно F_2 в качестве определения, если не сказано иное.

CrossEntropy

Для одного пикселя изображения кросс-энтропия определяется как:

$$-\mathbb{E}_q \log p = - \sum_x q(x) \log p(x)$$

где $q(x)$ – истинное распределение классов (обычно вырожденное), $p(x)$ – предсказанное распределение.

Значение функции ошибки для всего всего изображения можно посчитать, найдя среднее по всем пикселям.

IoU

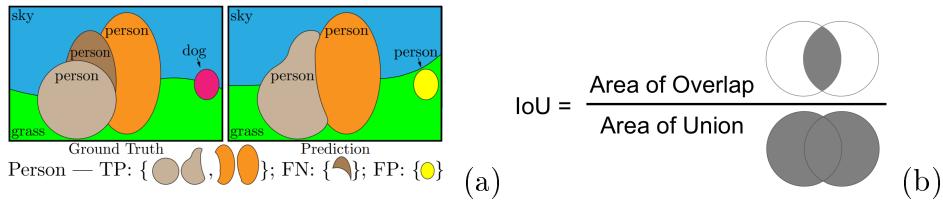


Рис. 4: (a) Элементы матрицы ошибок в контексте сегментации[2]
 (b) Визуальное представление формулы IoU [10]

Для маски соответствующей классу, IoU вычисляется по формуле:

$$IoU = \frac{|Y_{pred} \cap Y_{true}|}{|Y_{pred} \cup Y_{true}|} = \frac{TP}{TP + FN + FP}$$

где Y_{true} истинная маска, Y_{pred} предсказанная бинаризованная маска.

В задачах многоклассовой сегментации, достаточно посчитать среднее по всем маскам, соответствующим классам – такая метрика называется mIoU.

DiceLoss

Как и в случае IoU сначала определим DiceLoss для маски:

$$DiceLoss = 1 - 2 \frac{\sum_{pixels} y_{true} y_{pred}}{\sum_{pixels} y_{true}^2 + \sum_{pixels} y_{pred}^2}$$

где y_{pred} – предсказанная вероятность принадлежности пикселя к классу.

В задачах многоклассовой сегментации достаточно посчитать среднее по всем маскам, соответствующим классам.

Классические подходы

Сегментация кластеризацией



Рис. 5: Сегментация кластеризацией [20]

Идея кластеризации базируется на допущении, что объекты из некоторого класса "непрерывны". Тогда мы можем провести кластеризацию в 5D пространстве $XYRGB$ любым доступным методом кластеризации и получить сегментационные маски.

Условные случайные поля

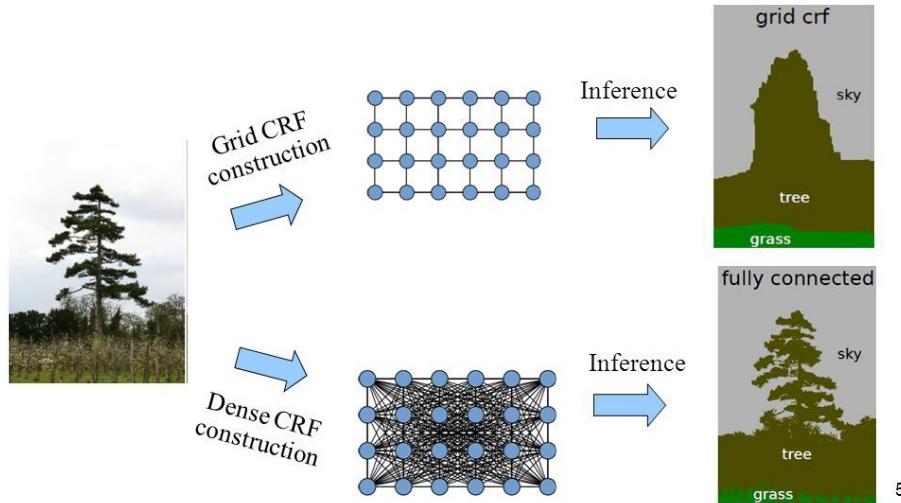


Рис. 6: Сегментация CRF [6]

Если мы считаем, что объекты непрерывны, то соседние пиксели скорее всего имеют одинаковые метки. Исходя из этого предположения, изображение можно представить в виде графа, причем соседи у вершин могут определяться произвольно. В методе условных случайных полей вводится некоторая функция, называемая энергией. Она состоит из двух слагаемых: первое отвечает за цену, которую мы "платим"

за отнесение пикселя к определенному классу, а второе цена за то, что соседние пиксели разного класса. Оптимизируя энергию, мы получаем итоговую сегментацию. На рисунке 7 показано как меняется качество сегментации в зависимости от способа задания соседей вершин. Метод с локальным соседством работает быстро, но не очень качественно. Метод с полносвязным соседством работает медленно, но качественно.

CRF+DL

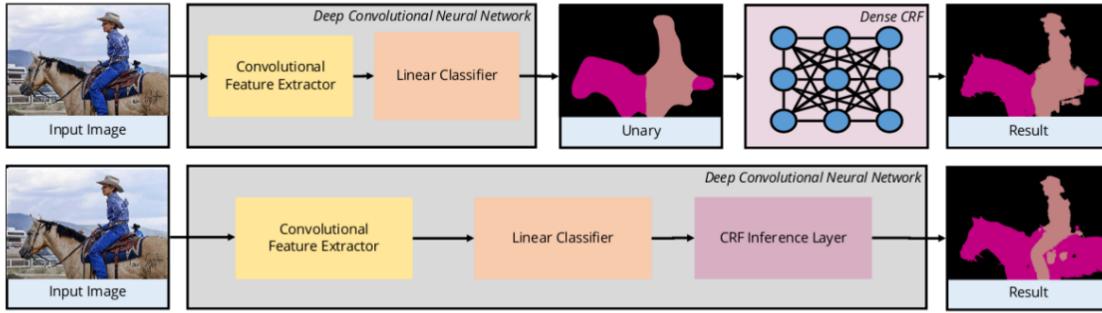
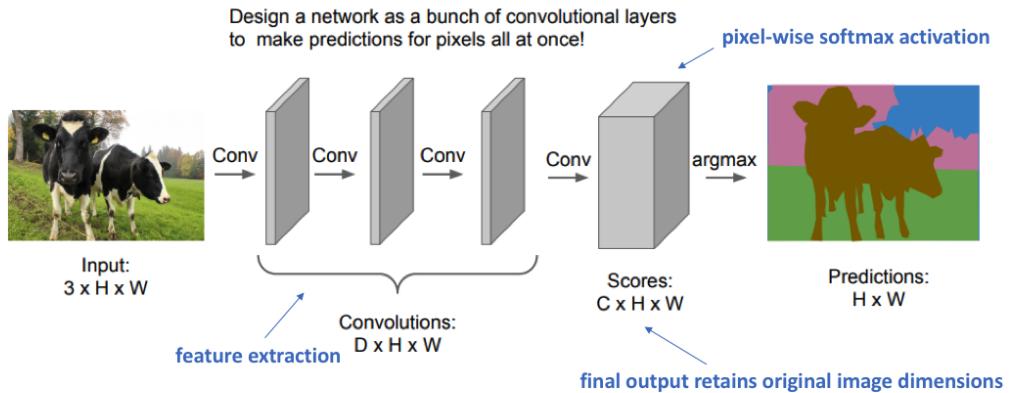


Рис. 7: Сегментация CRF+DL [6]

Некоторые методы используют сверточную нейросеть, чтобы извлекать признаки, которые потом используются как входы плотных условных случайных полей. Такой гибридный метод показывает хорошие результаты из-за способности CRF моделировать связи между пикселями.

