



УНИВЕРСИТЕТ ИТМО

Сегментация изображений

Ефимова Валерия Александровна

vefimova@itmo.ru

19.10.2022

ОБРАЗОВАТЕЛЬНЫЕ ПРОГРАММЫ В ОБЛАСТИ
ТЕХНОЛОГИЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

- Сегментация
 - Задача сегментации.
 - Функции ошибки для сегментации.
 - Сети для сегментации изображений (FCN, U-Net, PSPNet, DeepLab).
- Обработка видеопотока.
 - Камеры и специфика обработки видео.
 - Оптический поток, классические и нейросетевые методы его подсчета.
 - Трекинг объектов (SORT, DeepSORT).
 - DeepFake

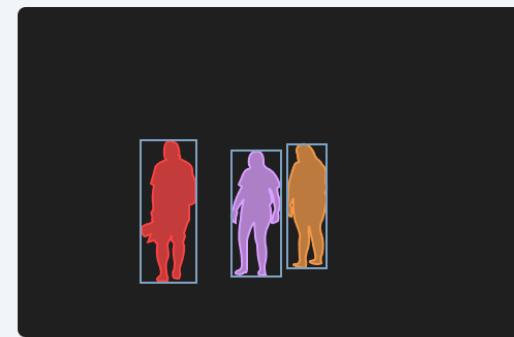
- **Семантическая сегментация** (Semantic Segmentation) – каждому пикселью сопоставлен класс, соответствующий объекту, который в нем находится, нет разницы между объектами.
- **Сегментация сущностей** (Instance Segmentation) – разделяем объекты одного класса на разные сущности, некоторые пиксели (в которых нет объектов, фон) могут быть не размечены.
- **Паноптическая сегментация** (Panoptic Segmentation) – есть разделение на объекты, классифицированы все пиксели (семантическая + сущности).



(a) Image



(b) Semantic Segmentation

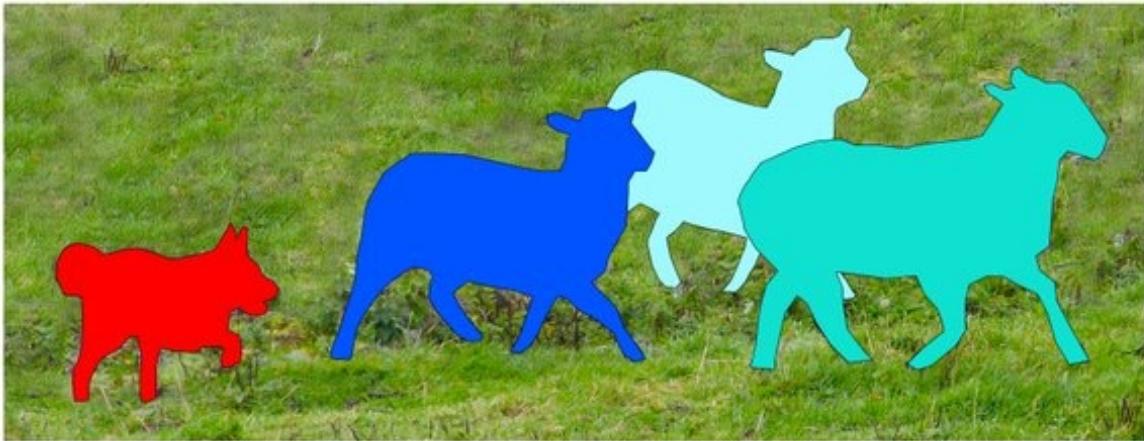


(c) Instance Segmentation

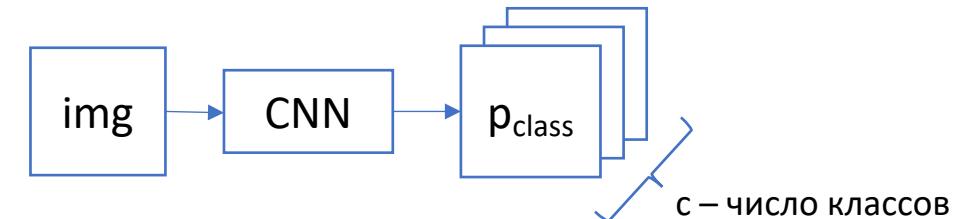


(d) Panoptic Segmentation

Какая сегментация?



- Вероятность каждого класса в отдельном канале.



- Ранее уменьшали размерности, чтобы охватить изображение целиком.
Проблема: тогда найдем только мелкие объекты или будет много вычислений.

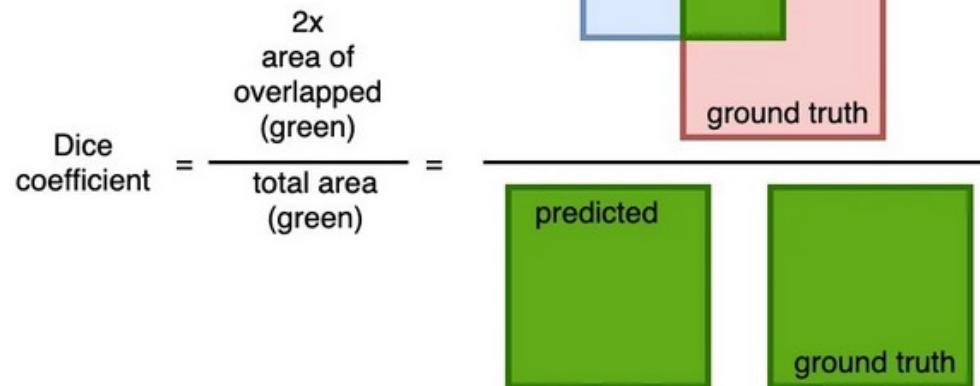
- **Кросс-энтропия:** $\mathcal{L} = \frac{1}{W \cdot H} \sum_{x,y} -\log p_{t_{x,y},x,y},$

где c – индекс класса, $p_{c,x,y}$ – предсказание, $t_{x,y}$ – реальные метки.

- **Dice:**

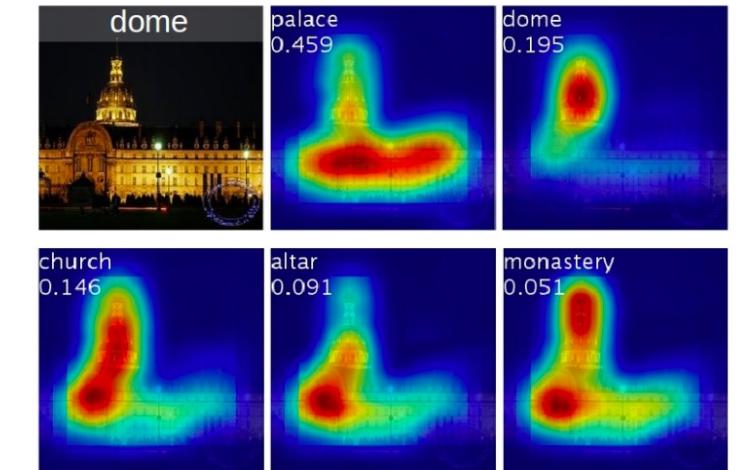
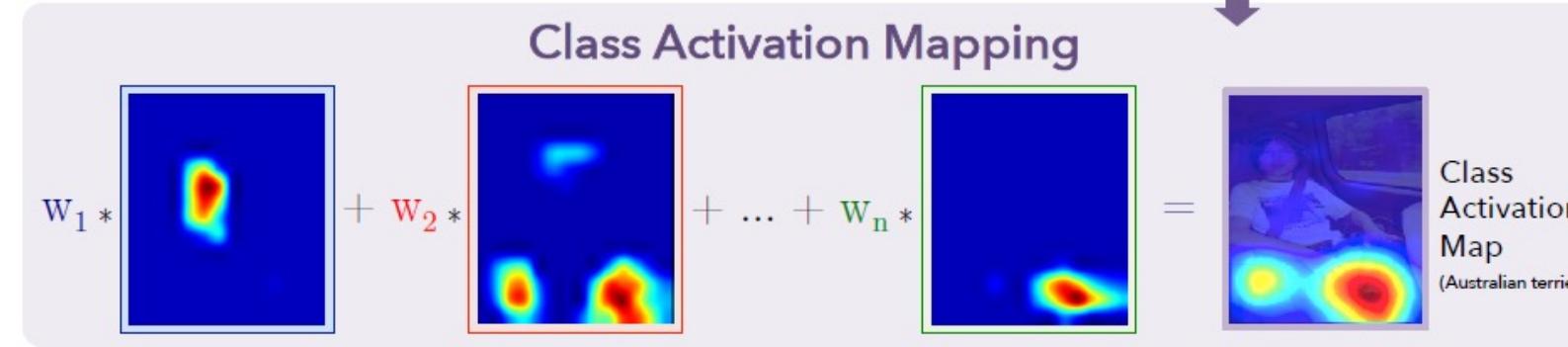
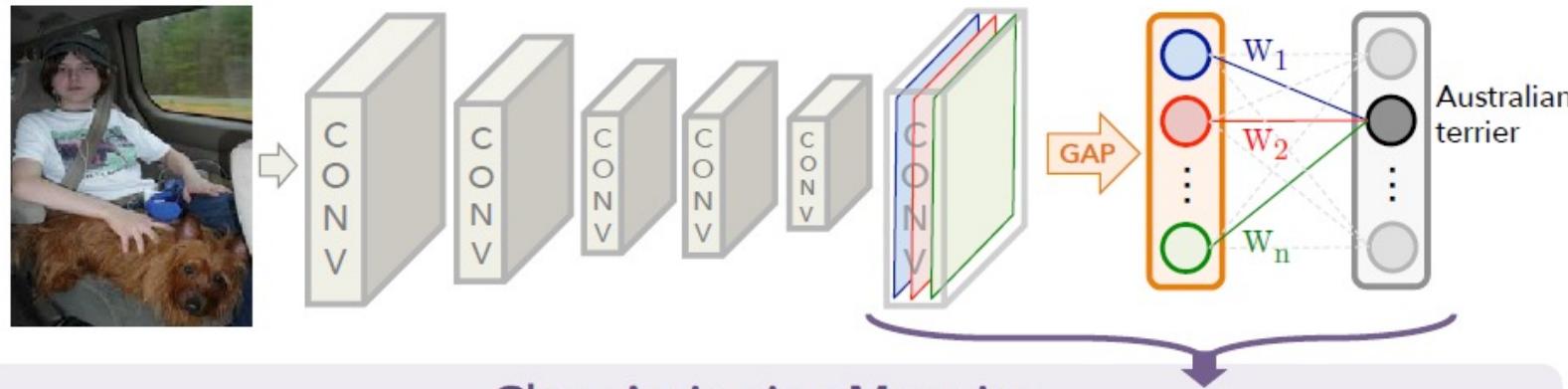
$$Dice = \frac{2}{W \cdot H} \sum_{x,y} \frac{|p_{x,y} \cdot t_{x,y}|}{|p_{x,y}| + |t_{x,y}|},$$

где $p_{x,y} = \{0, 1\}$.