Полностью сверточные сети



Downside: Preserving image dimensions throughout entire network will be computationally expensive.

Рис. 8: Базовая архитектура FCN [3]

Т.к. сегментация суть есть отображение тензора в тензор, то можно воспользоваться полностью сверточными архитектурами (Рис. 8). Основная проблема таких

сетей высокая размерность промежуточных слоев, что приводит к большому числу параметров и как следствие низкой скорости обучения. Необходимо понизить размерность слоев, а потом "развернуть" на выходе. (Рис. 12)

Upsampling и Transposed convolution

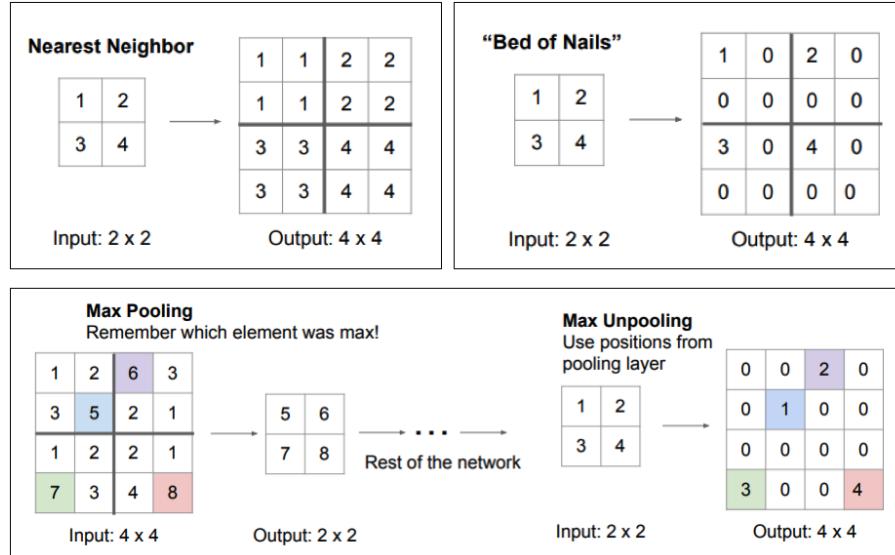


Рис. 9: Методы upsampling FCN [21]

Upsampling

Для увеличения разрешения изображения существуют классические подходы (Рис. 9):

- **Nearest Neighbor** – дублируем каждый пиксель по высоте и ширине.
- **Bed of Nails** – значение пикселя в левом верхнем соответствующего квадрата берется из исходного изображения, все остальные пиксели заполняются нулями.
- **Max unpooling** – в некотором смысле процедура обратная maxpooling. Если был сделан maxpooling, мы можем заполнить места максимальных элементов и выставить значения соответствующих пикселей на этих местах.²

Transposed convolution

Т.к. параметризованная есть операция понижения размерности(convolution), то можно попытаться придумать операцию "обратную" данной.

²На самом деле картинки дают более чем исчерпывающее объяснение механизма работы данных методов.

Напомним определение свертки:³

$$\begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix} * \begin{pmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{pmatrix} = H \begin{pmatrix} x_{11} \\ x_{21} \\ x_{31} \\ x_{21} \\ x_{22} \\ x_{23} \\ x_{31} \\ x_{32} \\ x_{33} \end{pmatrix}$$

$$ede H = \begin{pmatrix} k_{11} & k_{12} & 0 & k_{21} & k_{22} & 0 & 0 & 0 & 0 \\ 0 & k_{11} & k_{12} & 0 & k_{21} & k_{22} & 0 & 0 & 0 \\ 0 & 0 & 0 & k_{11} & k_{12} & 0 & k_{21} & k_{22} & 0 \\ 0 & 0 & 0 & 0 & k_{11} & k_{12} & 0 & k_{21} & k_{22} \end{pmatrix}$$

Можно транспонировать матрицу H и тогда при умножении ее на вектор, мы будем получим вектор исходной размерности.

Обратная свертка определяется так:⁴

$$\begin{pmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{pmatrix} *^T \begin{pmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{pmatrix} = H^T \cdot \begin{pmatrix} z_{11} \\ z_{21} \\ z_{12} \\ z_{22} \end{pmatrix} =$$

$$= (k_{11}z_{11}, \quad k_{12}z_{11} + k_{11}z_{12}, \quad k_{12}z_{12}, \\ k_{21}z_{11} + k_{11}z_{21}, \quad k_{22}z_{11} + k_{21}z_{12} + k_{12}z_{21} + k_{11}z_{22}, \quad k_{22}z_{12} + k_{12}z_{22}, \\ k_{21}z_{21}, \quad k_{22}z_{21} + k_{21}z_{22}, \quad k_{22}z_{22})$$

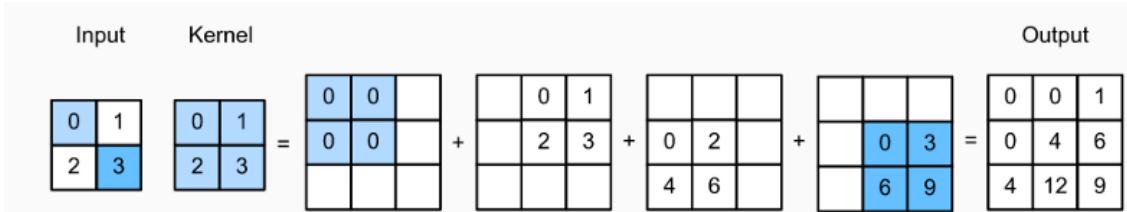


Рис. 10: Пример транспонированной свертки [21]

Сразу можно заметить проблему обратной свертки – центральные значения являются суммами с большим числом слагаемых. На практике это проявляется как “эффект шахматной” доски (Рис. 11 а).

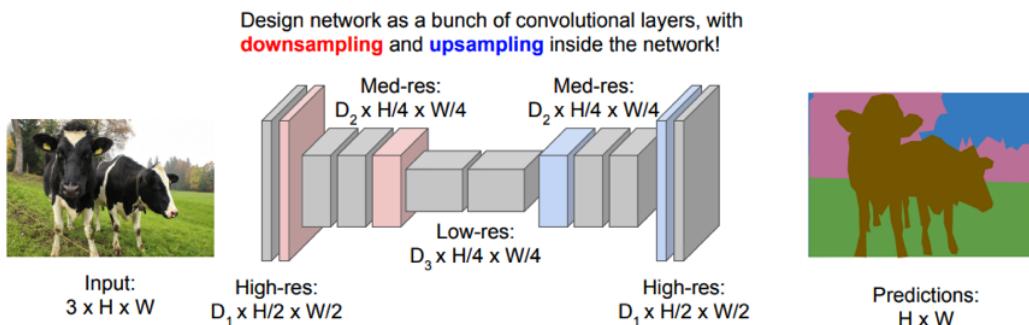
³Формулы взяты из работы предыдущего года

⁴формула взята из [1]



Рис. 11: (а) Шахматный эффект транспонированной свертки.
 (б) Сравнение работы сети с разным набором upsampling методов [9]

Один из подходов к частичному решению данной проблемы – выбрать размер ядра свертки кратный stride, избегая большого ”перекрытия”. Второй способ – это использование других методов увеличения разрешения, которые описаны выше.[9]. (см. Рис. 11 b)



Solution: Make network deep and work at a lower spatial resolution for many of the layers.

Рис. 12: Улучшенная архитектура FCN[3]

Теперь мы можем получить архитектуру сети, подобную архитектуре на рисунке 12.

Первая FCN

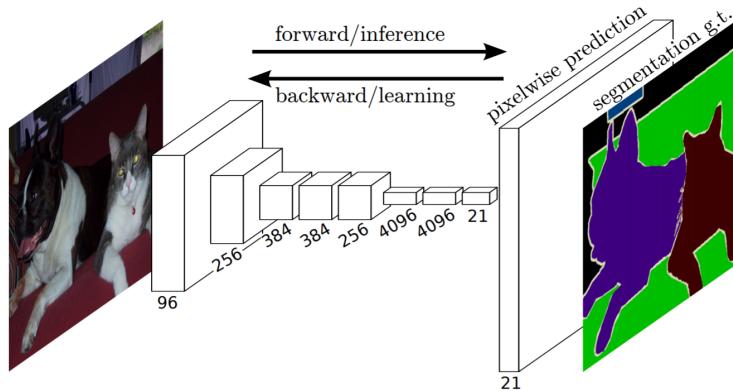


Рис. 13: Архитектура первой FCN [7]

Первой удачная полносверточная архитектура представлена на рисунке 13 [7]. Авторы предложили способ "перехода" от задачи классификации к задаче сегментации – заменяем полно связные слои на сверточные. Глядя на сеть можно заметить проблему "забывания".

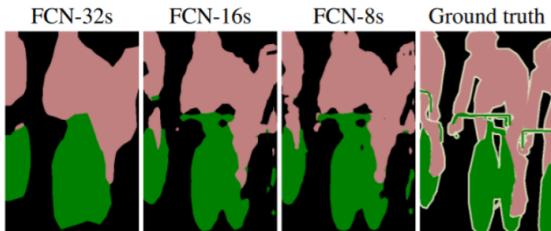


Figure 4. Refining fully convolutional nets by fusing information from layers with different strides improves segmentation detail. The first three images show the output from our 32, 16, and 8 pixel stride nets (see Figure 3).

Table 2. Comparison of skip FCNs on a subset of PASCAL VOC2011 validation⁷. Learning is end-to-end, except for FCN-32s-fixed, where only the last layer is fine-tuned. Note that FCN-32s is FCN-VGG16, renamed to highlight stride.

	pixel acc.	mean acc.	mean IU	f.w. IU
FCN-32s-fixed	83.0	59.7	45.4	72.0
FCN-32s	89.1	73.3	59.4	81.4
FCN-16s	90.0	75.7	62.4	83.0
FCN-8s	90.3	75.9	62.7	83.2

Рис. 14: Сравнение разных реализаций skip-connections[7]

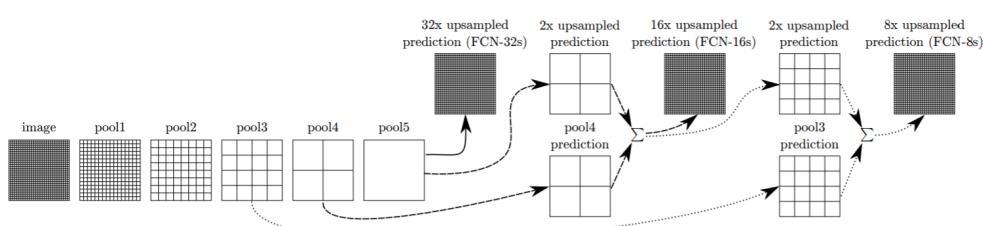


Рис. 15: Skip-connections в статье [7]

При простом восстановлении разрешения из низкоразмерного тензора мы понимаем, "что" изображено, но забываем "где". Авторы предлагают решать эту проблему с помощью **прокидывания связей** (skip-connections). Можно тензоры большего разрешения "прокидывать" к последнему upsampling слою (это может быть как и операция сложения, умножения, так и конкатенация двух тензоров)15. Далее на примере

сети Unet мы увидим развитие идеи skip-connections. На рис. 14 можно заметить как сильно возрастает качество при сборе информации с больших масштабов.

UNet

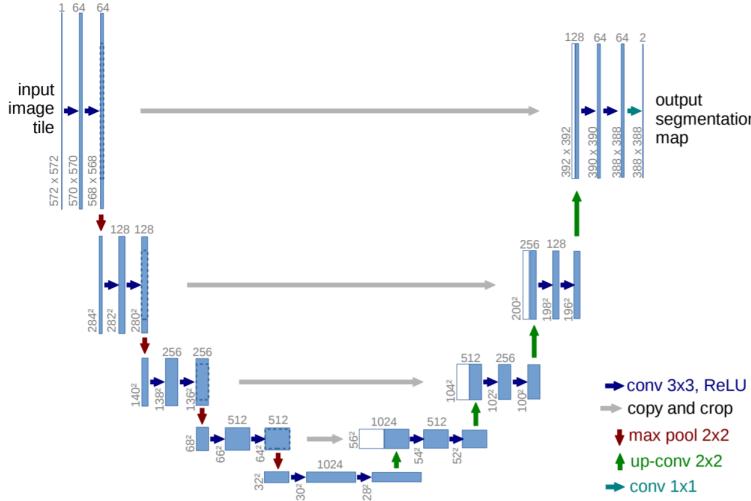


Рис. 16: Архитектура Unet [8]

Одной из самых популярных сетей для сегментации изображений по праву можно назвать Unet [8]. Основная идея состоит в архитектуре encoder-decoder, в которой соответствующие тензоры из encode блока "прокидываются" к тензорам из decode блока, в случае Unet выбрали авторы "прокидывание" через конкатенацию. Таким образом сеть "собирает" информацию с разных масштабов, для корректировки сегментации. Часто в архитектуру добавляют BatchNorm перед нелинейностями, но в Unet обошлись без них.