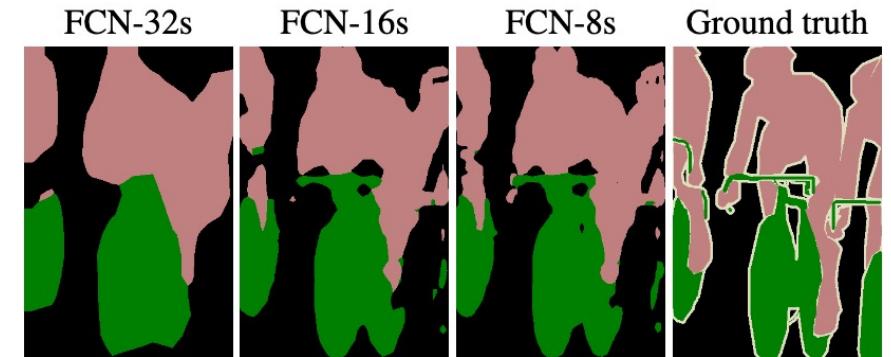
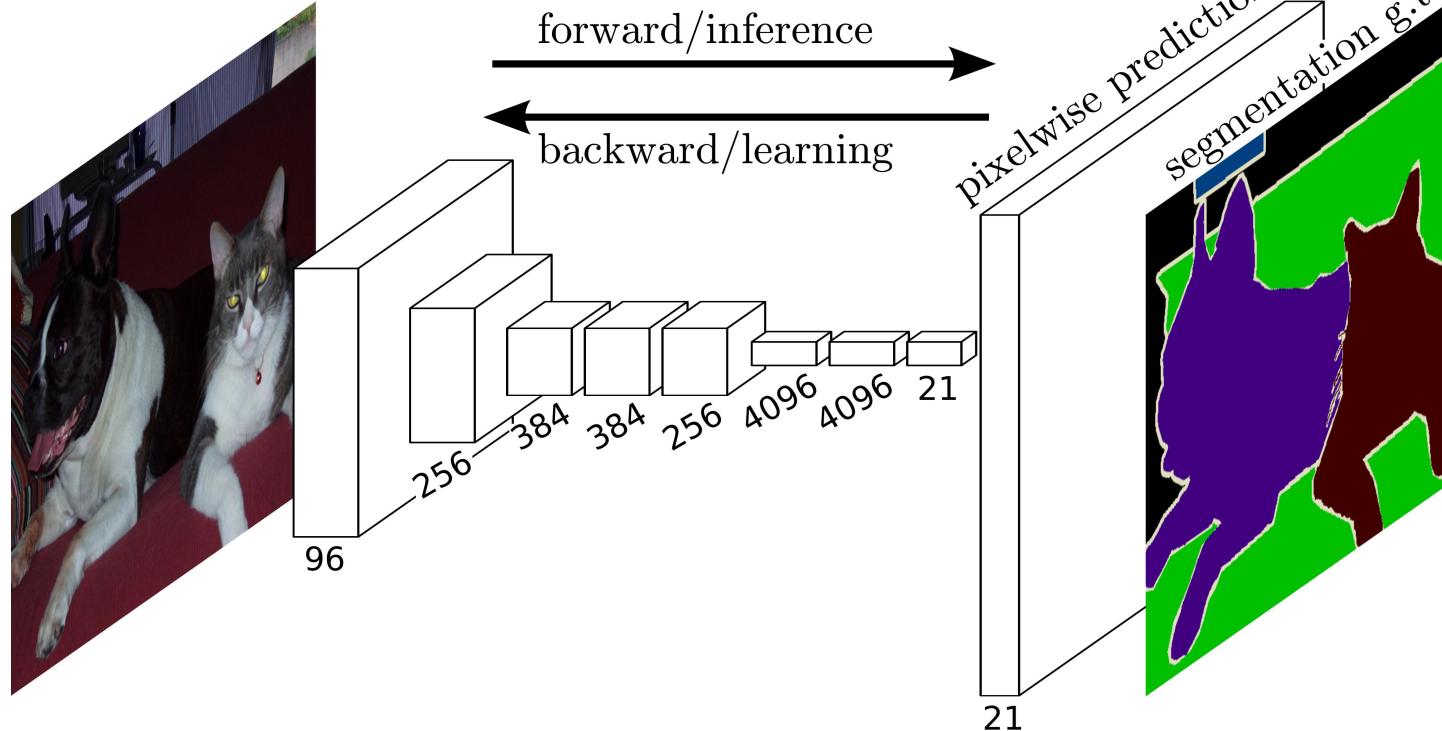


- Проблема: Dice – невыпуклая функция.
- Сначала оптимизируем CE, потом Dice.
- Dice проще для оптимизации, чем IoU.
- Если IoU используется как метрика, то как функцию ошибки лучше использовать Dice.

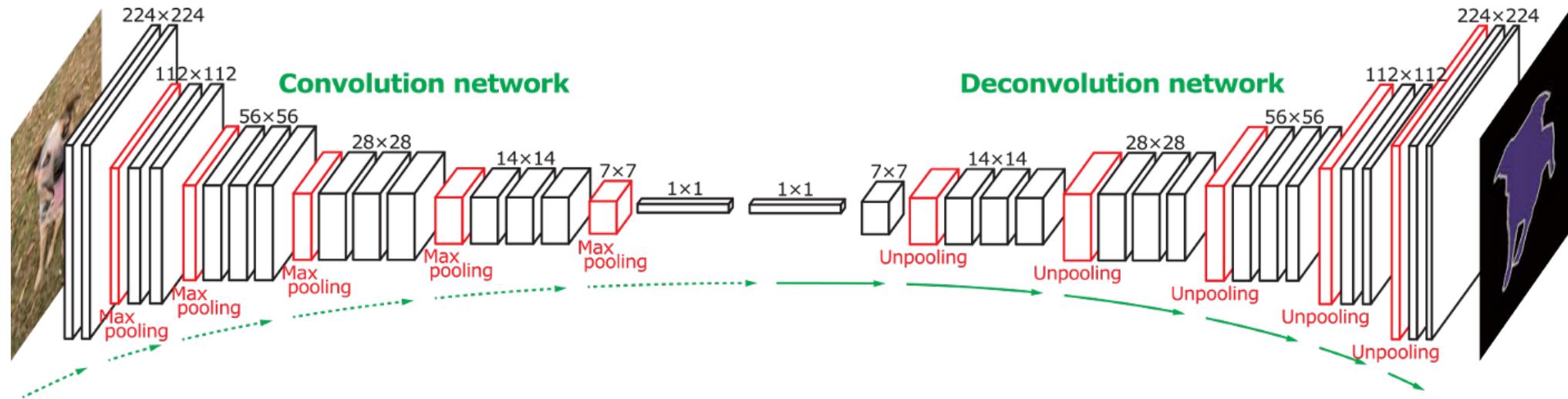
Карты активации классов (class activation maps)



У backbone для классификации отрежем последние полно связные слои.

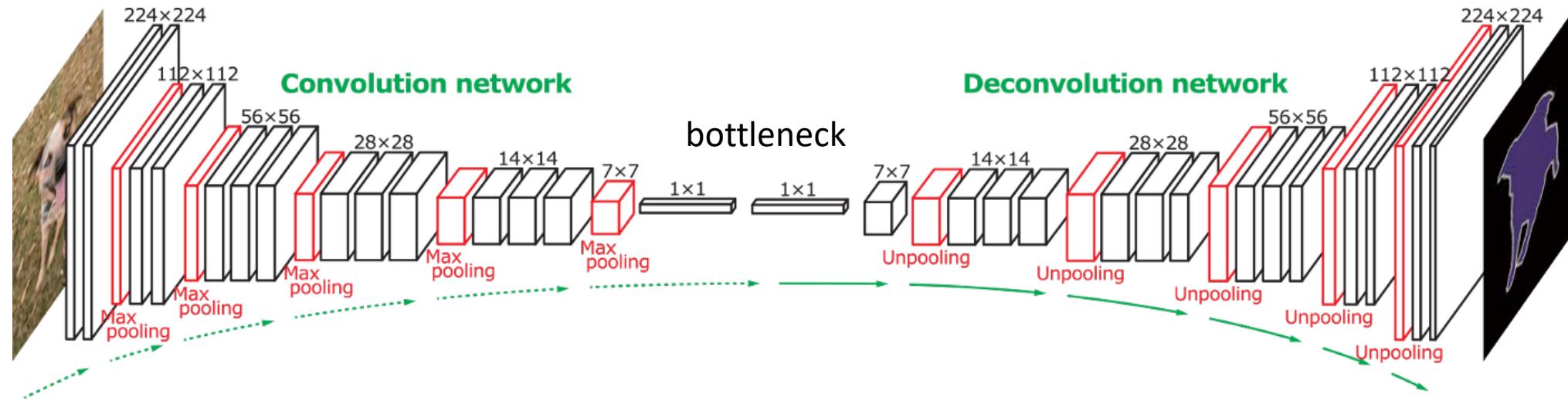


Max pooling – необратимая операция.

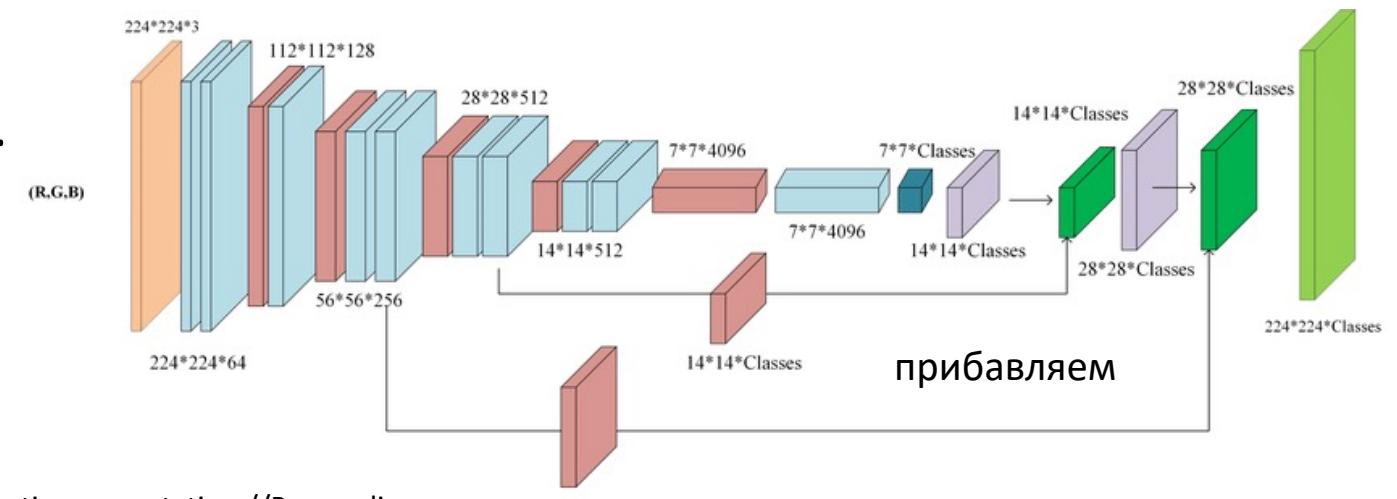


1. Ближайшие соседи (не обучается)
2. Би-линейная интерполяция (не обучается)
3. Max Unpooling (не обучается)
4. Transposed convolution
5. Pixel shuffle

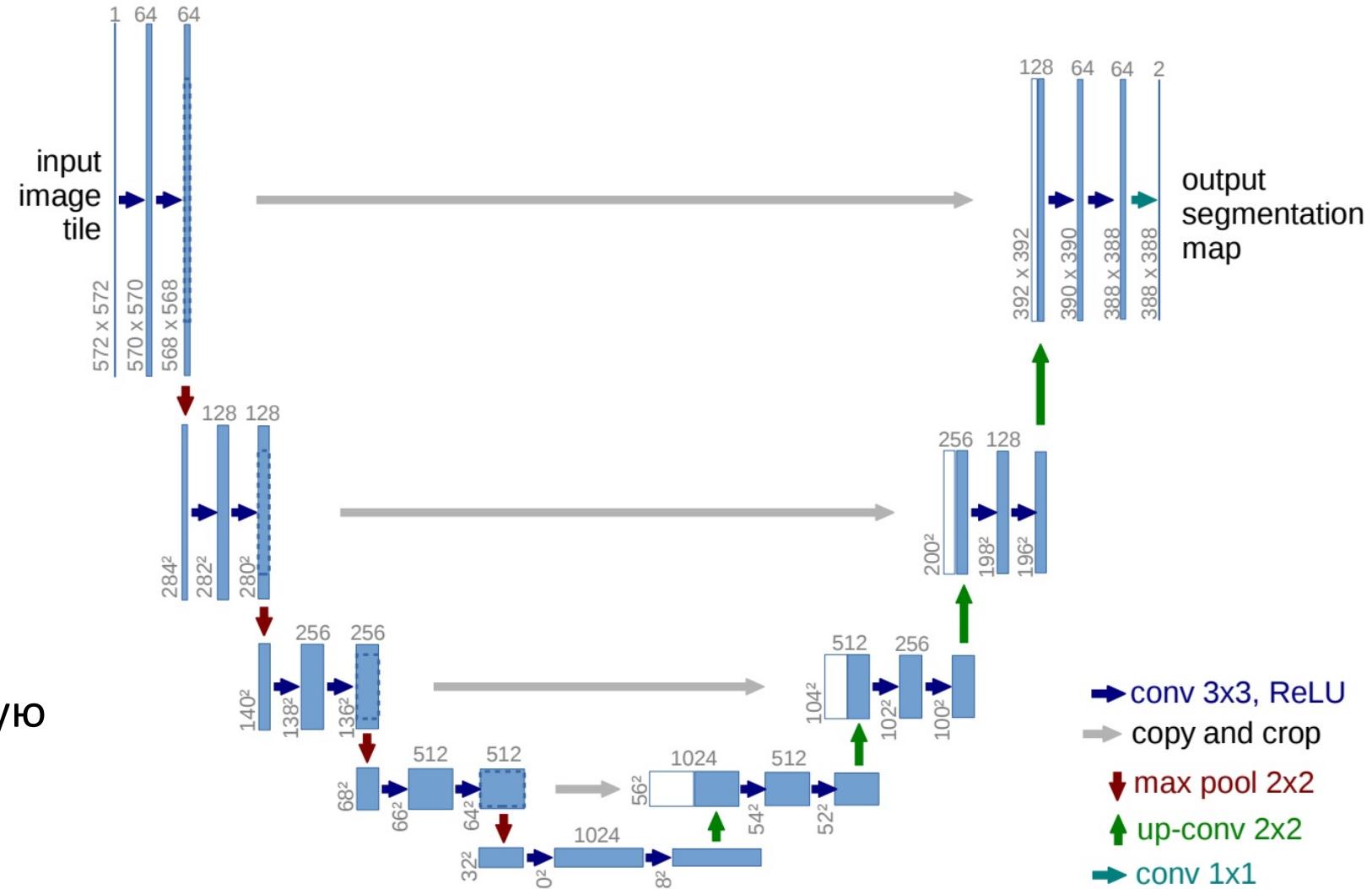
Max pooling – необратимая операция.

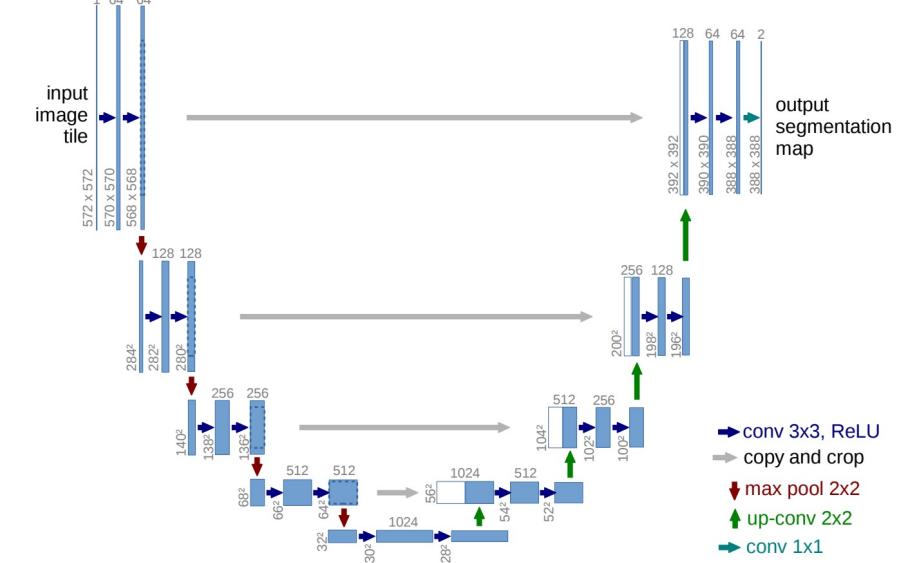
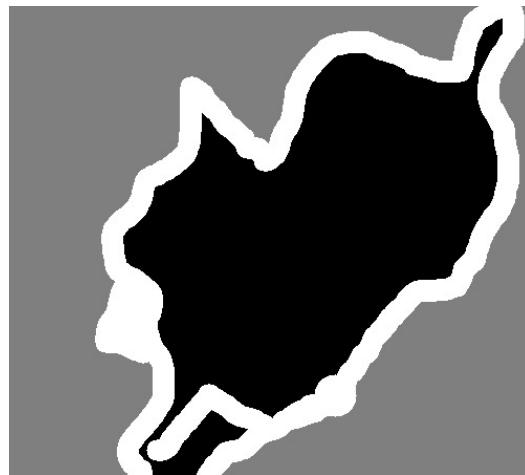
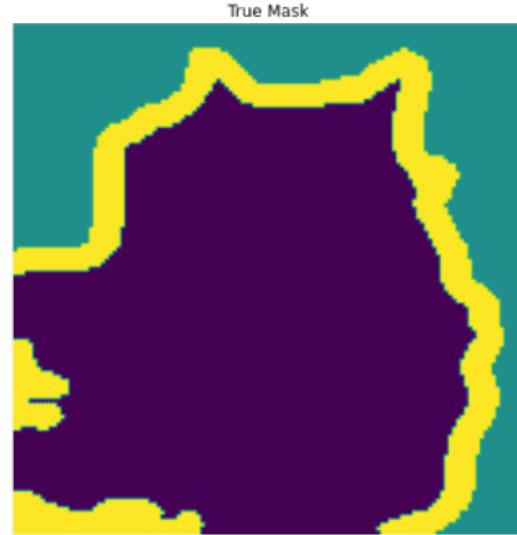
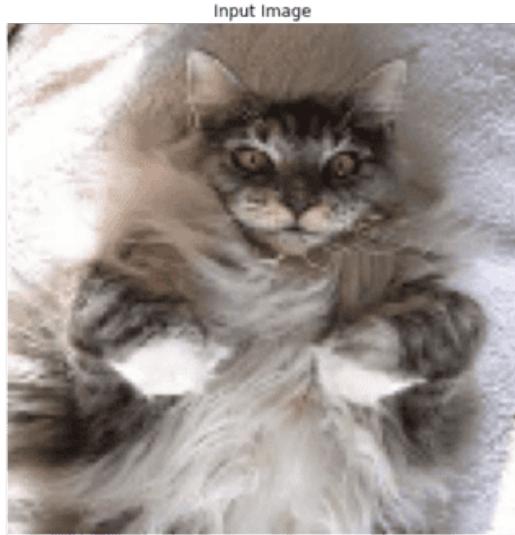


Учтем отброшенную информацию.
Добавляем дополнительные детали.