Аугментация

Для обучения Unet на медицинских изображениях авторы использовали большое количество методов аугментации. Среди них:

- Сдвиг
- Изменение яркости
- Упругие деформации

TernausNet

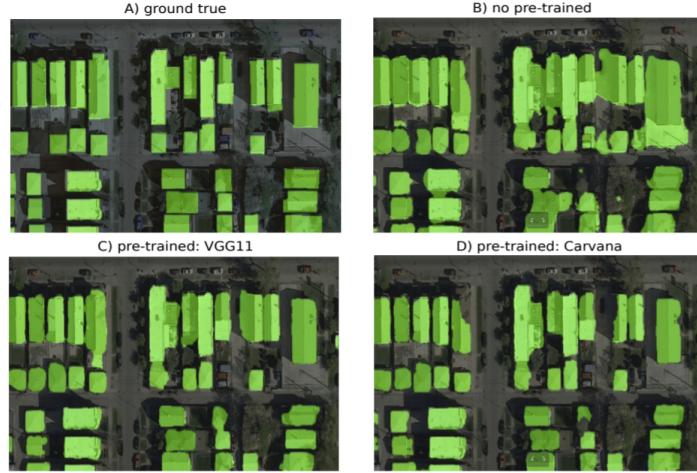


Рис. 17: Пример работы TernausNet [22]

Сеть TernausNet наследует архитектуру Unet, но в веса энкодера были взяты из сети VGG. Данный подход позволил автору занять первое место на [Kaggle соревновании](#) по сегментации автомобилей.

DenseNet+Unet

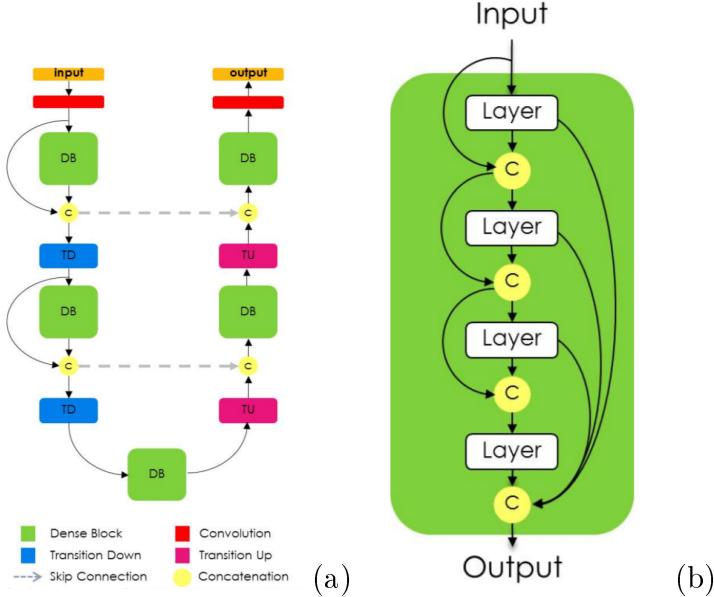


Рис. 18: (a) Архитектура "Тирамису". [12]
 (b) Вид Dense блока.

Основной идеей данной сети является добавление так называемых Dense блоков в архитектуру Unet (за это ее называют "Тирамису"), в которых все "карты признаков" от сверток соединены между собой прокидыванием связей (Рис. 18 (b)).

Layer	TransitionDown	TransitionUp
BN	BN	
ReLU	ReLU	
3X3 Conv	1X1 Conv	
Dropout p=0.2	Dropout p=0.2	
	2X2 MaxPooling	3x3 Up-Conv

Таблица 1: Структура блоков в "Тирамису"

В Таблице 1 приведены составные блоки сети. Интересно, что авторы использовали как батч-нормализацию, так и Dropout в блоке TransitionDown.

GridNet

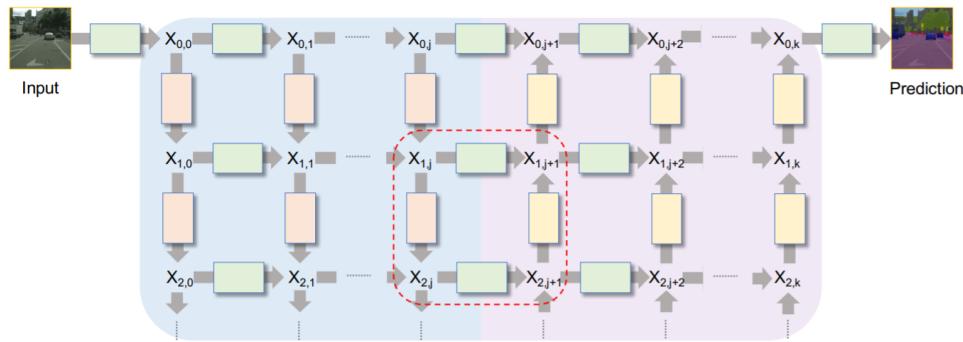


Рис. 19: (Архитектура GridNet. [17]

GridNet – это обобщение идей использования слоев, уменьшающих размер, апсемплинга и прокидывания связей. Вместо выбора конкретной архитектуры пусть сеть сама "выберет" по какому пути идти. В такой архитектуре помимо пути равноСильного базовой полносверточной архитектуры есть также путь, соответствующие Unet и Full-Resolution residual net (Рис. 20).

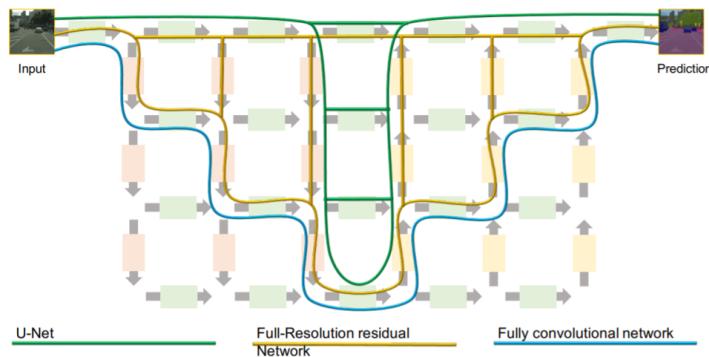


Рис. 20: Пути в GridNet и их интерпретация. [17]

На рисунке 19 синие блоки играют роль skip-connections, красные уменьшают разрешение вход сверточными слоями, а желтые апсемплят входы. На рисунке 21 изображено внутреннее устройство перечисленных блоков.