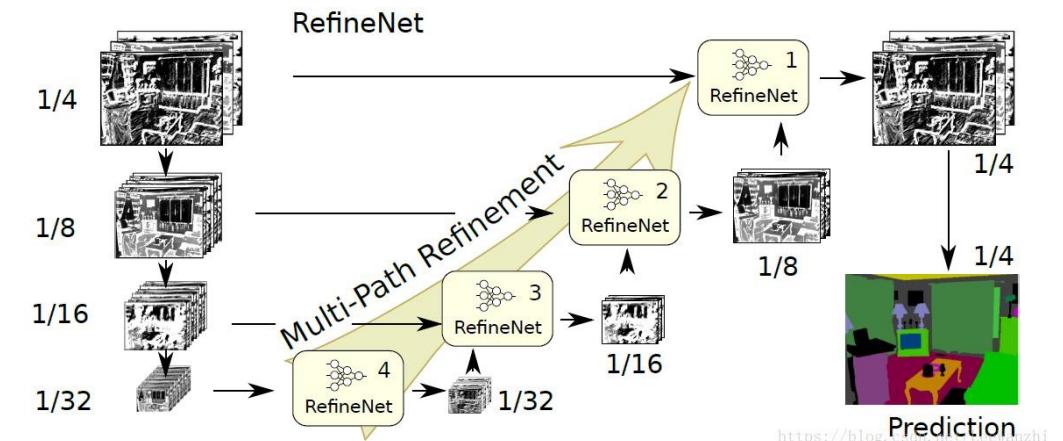
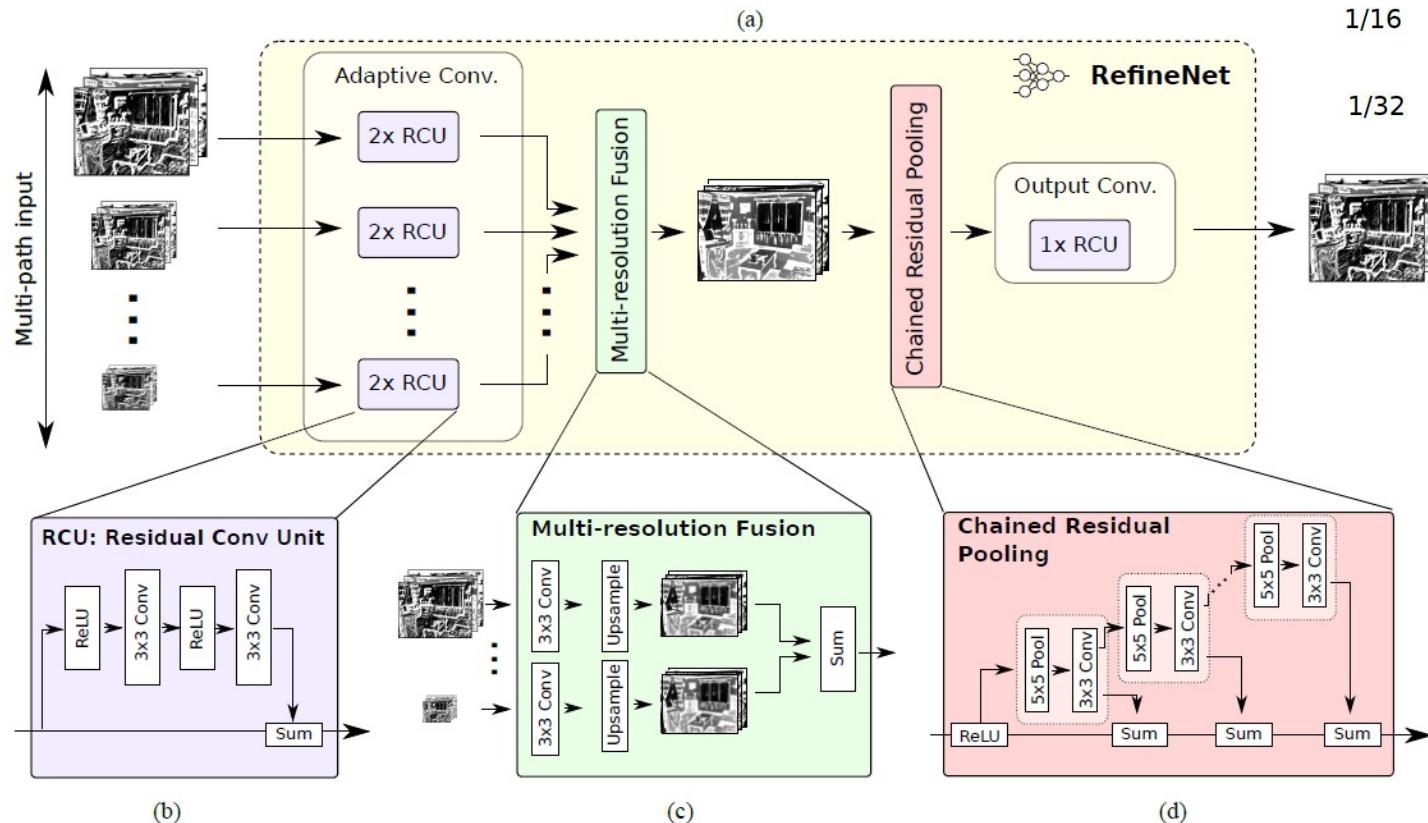
- Убираем bottleneck.
- Skip-connections нужны, чтобы информация не терялась, но из-за них проблемы с обучением.
- Перенос знаний: в качестве кодировщика стоит использовать предобученную сеть для классификации, например ResNet.





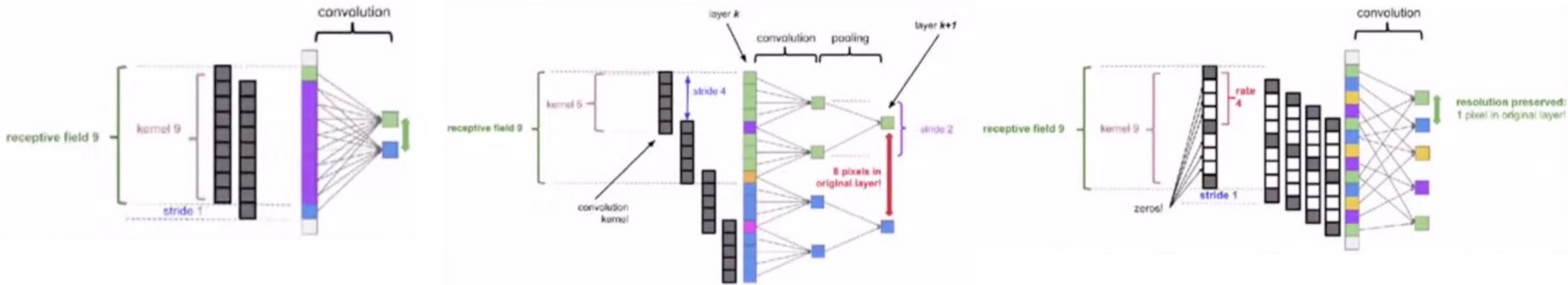
<https://pyimagesearch.com/2021/11/08/u-net-training-image-segmentation-models-in-pytorch/>

- Multi-Input U-Net
- 3D U-Net
- RefineNet – U-Net с Refine блоками.



- Получаются гладкие сегментации.
- В Chained Residual Pooling можно заменить свертки 3x3 на 1x1, тогда быстродействие увеличится x10.

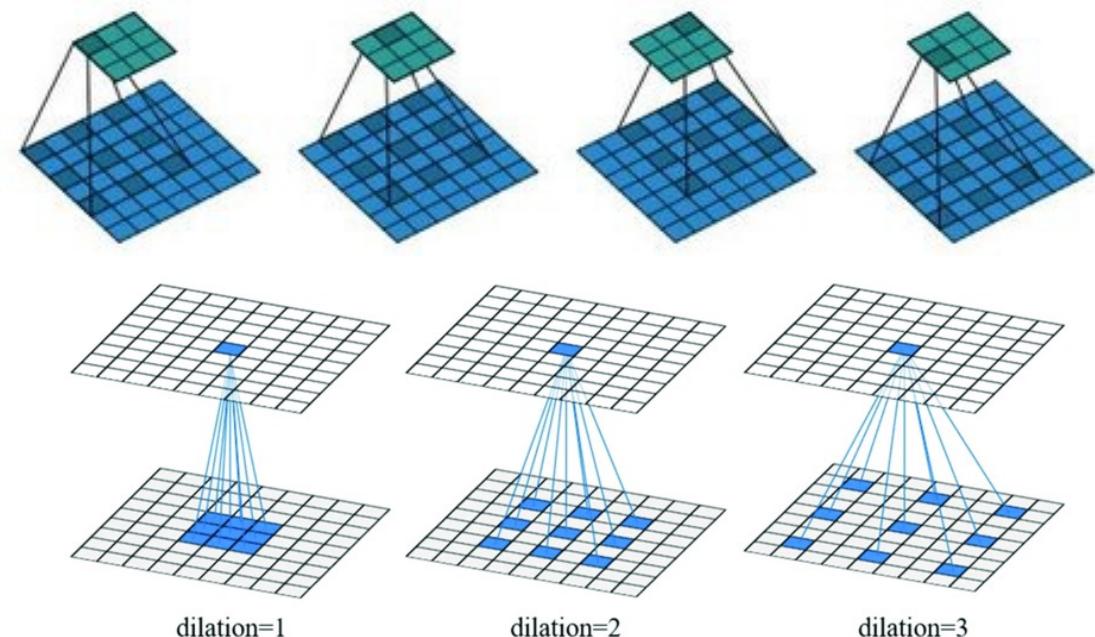




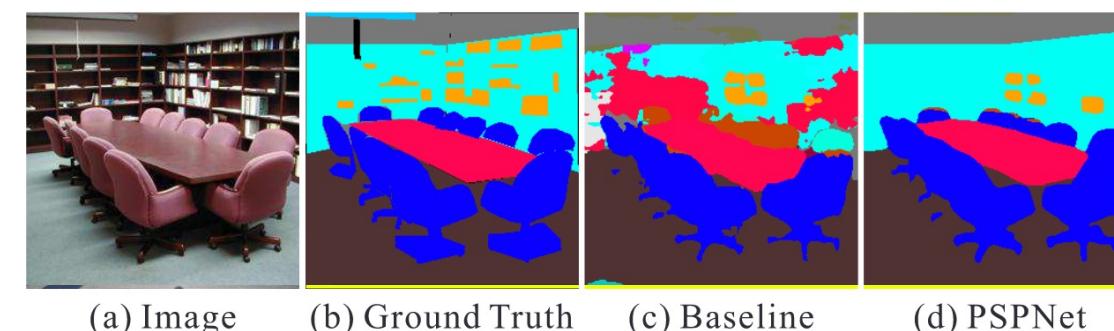
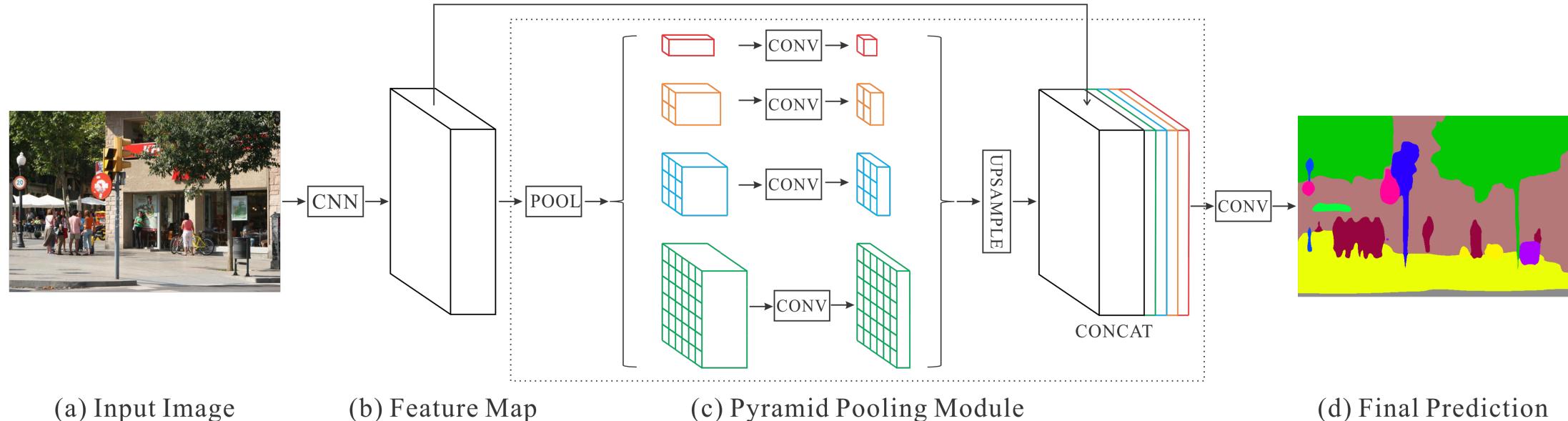
- Расширение ядра свёртки с помощью добавления пропусков между элементами.

$$y_i = \sum_{k=1}^K x_{i+dilation \cdot k} \cdot w_k$$

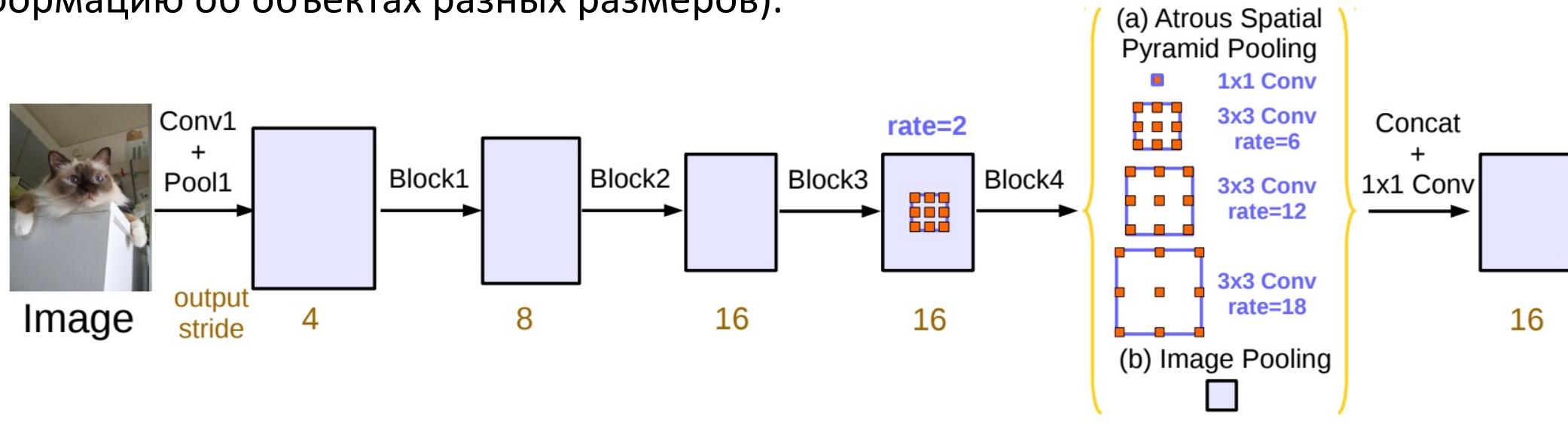
- Такое же поле восприятия (receptive field).
- Все выходы влияют на результат, но меньше раз обрабатываем каждый пиксель.
- Вычислительные ресурсы расходуются более эффективно.



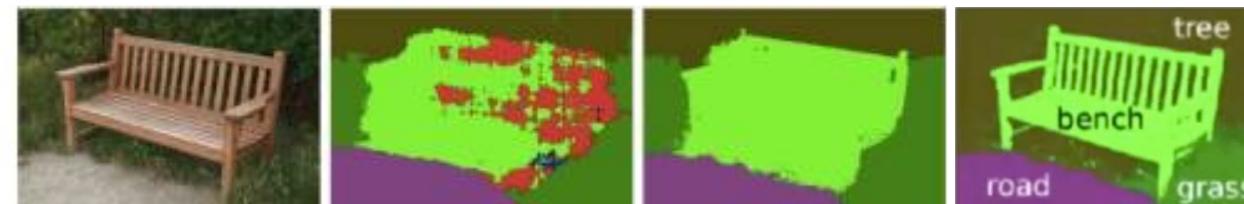
Свертки разного размера из одной карты признаков.



- Конкatenируем dilated convolutions разного размера (выделяют информацию об объектах разных размеров).



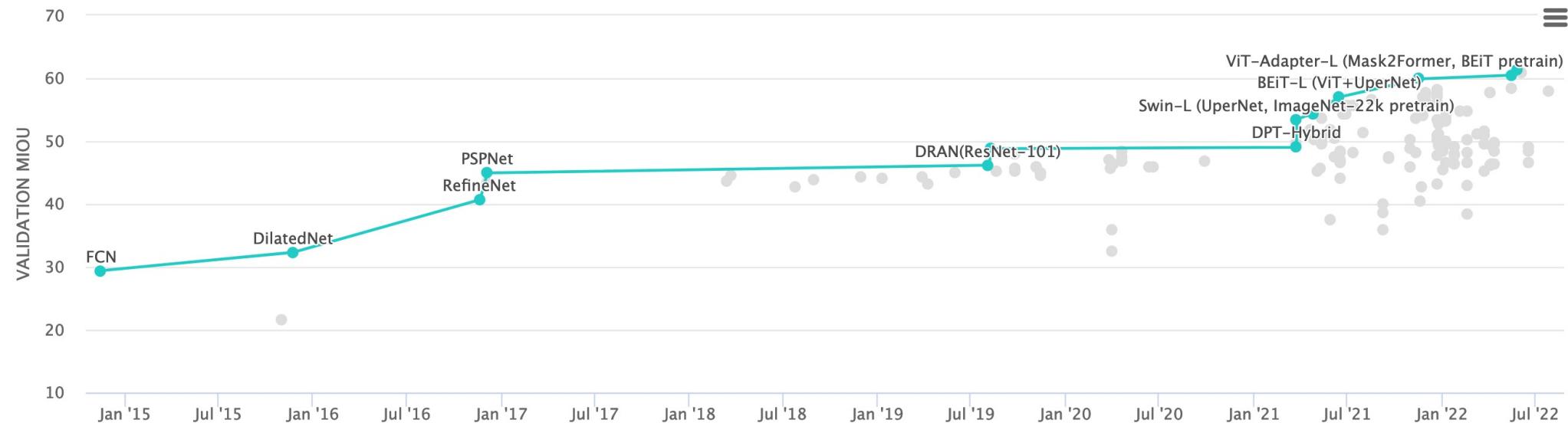
- Постпроцессинг
(Conditional Random Field, CRF)



Chen L. C. et al. Semantic image segmentation with deep convolutional nets and fully connected crfs //arXiv preprint arXiv:1412.7062. – 2014.

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. – С. 834-848.

Semantic Segmentation on ADE20K



- Промышленность
- Автономное вождение
- ...

- Сегментация бывает семантическая, сущностей и паноптическая.
- Модели сегментации тренируют с ошибкой кросс-энтропии и Dice, качество оценивают с помощью IoU.
- Сначала сегментацию строили с помощью FCN, но потом предложили U-Net и модификации.
- Для сегментации SOTA-модели используют dilated свертки.
- Модели семейства DeepLab используют не только dilated свертки, но и постпроцессинг.

- https://colab.research.google.com/github/pytorch/pytorch.github.io/blob/master/assets/hub/pytorch_vision_deeplabv3_resnet101.ipynb
- <https://colab.research.google.com/github/CSAILVision/semantic-segmentation-pytorch/blob/master/notebooks/DemoSegmenter.ipynb#scrollTo=98Y6Zs3pcvAI>
- https://github.com/qubvel/segmentation_models.pytorch



УНИВЕРСИТЕТ ИТМО

Спасибо за внимание!