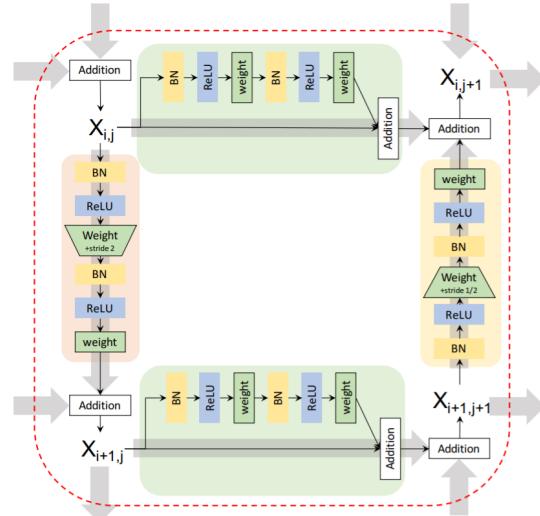


Рис. 21: Блок GridNet. [17]

При такой архитектуре можно по аналогии с Dropout "выключать" некоторые блоки, чтобы контролировать переобучение.

Dilated Convolution

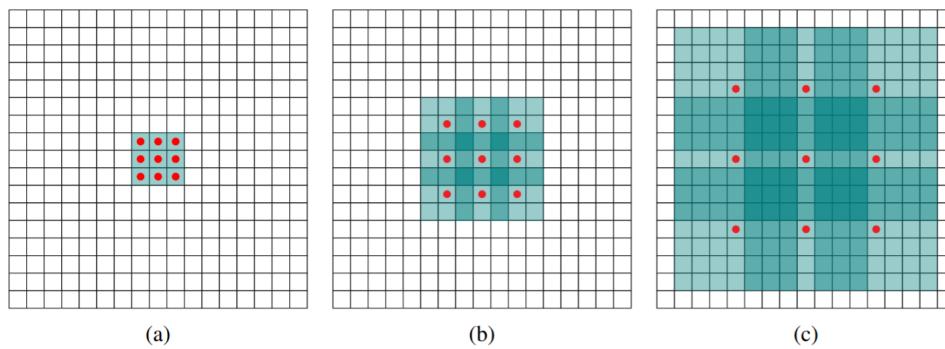


Рис. 22: (a) $d = 1$, (b) $d = 2$, (c) $d = 4$ [18]

Недостаток обычной свертки состоит в том, что она "собирает" информацию с небольшого участка изображения. Увеличение kernel-size не всегда хороший способ борьбы с данной проблемой т.к. быстро растет количество обучаемых параметров. В некоторых архитектурах предлагается делать последовательные свертки без нелинейности – так мы увеличиваем размер участка, имея меньше оптимизируемых параметров. В задаче сегментации активно используется другой способ "сбора" параметров с больших участков – расширенная (dilated) свертка (Рис .22)

Напомним определение свертки:

$$(F * k)(p) = \sum_{s+t=p} F(s)k(t)$$

Определение расширенной свертки:

$$(F *_l k)(p) = \sum_{s+lt=p} F(s)k(t)$$

Таким образом мы каждый пиксель зависит от более "далких" регионов, что в случае сегментации необходимо для понимания принадлежности к классу.

Gridding artifact

Использование расширенной свертки влечет за собой проблему сеточных артефактов. Например желтый пиксель на i -ом слое (Рис. 23) зависит от желтых пикселей на $i-1$ "шахматно" с регулярной сеткой.

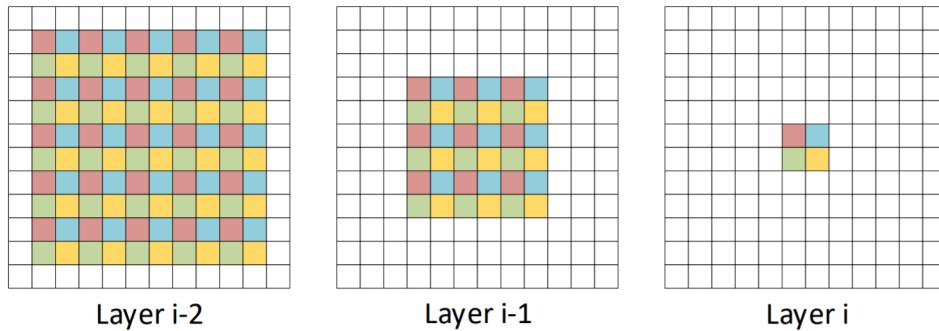


Рис. 23: Проблема сеточных артефактов. i -й слой зависит от предыдущего [19]

Потенциально это может доставить много проблем, поэтому с таким поведением борются с дополнительным полносвязным слоем после i -ого слоя. (Рис. 24)

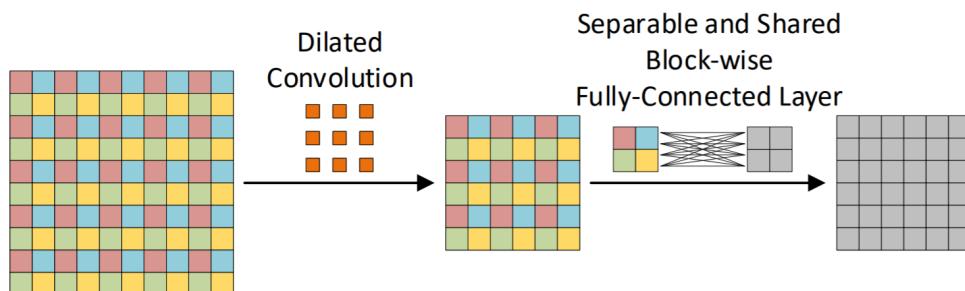


Рис. 24: Метод борьбы с сеточными артефактами при помощи полносвязных слоев[19]

Также перед расширенной сверткой можно сделать обычную и тогда шахматная зависимость пропадет. Каждый пиксель будет зависеть от каждого. (Рис. 25)

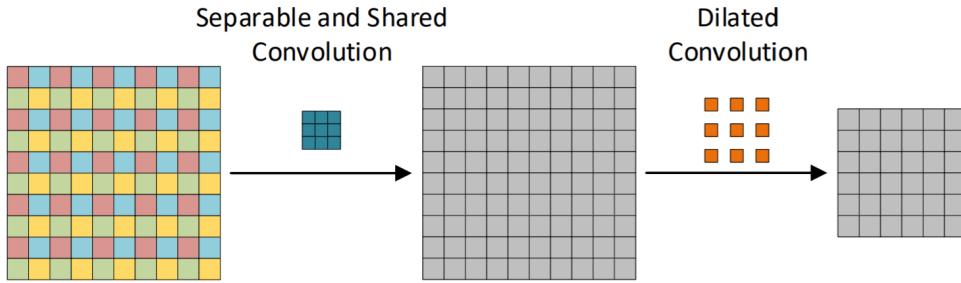


Рис. 25: Метод борьбы с сеточными артефактами при помощи обычной свертки[19]

DeepLabv

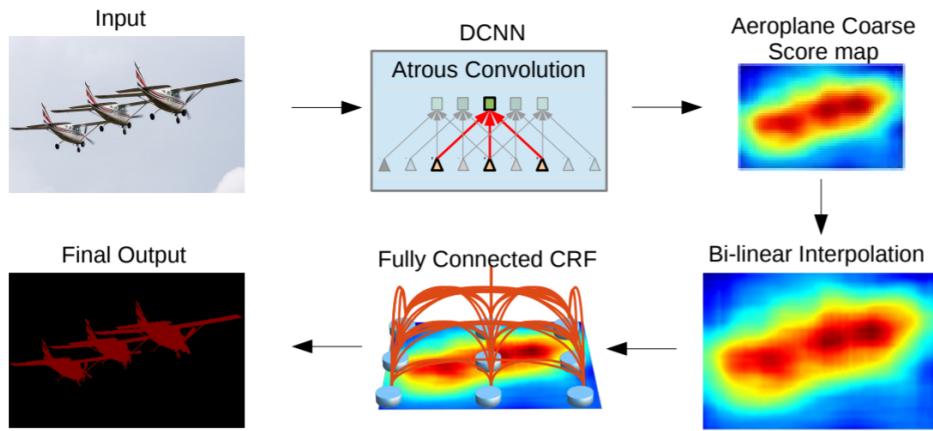


Рис. 26: DeepLabv1/2 использовал постпроцессинг CRF [14]

Одной из первых удачных сетей с использованием расширенной свертки была сеть DeepLabv1/2. Это не полносверточная архитектура т.к. используется случайные калибровочные поля в качестве постпроцессинга. (Рис. 26)

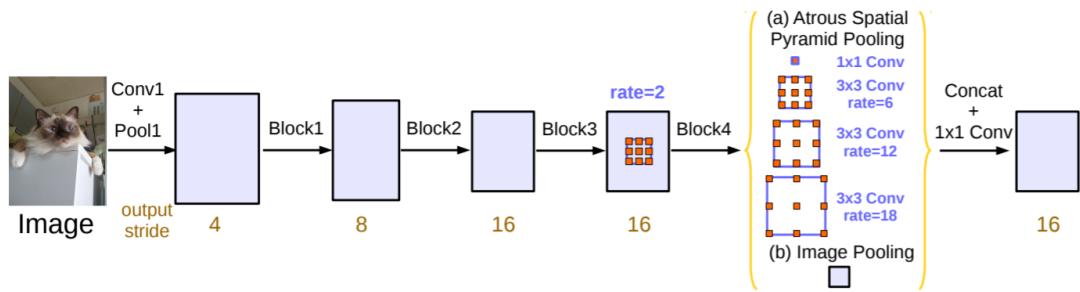


Рис. 27: DeepLabv3 и ASPP [15]

Хорошой идеей в последующих версиях DeepLabv было использование ASPP (Atrous Spatial Pyramid Pooling) – это блок, который применяет свертки с разными dilation и конкатенирует их(Рис. 27). Так мы можем получать информацию с разных

масштабов. В DeepLabv3+ к ASPP добавили skip-connections от одного из блоков энкодера.

High-resolution representations

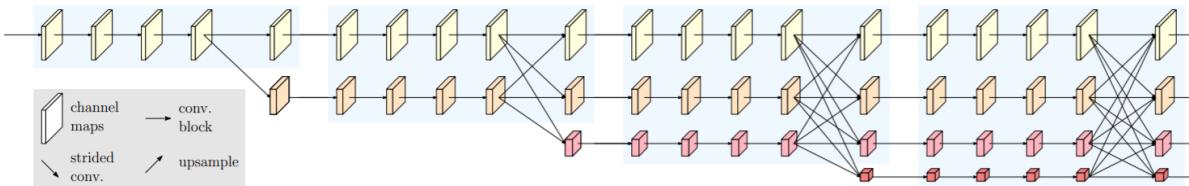


Рис. 28: Архитектура HRR [16]

High resolution representations использует идею использования информации на разных разрешениях (Рис. 28). сеть можно "поделить" на четыре вертикальных слоя – первый состоит из сверток больших разрешений, второй меньше и так далее. Каждый из этих слоев в определенных местах соединяется двумя "ребрами" с другими слоями.

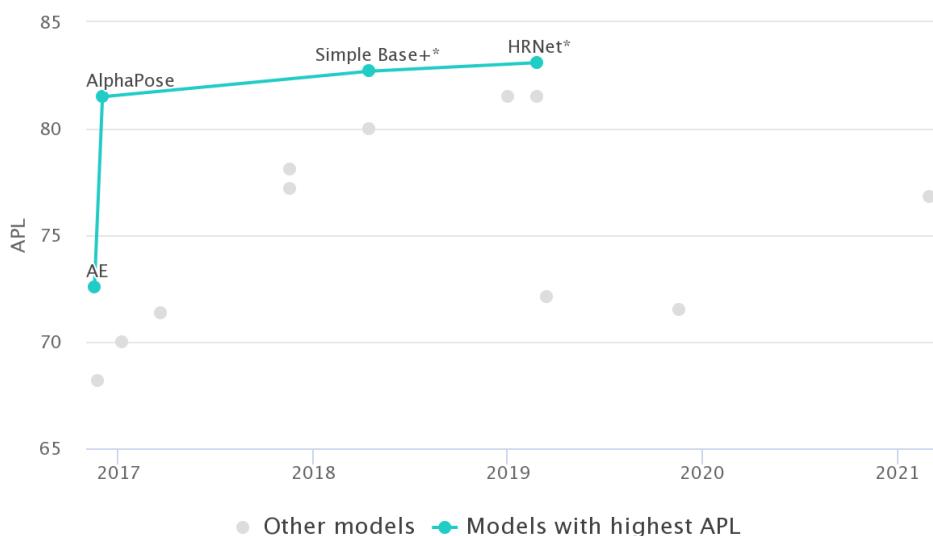


Рис. 29: SOTA в задаче оценки позы **scource**

Одна из особенностей HRR – использование ее для разных целей: сегментация, оценка позы, детекция. С 2019-го года она является SOTA в задаче оценки позы (Рис. 29)

SpineNet

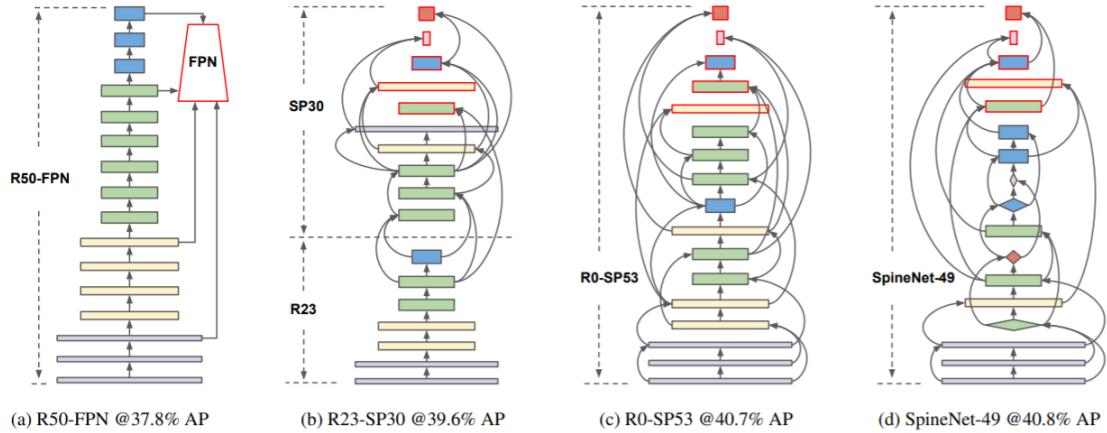


Figure 3: **Building scale-permuted network by permuting ResNet.** From (a) to (d), the computation gradually shifts from ResNet-FPN to scale-permuted networks. (a) The R50-FPN model, spending most computation in ResNet-50 followed by a FPN, achieves 37.8% AP; (b) R23-SP30, investing 7 blocks in a ResNet and 10 blocks in a scale-permuted network, achieves 39.6% AP; (c) R0-SP53, investing all blocks in a scale-permuted network, achieves 40.7% AP; (d) The SpineNet-49 architecture achieves 40.8% AP with 10% fewer FLOPs (85.4B vs. 95.2B) by learning additional block adjustments. Rectangle block represent bottleneck block and diamond block represent residual block. Output blocks are indicated by red border.

Рис. 30: (a) ResNet+FPN, (b-d) Различные архитектуры SpineNet [11]

Авторы утверждают, что архитектуры encoder-decoder не эффективны для генерации хороших признаков и предлагают свой вариант. Их основная идея в добавлении так называемых scale-permuted network, вместо привычных архитектур с постепенным уменьшением разрешения (scale-decreased network). (Рис. 31 (b))

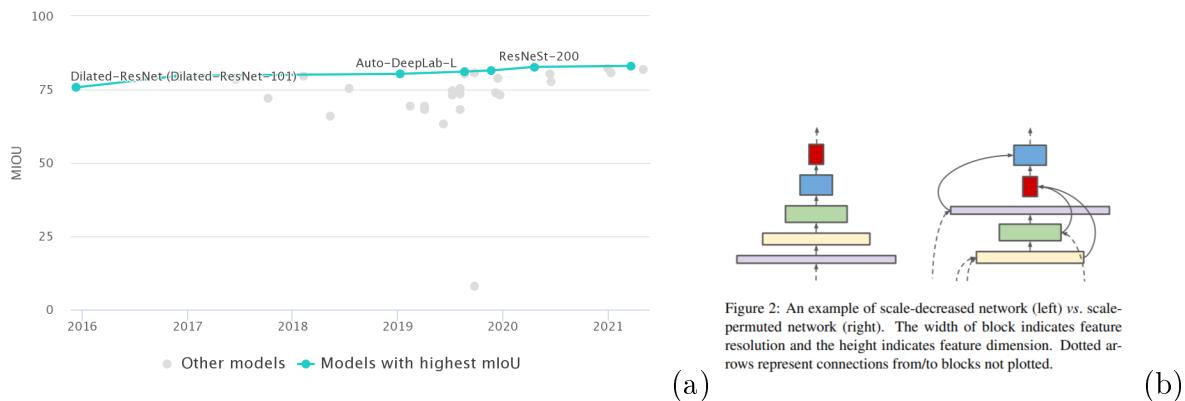


Рис. 31: (a) SOTA в сегментации (Cityscapes val) scource
 (b) Отличие scale-decreased network от scale-permuted network

Также авторы использовали поиск нейронных архитектур (Neural Architecture Search) и смогли достичь SOTA на датасете Cityscapes val. (Рис. 31)

Multi Scale Spatial Attention

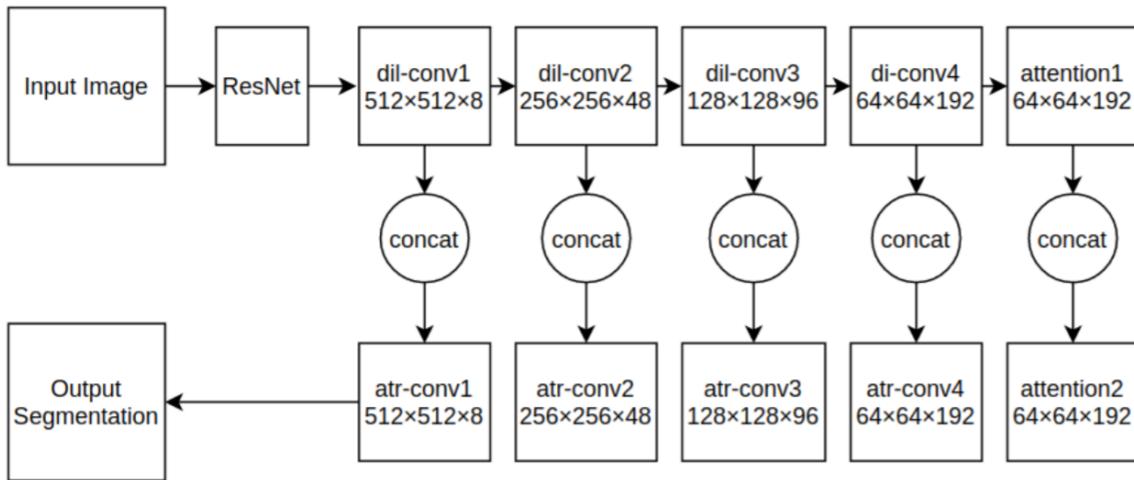


Рис. 32: Архитектура MSSA[13]

Помимо достижения лучшего качества сетей в задаче сегментации немаловажной задачей является разработка методов максимизирующих соотношение качество/скорость. Это принципиально для беспилотников, для которых важно хорошо понимать происходящее в реальном времени. Авторы предложили архитектуру, использующую ResNet в качестве backbone, модуль внимания (attention) и расширенные свертки. (Рис. 32)

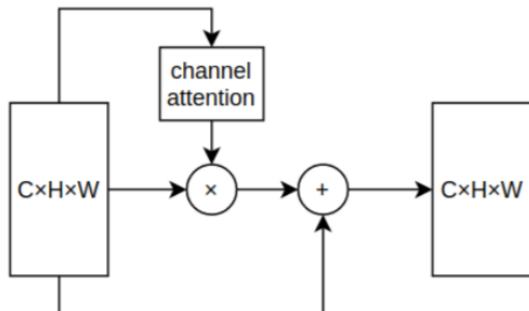


Рис. 33: Модуль attention [13]

Основная идея модуля внимания состоит в том, что на выходе из него мы получаем взвешенные созависимые карты признаков (то есть поканальные зависимости), которые суммируем с исходной картой признаков.

Model	InputSize	FLOPs	Params	Time(ms)	Frame(fps)	mIoU(%)
PSPNet (Zhao et al., 2017)	713×713	412.2G	250.8M	1288	0.78	81.2
DeepLab (Chen et al., 2017)	512×1024	457.8G	262.1M	4000	0.25	63.1
SegNet (Badrinarayanan et al., 2017)	640×360	286G	29.5M	16	16.7	57
ENet (Paszke et al., 2016)	640×360	3.8G	0.4M	7	135.4	57
CRF-RNN (Zheng et al., 2015)	512×1024	-	-	700	1.4	62.5
FCN-8S (Long et al., 2015)	512×1024	136.2G	-	500	2	63.1
FRRN (Pohlen et al., 2017)	512×1024	235G	-	469	0.25	71.8
ICNet (Zhao et al., 2018)	1024×2048	28.3G	26.5M	33	30.3	69.5
BiSeNet1 (Yu et al., 2018a)	768×1536	14.8G	5.8M	13	72.3	68.4
BiSeNet2 (Yu et al., 2018a)	768×1536	55.3G	49M	21	45.7	74.7
DFANet A (Li et al., 2019)	1024×1024	3.4G	7.8M	10	100	71.3
DFANet B (Li et al., 2019)	1024×1024	2.1G	4.8M	8	120	67.1
Ours	1024×1024	1.8G	5.5M	6	134	72.4

Рис. 34: Сравнение качества и скорости работы [13]

Авторам удалось добиться впечатляющей скорости – 135 fps при небольшом количестве параметров и лучшем $mIoU = 72.4$ для сетей, со скоростью >100 fps.

Заключение

В данном эссе была дана постановка задачи сегментации, описаны основные определения, рассмотрены как классические, так и самые современные методы решения данной задачи. Описана главная идея современных методов, заключающаяся в использовании полносверточной архитектуры нейросети (FCN) на примере: UNet, TernausNet, DenseNet+Unet, GridNet, DeepLabv. Каждая из перечисленных архитектур привносит новые идеи, которые сильно влияют на направление исследований в этой области. Среди разобранных: encoder-decoder архитектура, skip-connections, "автоматический выбор пути", dilated convolution. Также описаны архитектуры самых современных нейросетей [11, 13], достигающие SOTA на некоторых бенчмарках.

Список литературы

- [1] Александр Дьяконов. Сегментация и поиск изображений.
https://github.com/Dyakonov/DL/blob/master/2020/DL2020_034segmentation_06n.pdf
- [2] Kirillov A. et al. Panoptic segmentation //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2019. – C. 9404-9413.
- [3] Jeremy Jordan. An overview of semantic image segmentation.
<https://www.jeremyjordan.me/semantic-segmentation>
- [4] COCO Dataset.
<http://host.robots.ox.ac.uk/pascal/VOC/voc2012/#devkit>
- [5] Nikita Shiledarbaxi. Semantic vs Instance vs Panoptic: Which Image Segmentation Technique To Choose <https://analyticsindiamag.com/semantic-vs-instance-vs-panoptic-which-image-segmentation-technique-to-choose/>
- [6] Семантическая сегментация: краткое руководство
<https://neurohive.io/ru/osnovy-data-science/semantic-segmention/>
- [7] Long J., Shelhamer E., Darrell T. Fully convolutional networks for semantic segmentation //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2015. – C. 3431-3440.
- [8] Ronneberger O., Fischer P., Brox T. U-net: Convolutional networks for biomedical image segmentation //International Conference on Medical image computing and computer-assisted intervention. – Springer, Cham, 2015. – C. 234-241.
- [9] AUGUSTUS ODENA et al. Deconvolution and Checkerboard Artifacts.
<https://distill.pub/2016/deconv-checkerboard/>
- [10] Cyborg Nitr. MIoU Calculation <https://medium.com/@cyborg.team.nitr/miou-calculation-4875f918f4cb>
- [11] Du X. et al. Spinenet: Learning scale-permuted backbone for recognition and localization //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2020. – C. 11592-11601.
- [12] Jégou S. et al. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation //Proceedings of the IEEE conference on computer vision and pattern recognition workshops. – 2017. – C. 11-19.
- [13] Sagar A., Soundrapandian R. K. Semantic Segmentation With Multi Scale Spatial Attention For Self Driving Cars //arXiv preprint arXiv:2007.12685. – 2020.
- [14] Chen L. C. et al. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs //IEEE transactions on pattern analysis and machine intelligence. – 2017. – Т. 40. – №. 4. – C. 834-848.

- [15] Chen L. C. et al. Rethinking atrous convolution for semantic image segmentation //arXiv preprint arXiv:1706.05587. – 2017.
- [16] Sun K. et al. High-resolution representations for labeling pixels and regions //arXiv preprint arXiv:1904.04514. – 2019.
- [17] Fourure D. et al. Residual conv-deconv grid network for semantic segmentation //arXiv preprint arXiv:1707.07958. – 2017.
- [18] Yu F., Koltun V. Multi-scale context aggregation by dilated convolutions //arXiv preprint arXiv:1511.07122. – 2015.
- [19] Wang Z., Ji S. Smoothed dilated convolutions for improved dense prediction //Data Mining and Knowledge Discovery. – 2021. – C. 1-27.
- [20] Yuvrajchopra25. Project-8-Image-Segmentation-using-Sklearn-and-K-Means <https://github.com/Yuvrajchopra25/Project-8-Image-Segmentation-using-Sklearn-and-K-Means>
- [21] Divyanshu Mishra. Transposed Convolution Demystified. <https://towardsdatascience.com/transposed-convolution-demystified-84ca81b4baba>
- [22] Iglovikov V., Shvets A. Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation //arXiv preprint arXiv:1801.05746. – 2018.