ОБРАЗОВАТЕЛЬНЫЕ ПРОГРАММЫ В ОБЛАСТИ
ТЕХНОЛОГИЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА



УНИВЕРСИТЕТ ИТМО

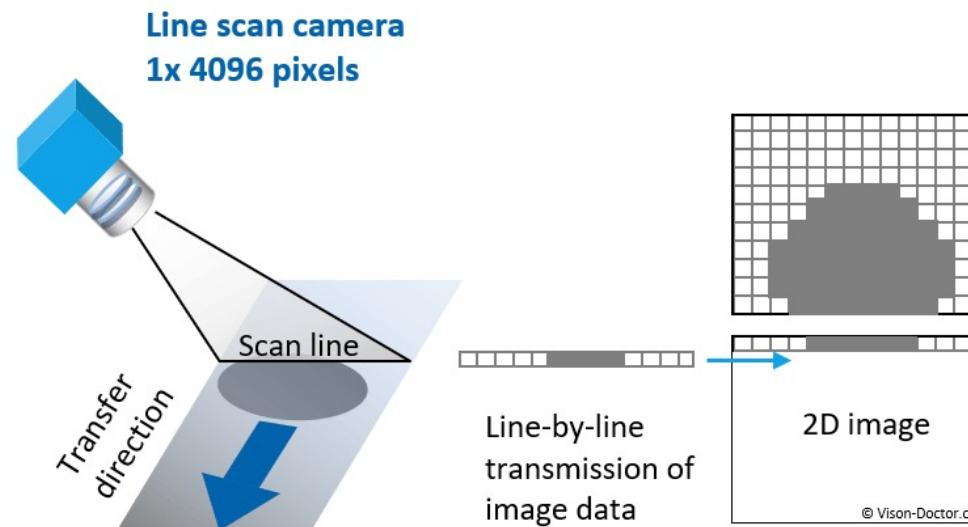
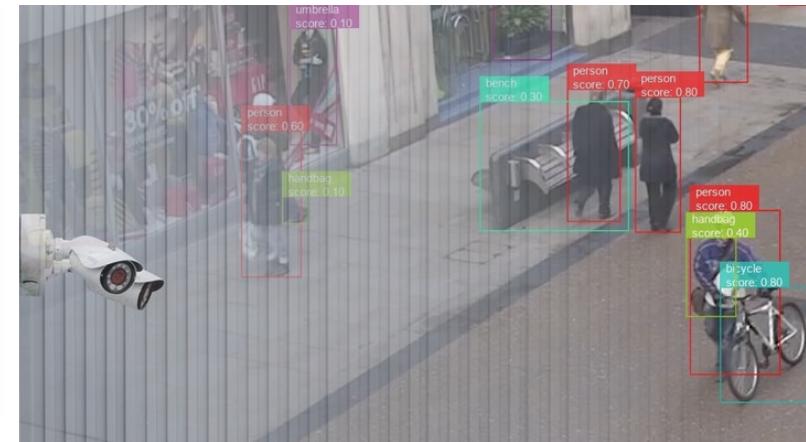
Обработка видеопотока

Ефимова Валерия Александровна

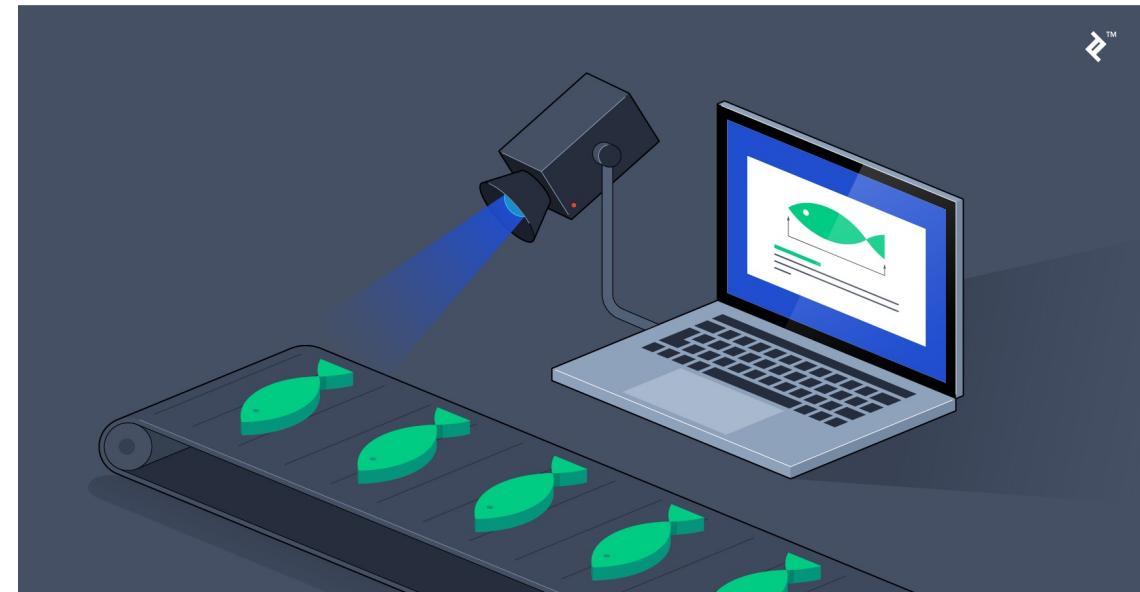
vefimova@itmo.ru

19.10.2022

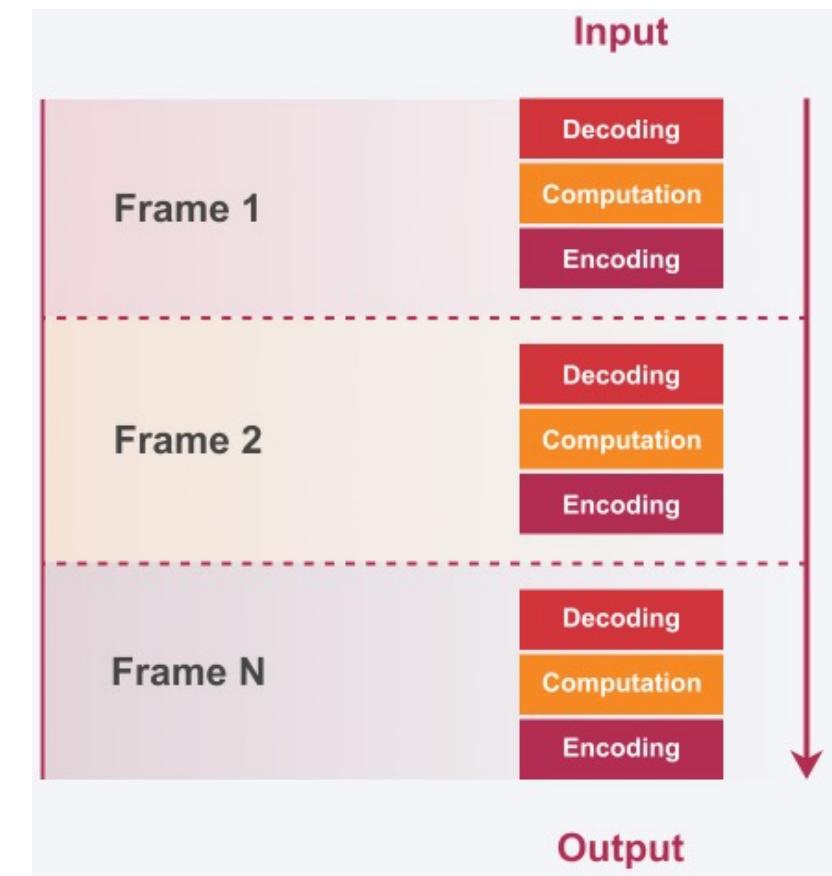
- Камеры наружного видеонаблюдения (CCTV)
 - + Совместимость, низкая стоимость
 - Оптические искажения
- Линейные камеры (line scan) – формируют изображение из линий
 - + Нет искажений, высокое разрешение (до 12K), высокая частота, можно настроить частоту кадров, диафрагму и фокусное расстояние.
 - Высокая стоимость



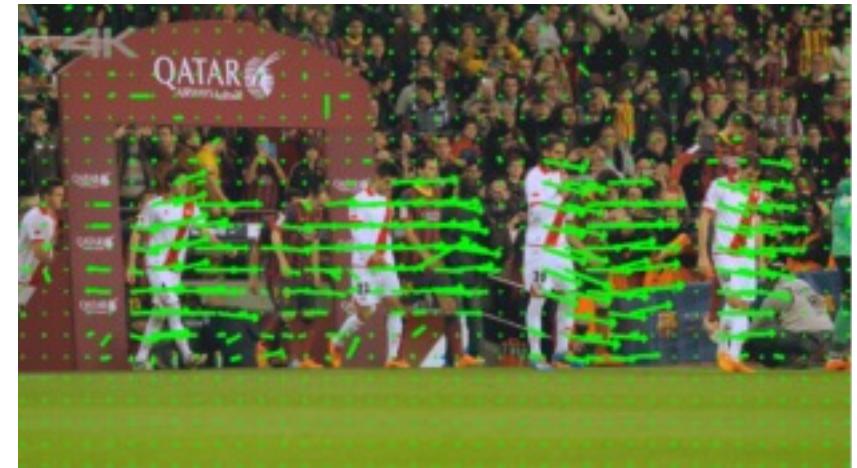
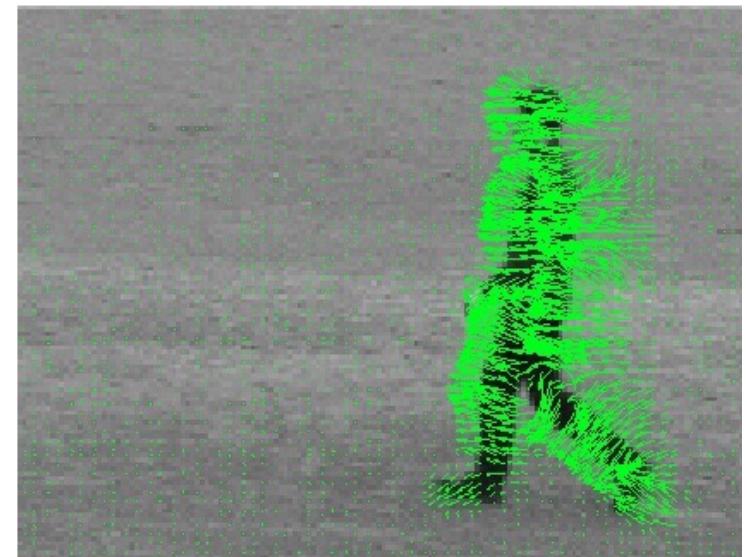
- Видео — это просто упорядоченная последовательность кадров (изображений) одинакового разрешения, захватываемая и отображаемая с заданной частотой (кадры в секунду, frames per second, FPS).
- Обработка видеопотока позволяет решить задачи:
 - Распознавание видео (video recognition).
 - Нахождение объектов на видео.
 - Распознавание лиц.
 - Наблюдение за перемещениями объектов.
 - Понимание видео (video understanding).
 - Определение действий на видео.
 - Улучшение качества видео.
 - Стабилизация.
 - Уменьшение размытия (deblurring).
 - Удаление шума (denoizing).
 - Суммаризация видео.
 - Создание и определение фейков.



- Обработка видео – последовательность операций, примененная в какой-то момент времени.
- Она включает в себя декодирование, вычисления и кодирование.
- Но важно учитывать скорость и точность обработки (и специфику задачи).
- Ускорение: параллельная обработка, оптимизация алгоритмов.
 - Использовать mixed precision (float16 для некоторых слоев, вместо float32).
 - Более быстрый инференс (TensorRT, ONNX).
 - Уменьшение размера кадра.
 - Обработка батча кадров.



- Объекты на соседних кадрах находятся в примерно одинаковых положениях.
- Нужно понять, куда сместилась точка.
- Нужно определить положение объекта в пикселе (x, y) начального кадра на следующем кадре – (x', y') .
- В мозге задача оптического потока легко решается.
- Нужен не только для трекинга объектов:
 - Компенсация тряски мобильного телефона.
 - Интерполяция между кадрами.
 - Стереозрение.
 - 3D-модель по видео.
- Наборы данных: MiddleBurry, KITTY, Sintel, Flying Chairs.

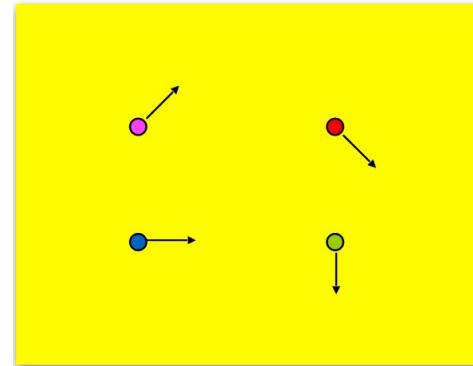


- Изображения I_1 и I_2 , \vec{u} – поток. Цвет и его насыщенность должны остаться прежними:

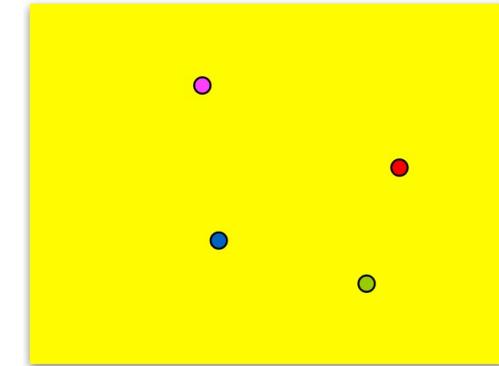
$$I_1(x + u_x, y + u_y, t) = I_2(x, y, t + 1).$$

$$\frac{\partial I}{\partial t} \cdot dt + \frac{\partial I}{\partial x} \cdot dx + \frac{\partial I}{\partial y} \cdot dy = 0$$

- Пусть $dt = 1$, $dx = u_x$, $dy = u_y$.
- Предположим, что соседние пиксели движутся примерно одинаково.
- Просуммируем по всем пикселям изображения и будем решать задачу оптимизации.



I_1



I_2

$$\sum_{i,j} g(x_i, y_j) \left[u_x \cdot \frac{\partial I(x_i, y_j, t)}{\partial x} + u_y \cdot \frac{\partial I(x_i, y_j, t)}{\partial y} + \frac{\partial I(x_i, y_j, t)}{\partial y} \right] \rightarrow \min_{u_x, u_y}$$

- Задача минимизации: берем производные и приравниваем их нулю.
- Можно решить систему уравнений аналитически.

- Чтобы понять, куда сместилаясь точка, возьмем ее окрестность и будем искать на втором изображении.
- Накладываем такой паттерн на все точки второго изображения и считаем корреляцию.

$$r_{ij} = \frac{\sum_m \sum_n [f(m + i, n + j) - \bar{f}][g(m, n) - \bar{g}]}{\sqrt{\sum_m \sum_n [f(m, n) - \bar{f}]^2 \sum_m \sum_n [g(m, n) - \bar{g}]^2}}.$$

- Она будет максимальной в том месте, которое наиболее похоже на паттерн.
- Но вычисление требует очень много ресурсов → считаем только для некоторых точек (например, окрестности).
- Проблемы: поворот, несколько похожих частей изображения.



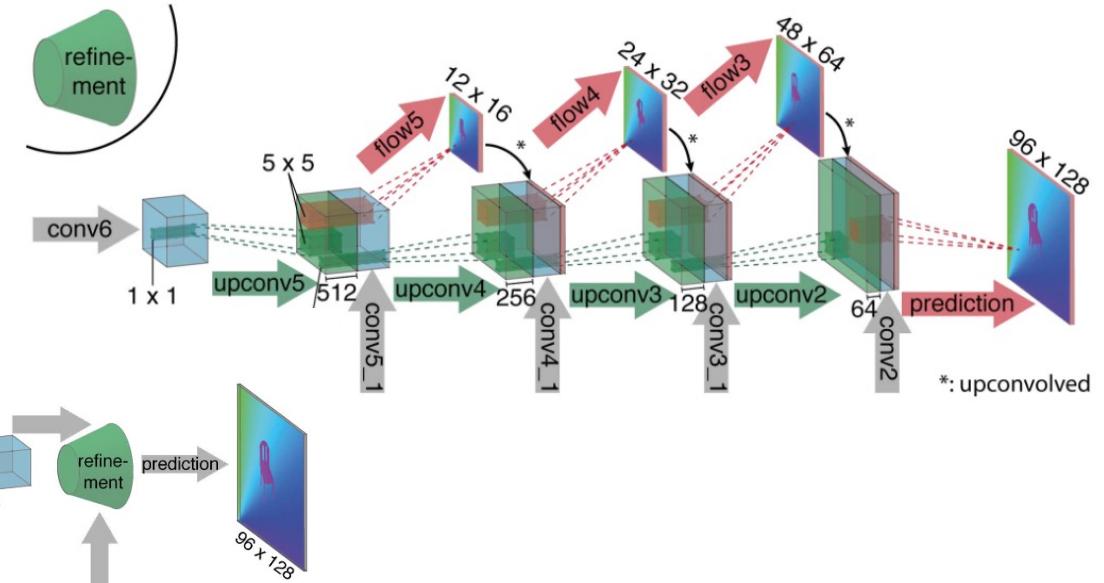
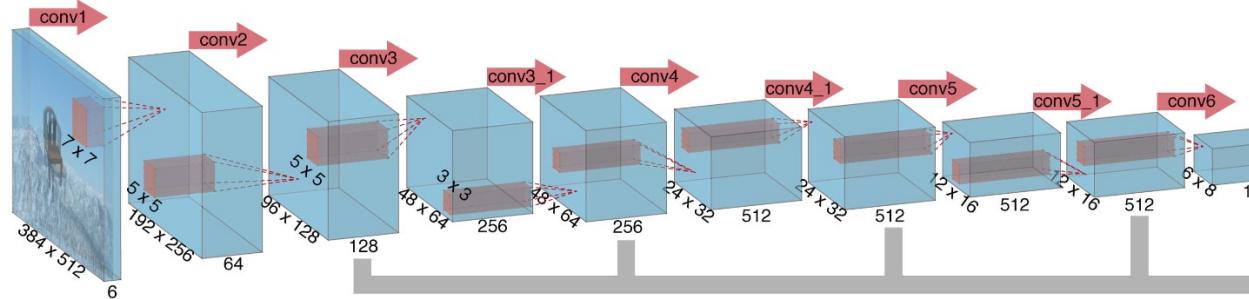
Евклидово расстояние между целевой точкой и предсказанной.

$$\mathcal{L} = \frac{1}{HW} \sum_{i=1}^{HW} \|\hat{u}_i - u_i\|_2,$$

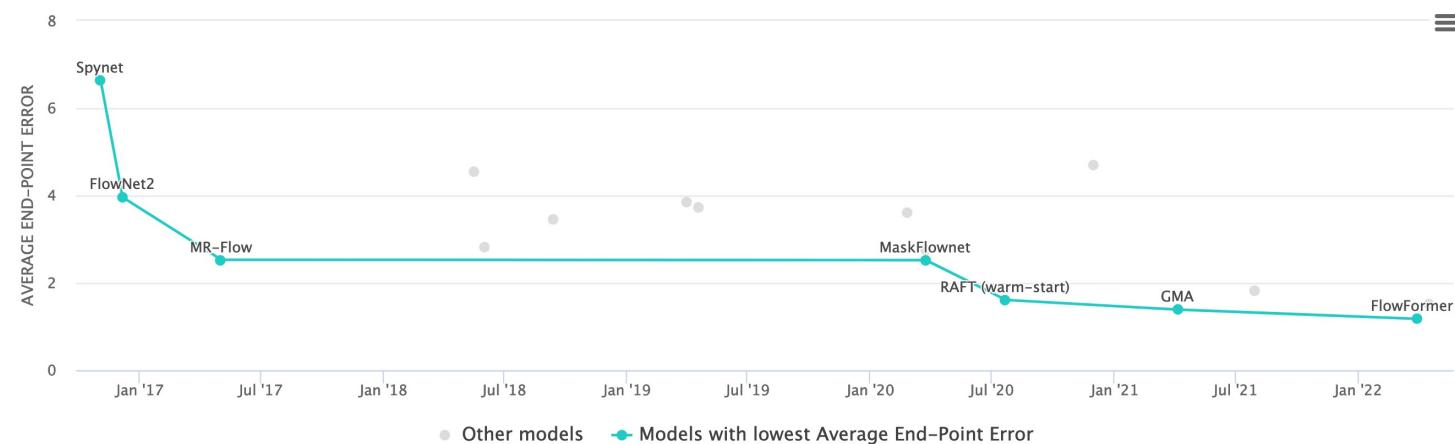
где \hat{u}_i – предсказанный оптический поток, u_i – настоящий оптический поток.

- FlowNet^[1]

- Кодировщик: конкатенируем изображения, пропускаем через CNN (VGG 16).
- Декодировщик: на всех шагах дополнительно предсказываем оптический поток.



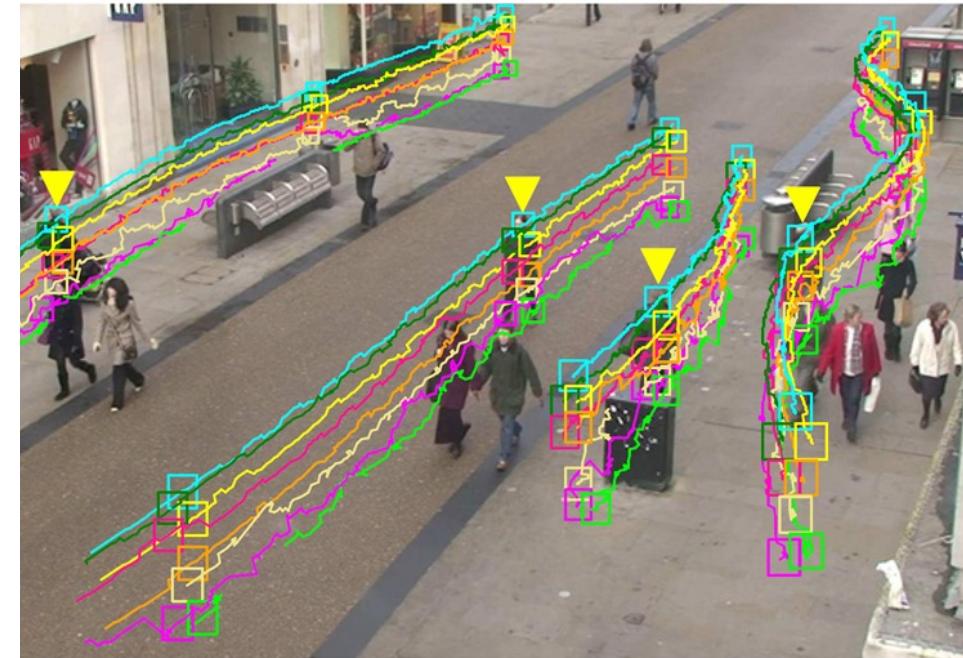
- FlowFormer^[2]



1. Dosovitskiy A. et al. Flownet: Learning optical flow with convolutional networks //Proceedings of the IEEE international conference on computer vision. – 2015. – С. 2758-2766.

2. Huang Z. et al. FlowFormer: A Transformer Architecture for Optical Flow //arXiv preprint arXiv:2203.16194. – 2022.

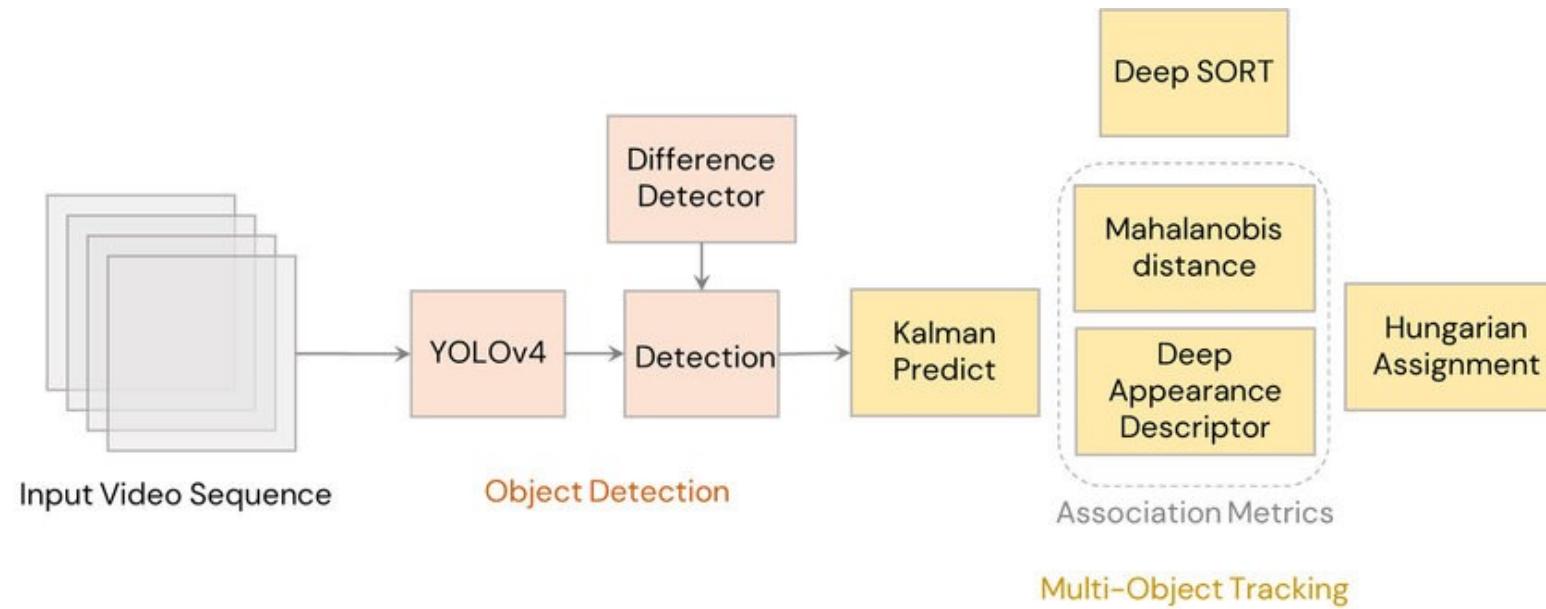
- Отслеживание объектов предназначено для обнаружения целевого объекта в видеопоследовательности с учетом его местоположения в первом кадре.
- Дано множество баундинг боксов и уникальных индексов объектов, надо отследить их перемещение на следующих кадрах.
- Обычно два шага:
 - Детекция объектов.
 - Предсказание движения.
- Нужно, так как:
 - Иногда объект может не найтись методами детекции.
 - Присвоение идентификатора.
 - Предсказания в реальном времени.



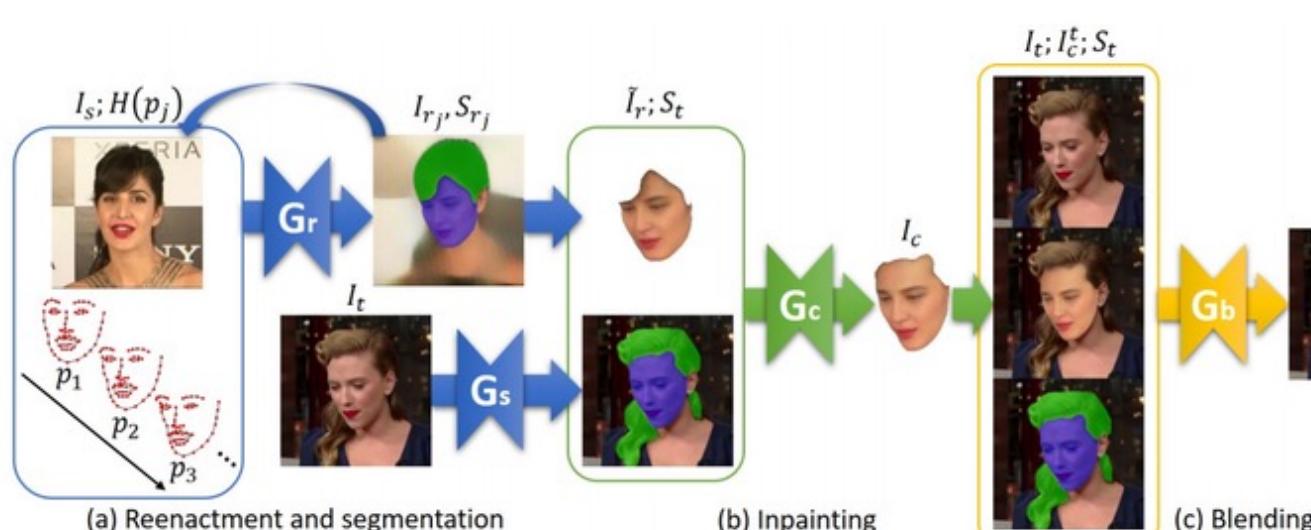
Метод трекинга объектов в 4 шага:

1. Детекция – детекция объекта в кадре с помощью любых методов детекции (YOLO, Fast R-CNN и тд).
2. Вычисление – определяем как изменились позиции объектов, которые находились в баундинг боксах с помощью фильтра Калмана.
3. Соответствие данных – подсчет IoU между полученными ранее и новыми позициями объектов и построение соответствий между старыми и новыми баундинг боксами с помощью венгерского алгоритма.
4. Создание и удаление сущностей – объекты появляются в кадре или покидают его.
 - много изменений идентификаторов.
 - ошибки при перекрытии.

- Сопоставляет баундинг боксы не только по перемещению, но и по виду.
- Дополнительное обучение CNN для описания объектов.



- Синтез голоса или лица человека и замена их на изображении или в видео.
- Шаги:
 - Сегментация лица.
 - Реконструкция лица.
 - Перенос сгенерированного лица.
 - Вписывание лица.



- Видео можно получать из разных источников, нужный результат обычно определяет способ получения картинки.
- Обработка видео требует много вычислений.
- Для определения, куда переместился объект, используют оптический поток, а также трекинг объектов.



УНИВЕРСИТЕТ ИТМО

Спасибо за внимание!

ОБРАЗОВАТЕЛЬНЫЕ ПРОГРАММЫ В ОБЛАСТИ
ТЕХНОЛОГИЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА