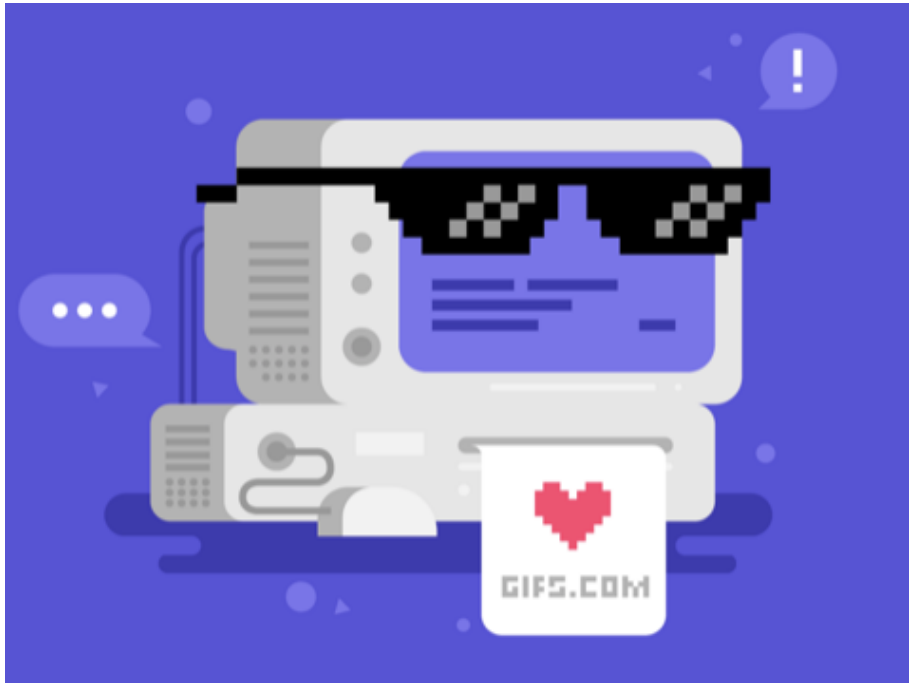
Сверточные нейронные сети для различных задач CV

Обзор задач компьютерного зрения и как их примерно решают

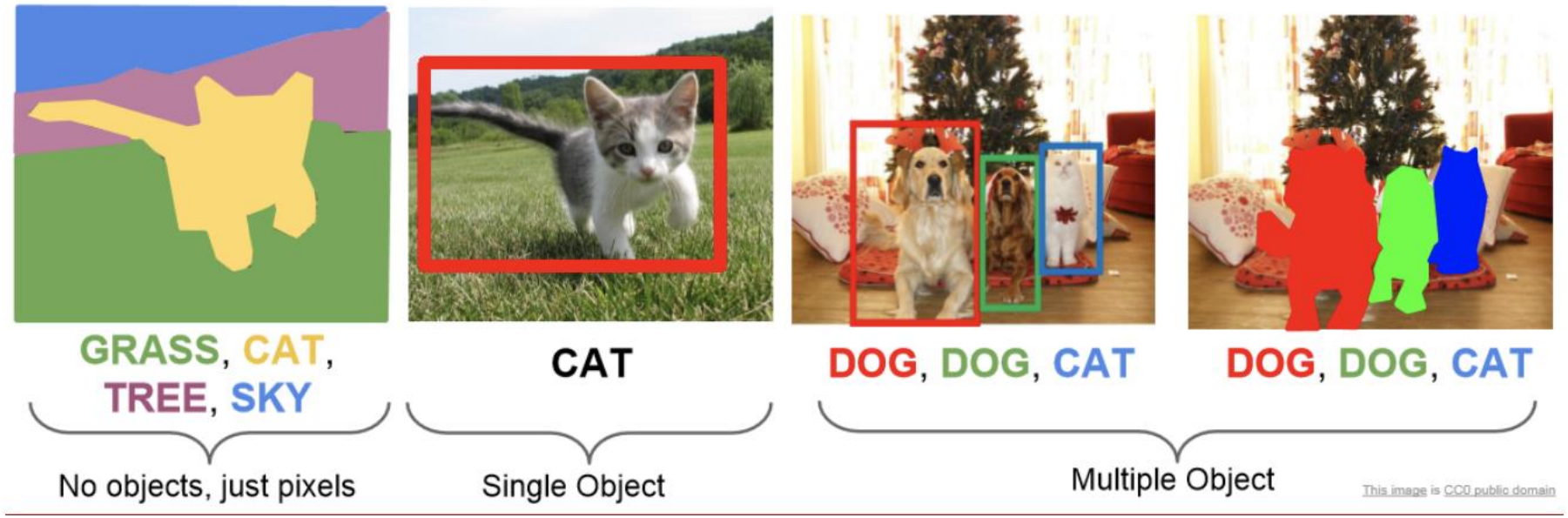
17.11.2020 Сибагатова Софья, Барановская Дарья

Что такое Computer Vision?

- это область компьютерных наук, включающая в себя теорию и технологии создания машин, способных анализировать изображения или видео
- это автоматическое извлечение информации из изображений (начиная от 3D-объектов, положения камеры, заканчивая группировкой и поиском содержимого изображений)
- цель - понять содержание цифровых изображений (обычно включает в себя разработку методов, которые пытаются воспроизвести возможности человеческого зрения)



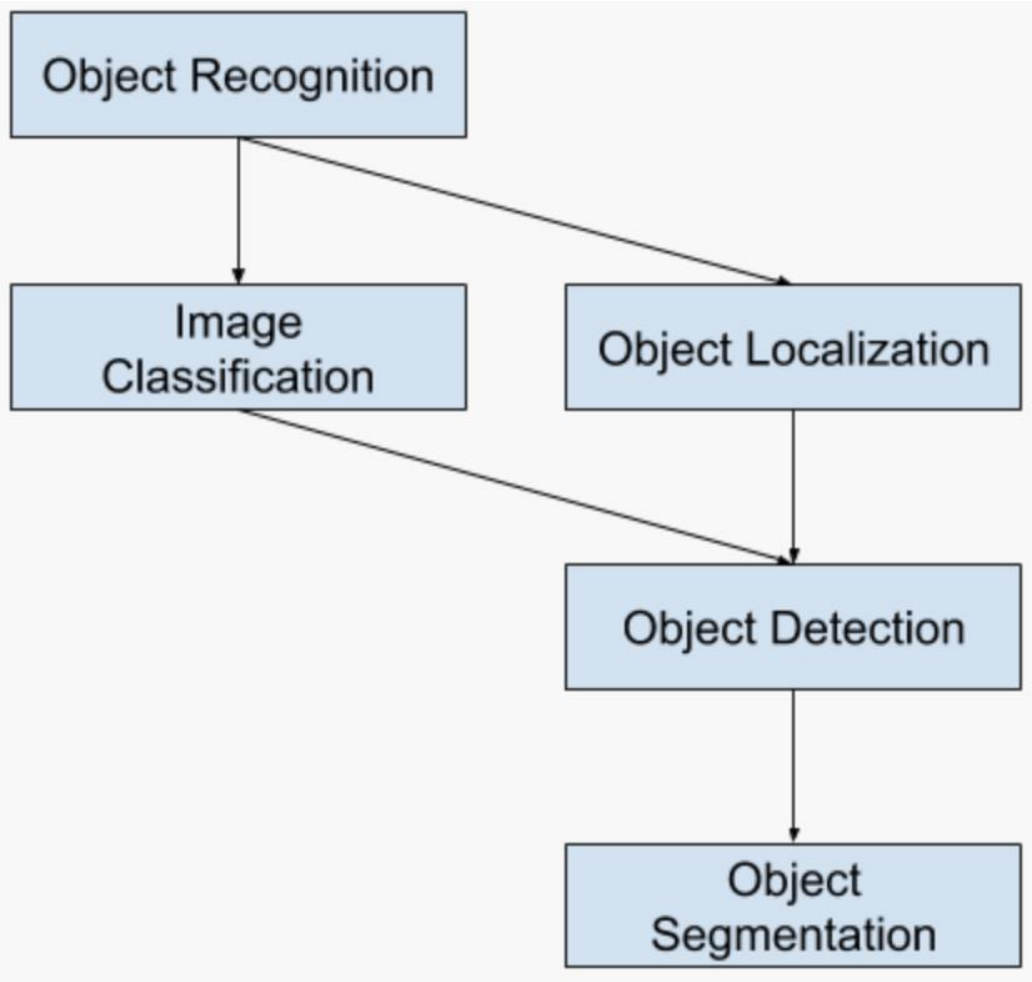
Задачи CV



Semantic Segmentation	Classification + Localization	Object Detection	Instance Segmentation
Отнесение каждого пикселя к соответствующему ему классу	Отнесение картинки к одной из фиксированных категорий и определение региона, где расположен этот объект	Распознавание нескольки х объектов вместе с соответствующими им регионами. Ни количество объектов, ни их размер не известны заранее	Отнесение пикселей к объекту класса, к которому он принадлежит

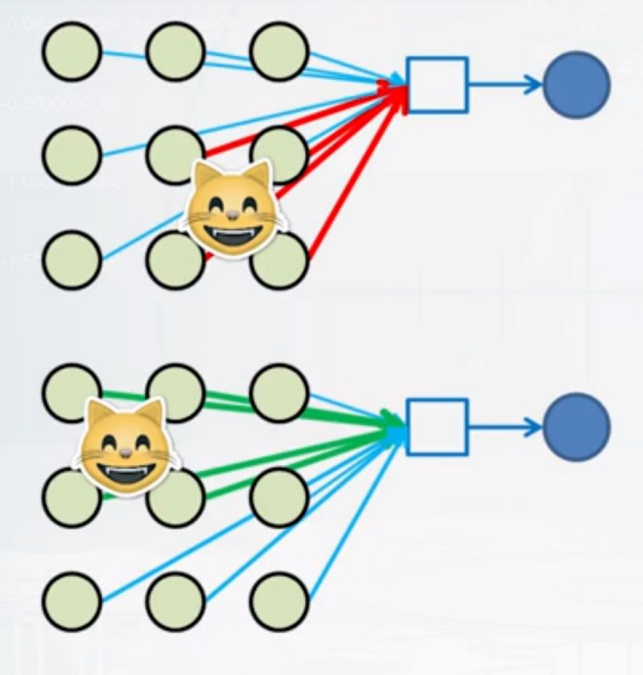
Как далее будет построен доклад

1. CNN
2. Классификация + Локализация
3. Сегментация
4. Детекция
5. Перенос стиля



Почему не MLP?

1. Количество параметров обучения: для изображения размером 224 x 224 пикселей с 3 цветными каналами требуется около 150000 весов, которые необходимо обучить
2. MLP по-разному реагирует на изображение и его сдвинутую версию, он не инвариантен к сдвигу: детекция кота на изображении.



На первом изображении красные веса будут изменяться для того, чтобы лучше детектировать кота, а на втором - зеленые

↓

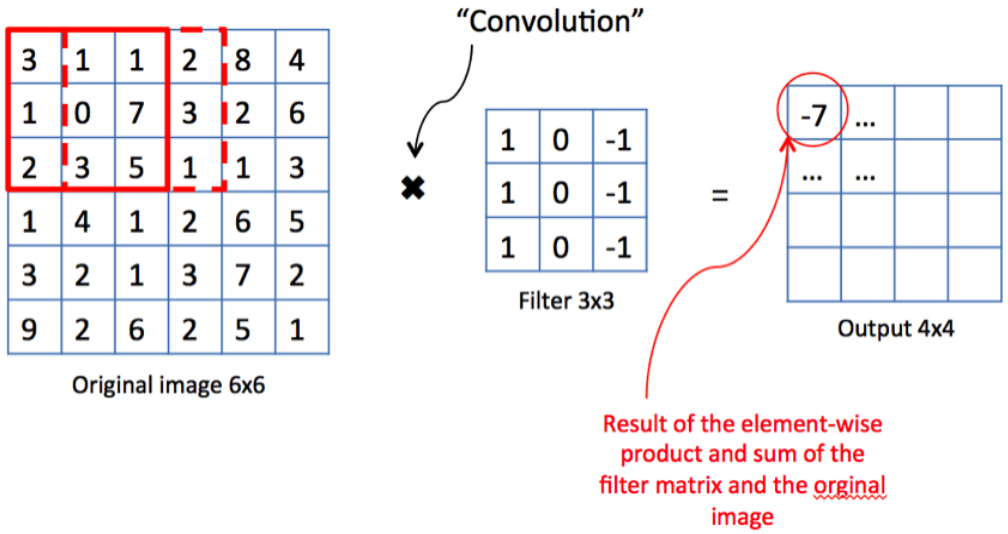
Мы изучаем одни и те же "кошачьи признаки" в разных частях изображения

↓

Теряется пространственная информация

Сверточный слой (convolutional layer)

Свертка (**convolution**) - это скалярное произведние **ядра** (**kernel**) или, по-другому, **фильтра** (**filter**), и фрагмента изображения такого же размера (**local receptive field**). Данное скалярное произведение считается для всех подматриц размера ядра с некоторым **шагом** (**stride**). Также, к исходному изображению можно добавить **рамку** (**padding**).



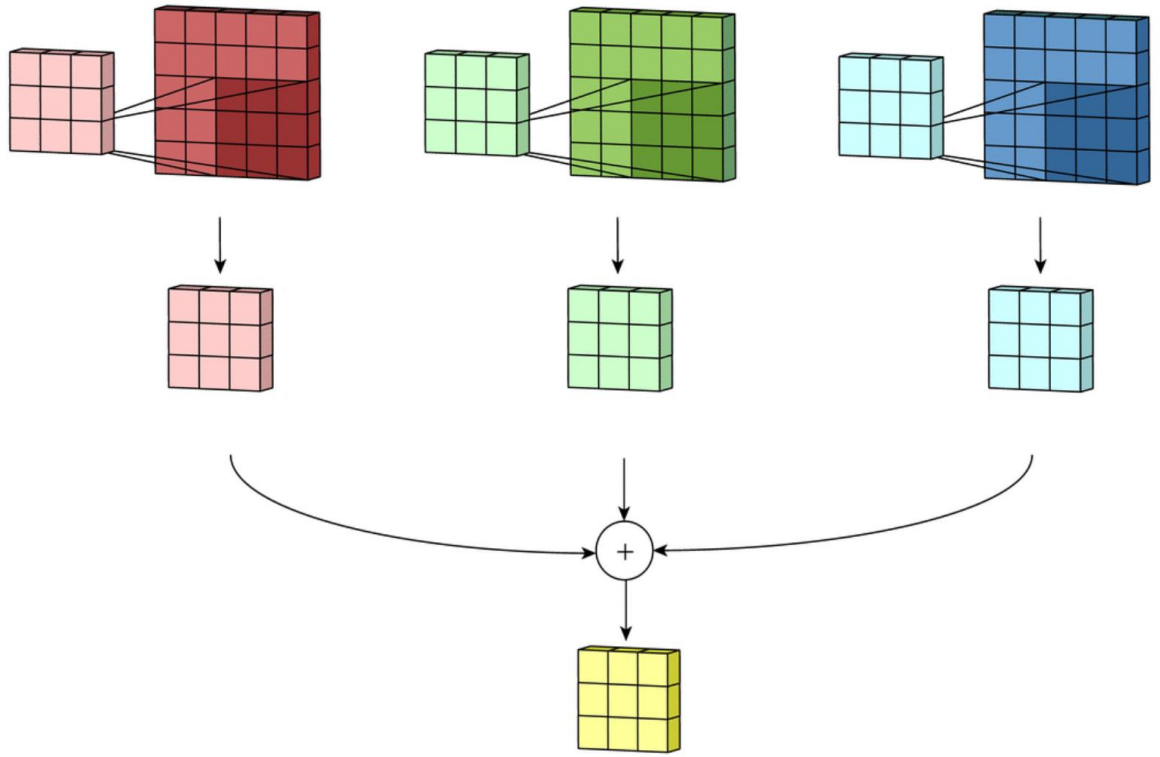
W - размер входного изображения
F - размер ядра
P - ширина рамки
S - длина шага

Размер выхода: $\frac{W - F + 2P}{S} + 1$

- Свертка имеет сходство с корреляцией
- Convolution is translation equivariant

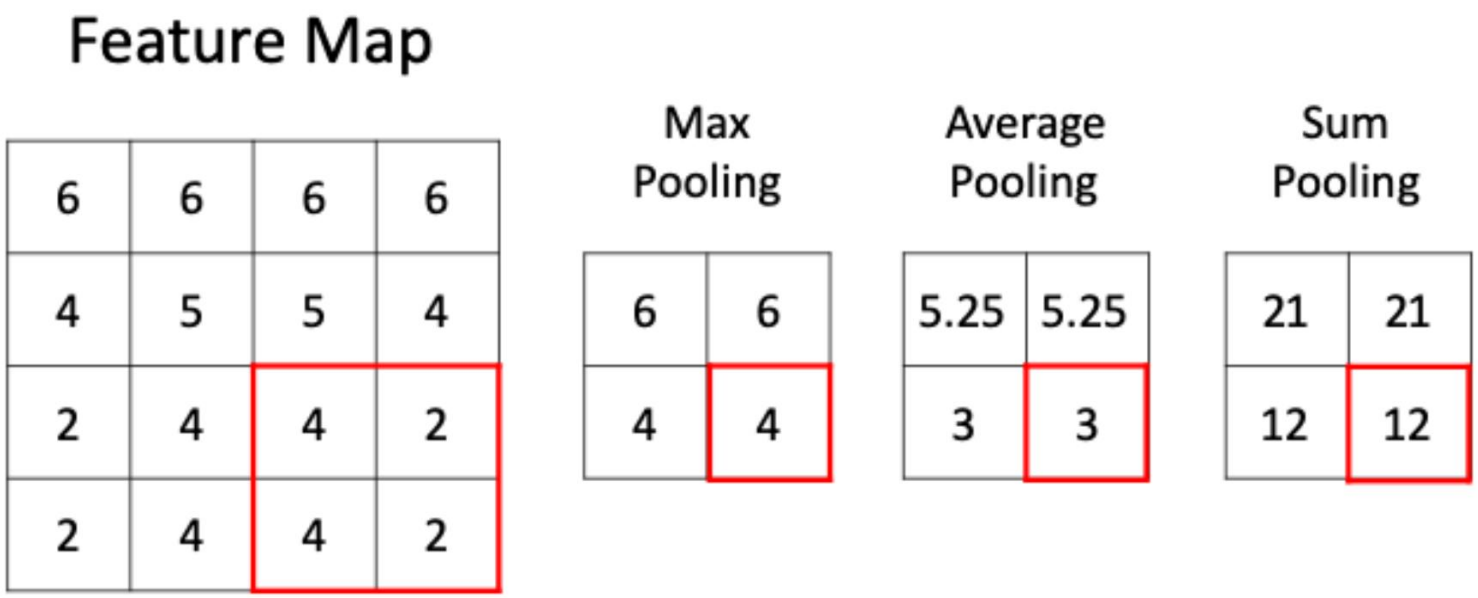
Сверточный слой (convolutional layer)

Если входное изображение имеет несколько **каналов (chan nels),** то для каждого отдельного канала необходимо свое ядро и, после применения соответствующих ядер, полученные матрицы суммируются для формирования единого выхода.



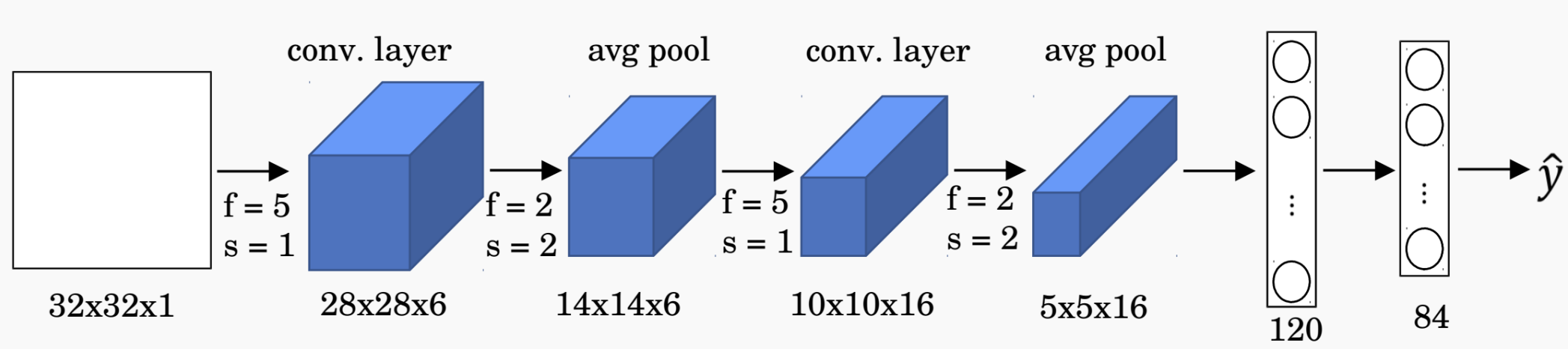
Субдискретизирующий слой (pooling)

Данный слой работает так же, как сверточный, но не использует ядро, а вместо этого считает максимум, среднее или сумму подматриц изображения с некоторым шагом.



LeNet-5

Разработана в 1998 году Янном ЛеКуном для распознавания рукописных цифр в наборе данных MNIST. Этот набор данных содержит 10 кластеров рукописных цифр от 0 до 9 (60000 изображений для обучения и 10000 - для теста).



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

TABLE I

EACH COLUMN INDICATES WHICH FEATURE MAP IN S2 ARE COMBINED BY THE UNITS IN A PARTICULAR FEATURE MAP OF C3.

Error rate: 0.95%
Trainable parameters: 60.850
Connections: 340.918

LeNet-5

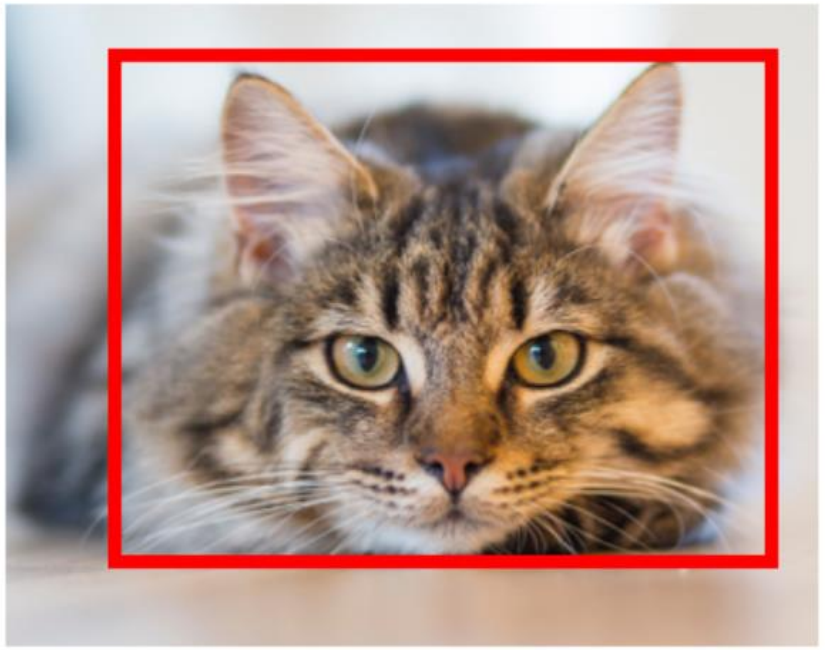
Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	32x32	-	-	-
1	Convolution	6	28x28	5x5	1	tanh
2	Average Pooling	6	14x14	2x2	2	tanh
3	Convolution	16	10x10	5x5	1	tanh
4	Average Pooling	16	5x5	2x2	2	tanh
5	Convolution	120	1x1	5x5	1	tanh
6	FC	-	84	-	-	tanh
Output	FC	-	10	-	-	softmax

Функция активации выходного слоя: **Euclidean Radial Basis Function**

$$y_i = \sum_j (x_j - w_{ij})^2$$

Классификация & локализация

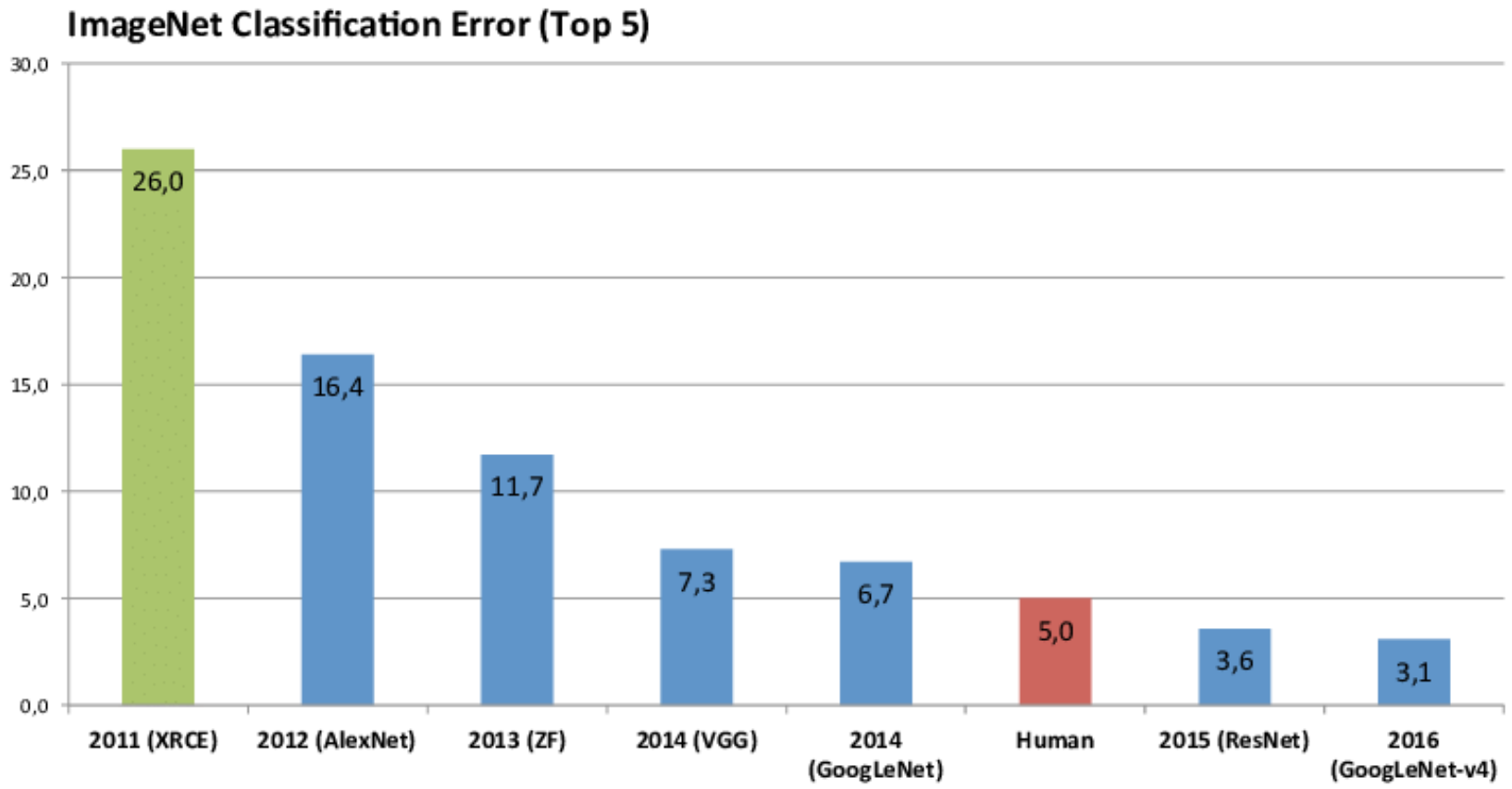
Classification
+ Localization



CAT

ImageNet

Набор данных ImageNet имеет более 1 миллиона изображений, разделённых на 1000 классов.



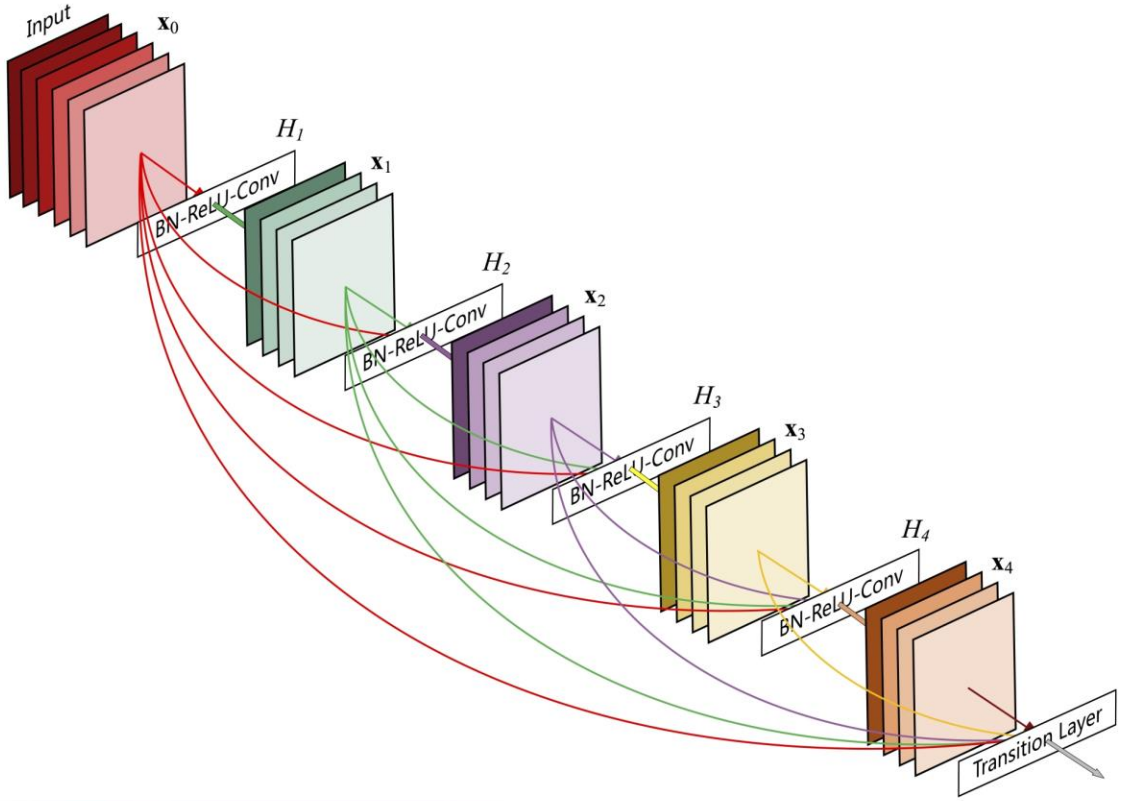
CNN для классификации

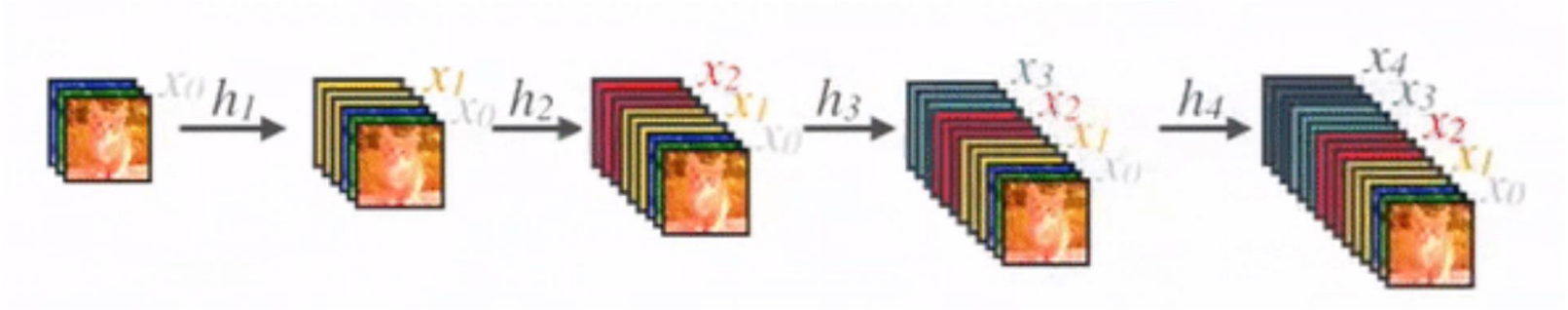
CNN	Что использует?	параметры	GPU	время
AlexNet (2012)	Свертки 11x11, 5x5, 3x3, Max Pooling с перекрыванием, DropOut, аугментация, функция активации ReLU, SGD с импульсом	60.000.000	2	6 дней
VGG (2015)	AlexNet, но меньшие свертки (3x3, а также 1x1)	138.000.000	4	2-3 недели
Inception V3	Похожа на AlexNet, но использует batch normalization, RMSProp, а также блоки Inception (применение сверток разного размера параллельно + свертки 1x1)	25.000.000	8	2 недели
ResNet (2015)	Batch Normalization, Max и Average Pooling, свертки 7x7, 3x3, быстрые соединения (residual connections)	60.000.000	8	2-3 недели

DenseNet

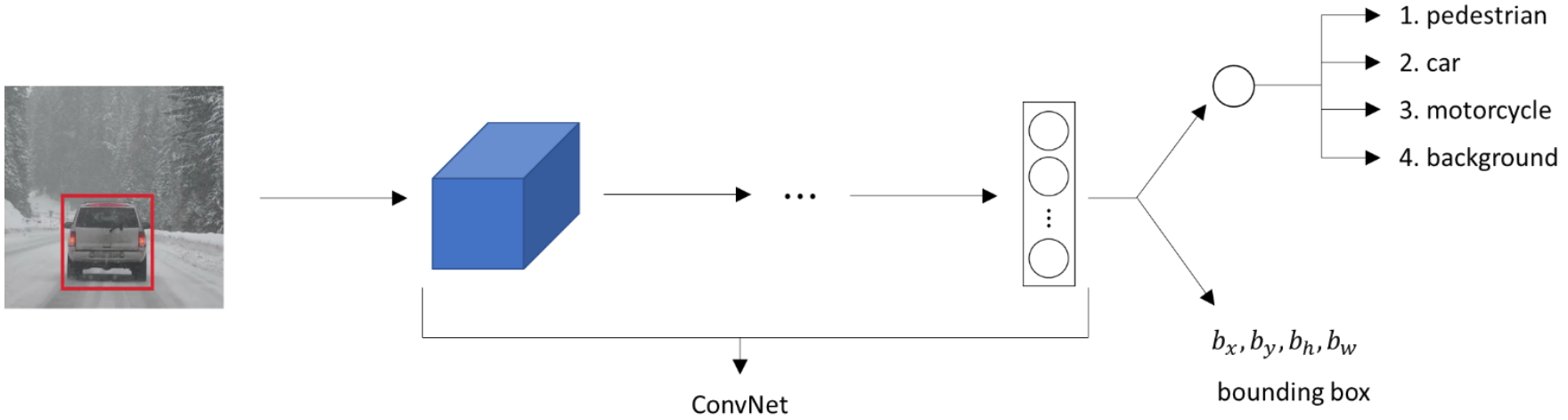
Каждый слой принимает все предыдущие карты признаков в качестве входных данных.

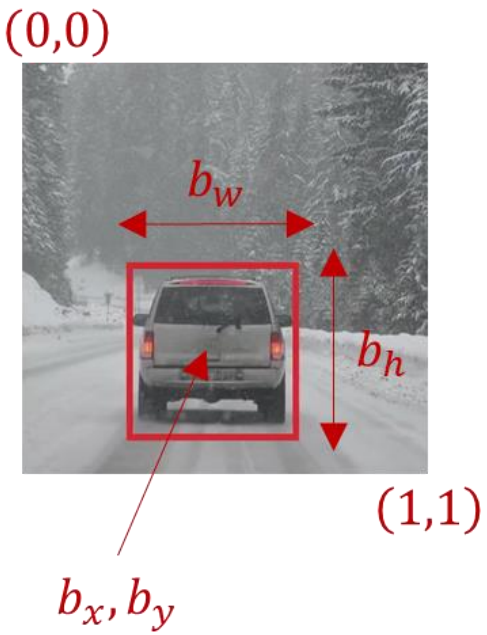
Dense-блок (плотный блок) - блок, который соединяет каждый слой с каждым другим слоем. Признаки прежде чем они будут переданы в следующий слой, конкатенируются в единый тензор.





Локализация





$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

p_c - индикатор того, есть ли на изображении объект (p_c = 0 для фона)

c_i - индикаторы принадлежности классу i (есть, только если p_c = 1)

Функция потерь: MSE

Сегментация

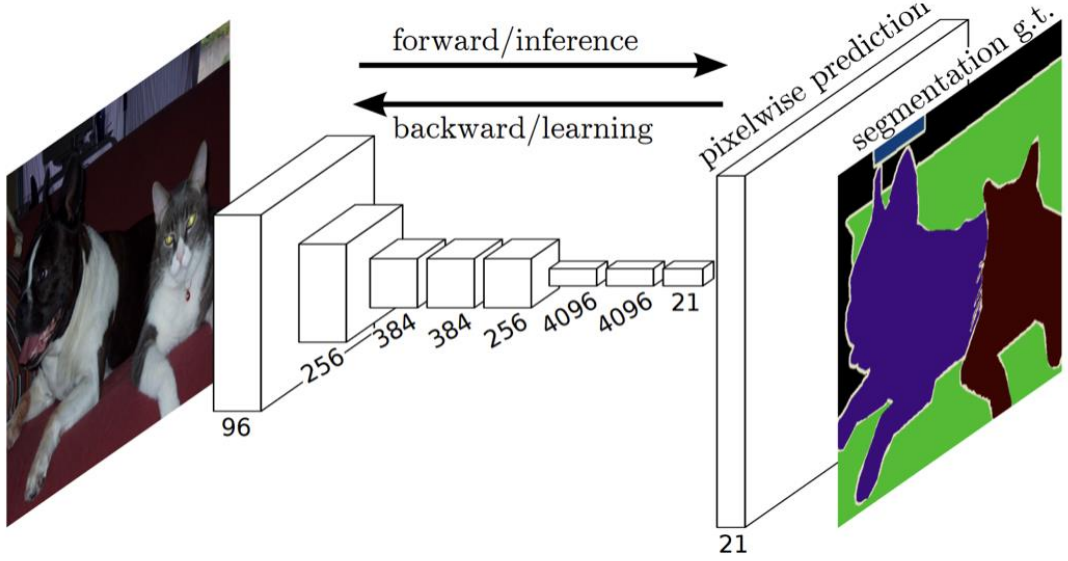
Semantic segmentation

- Задача: разбить картинку на классы
- Идея: попиксельная нейросетевая классификация.
- Примеры использования: размытие фона при портретной съемке
- Способы решения

Semantic Segmentation

1. Efficient Graph-Based Image Segmentation
2. Region based Semantic Segmentation: делается в два этапа - выделение региона, в котором лежит некоторый объект(задача распознания объекта) и отнесение пикселей к содержащему его региону с наивысшим значением score функции
3. Fully Convolutional Network-Based
4. CNN encoder-decoder network

Fully-Convolutional Network



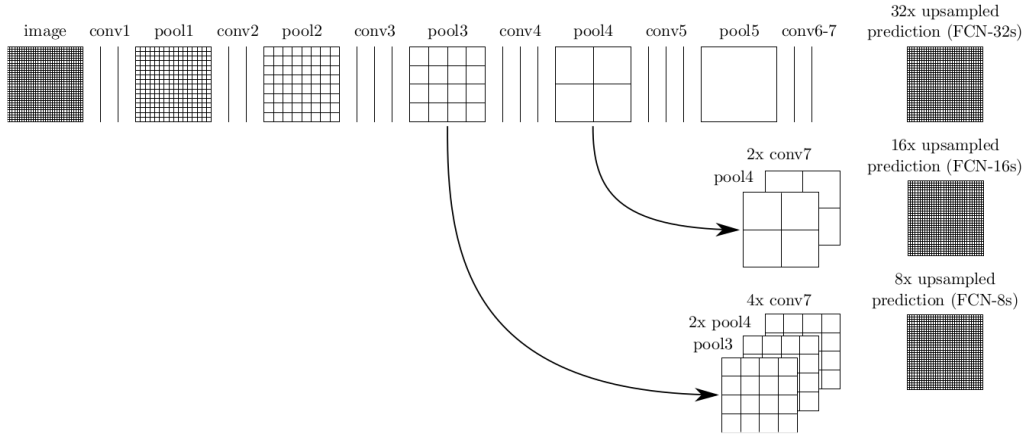
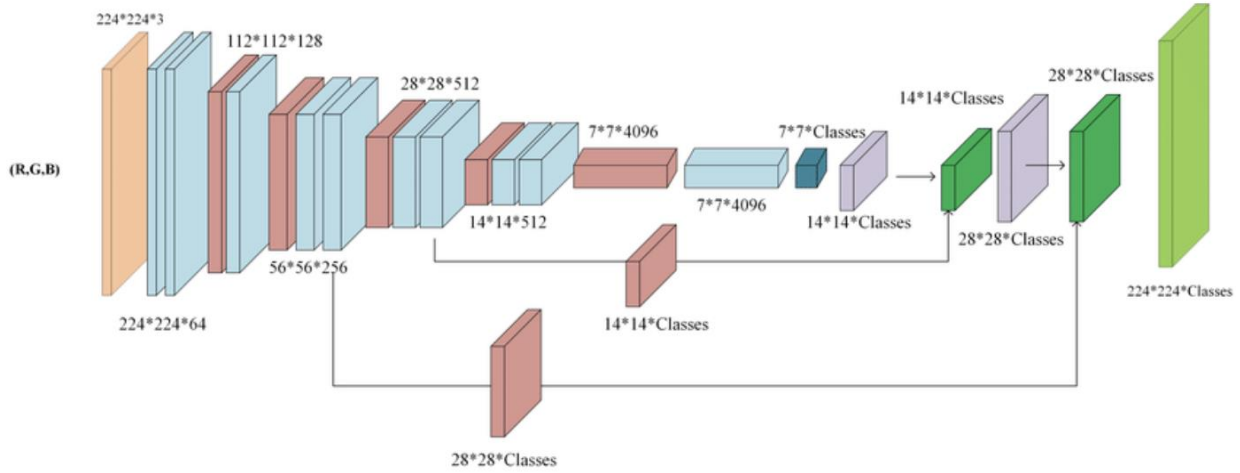


Figure 2 : FCN architecture (from FCN paper)

- Основана на VGG16
- Downsample then upsample
- no fully_connected layers for classification
- softmax probability function
- Include skip connections



Fully convolutional neural network architecture (FCN-8).

FCN Results

- pixel accuracy: $\sum_i n_{ii} / \sum_i t_i$
- mean accuraccy: $(1/n_{\text{cl}}) \sum_i n_{ii}/t_i$
- mean IU: $(1/n_{\text{cl}}) \sum_i n_{ii} / \left(t_i + \sum_j n_{ji} - n_{ii}\right)$
- frequency weighted IU: $\left(\sum_k t_k\right)^{-1} \sum_i t_i n_{ii} / \left(t_i + \sum_j n_{ji} - n_{ii}\right)$

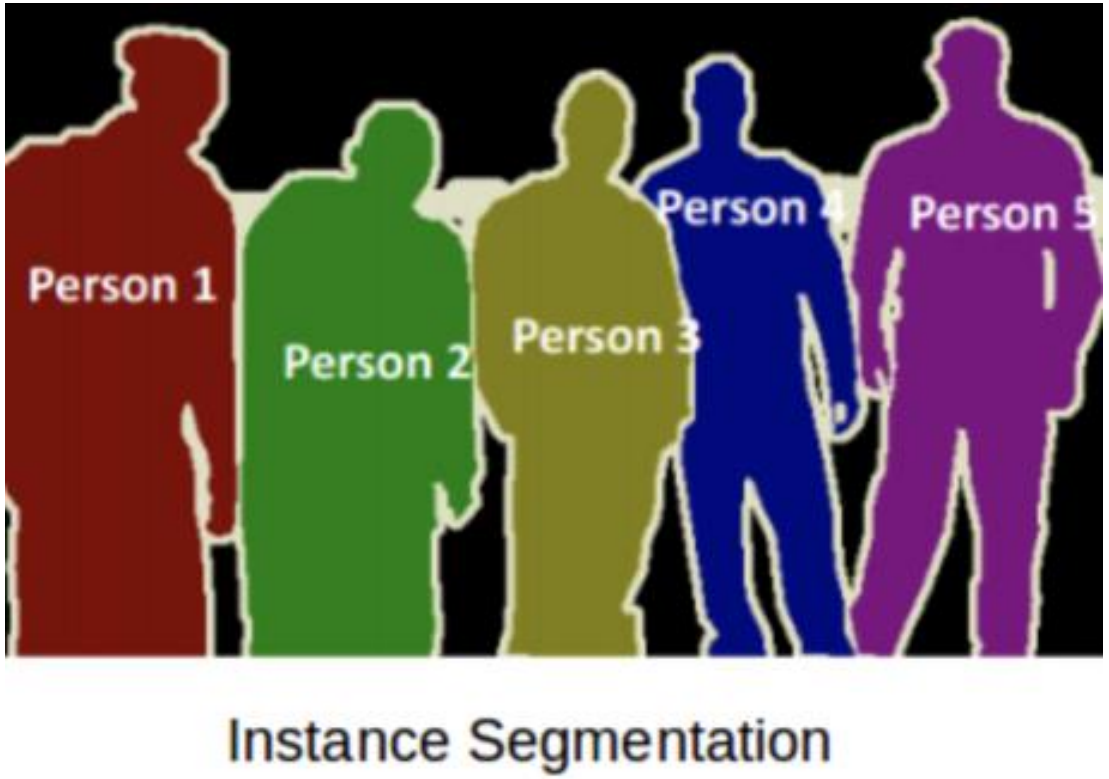
	mean IU VOC2011 test	mean IU VOC2012 test	inference time
R-CNN [10]	47.9	-	-
SDS [15]	52.6	51.6	$\sim$ 50 s
FCN-8s	**62.7**	**62.2**	$\sim$ **175 ms**

- IU = intersection over union
- n_{ij} - число пикселей класса j, отнесенных к классу i
- n_cl - число классов
- t_i - число пикселей класса i

Сегментация

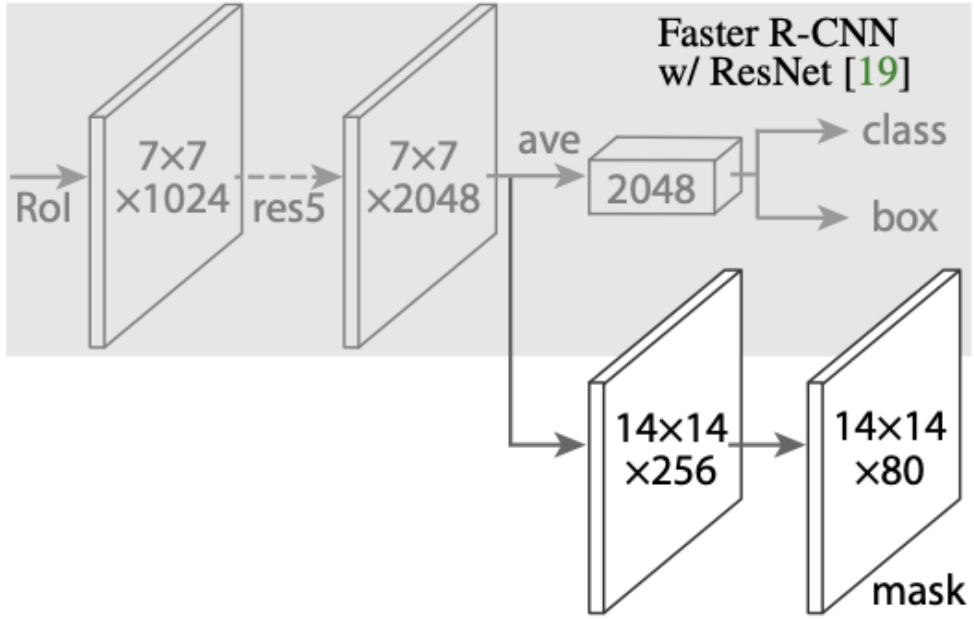
Instance segmentation

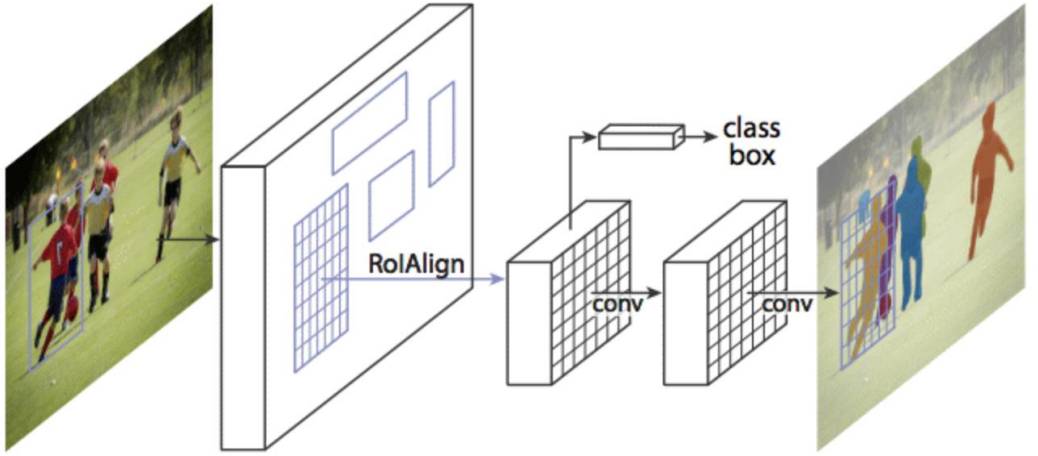
- Задача: разбить картинку на классы, также выделить отделить разные объекты одного и того же класса друг от друга
- Не требуется относить каждый пиксель к какому-либо классу
- Основано на аналогичных RCNN алгоритмах, однако требуется также определить и форму самого объекта, отделив его от фона



Mask R-CNN

- Развивает архитектуру Faster R-CNN, добавляя ветку, которая предсказывает положение маски, покрывающей найденный объект
- Маска - матрица из 0 и 1, где 1 - принадлежит заданному классу, 0 - нет
- Условное разделение архитектуры: объединение частей, отвечающих за предсказание охватывающей рамки, классификацию объекта и определение его маски
- Для каждого ROI и каждого класса своя маска. loss L_{mask} определены только на положительных RoIs

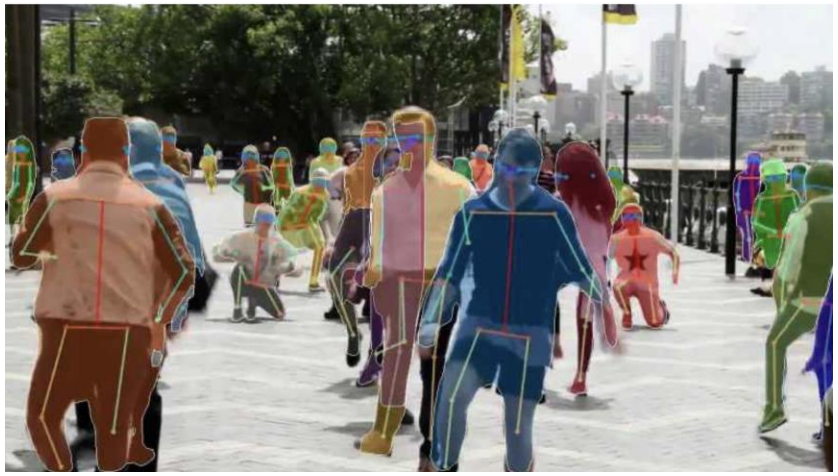


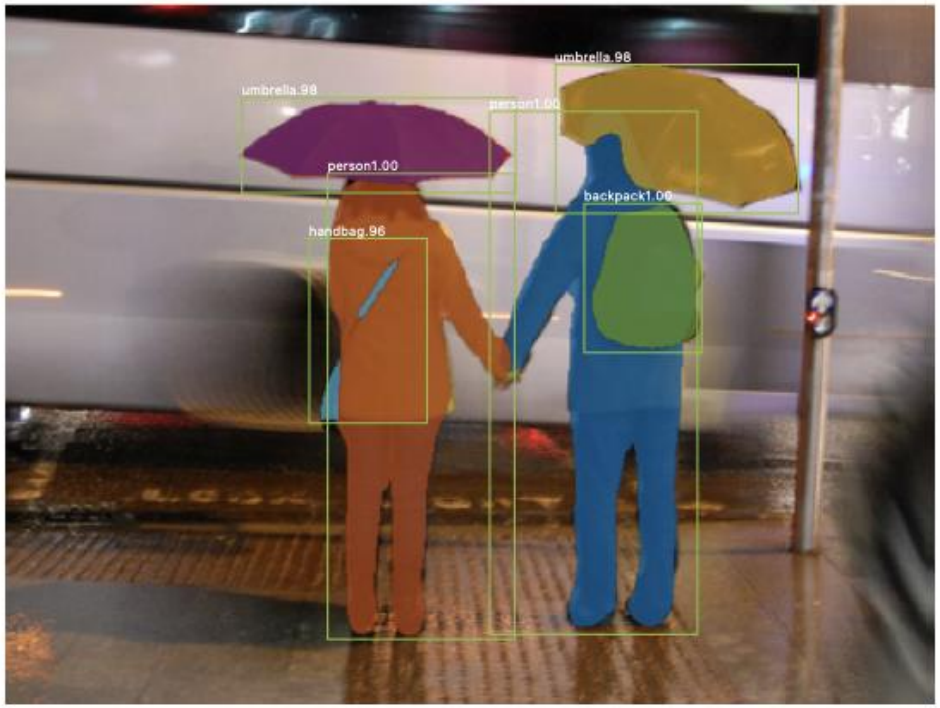


Результаты Mask RCNN

$$L = L_{cls} + L_{box} + L_{mask}$$

- L_cls и L_box из Fast RNN
- Lmask - ошибка по k маскам размера n*n





	training data	AP [val]	AP	AP_{50}	person	rider	car	truck	bus	train	mcycle	bicycle
InstanceCut [23]	fine + coarse	15.8	13.0	27.9	10.0	8.0	23.7	14.0	19.5	15.2	9.3	4.7
DWT [4]	fine	19.8	15.6	30.0	15.1	11.7	32.9	17.1	20.4	15.0	7.9	4.9
SAIS [17]	fine	-	17.4	36.7	14.6	12.9	35.7	16.0	23.2	19.0	10.3	7.8
DIN [3]	fine + coarse	-	20.0	38.8	16.5	16.7	25.7	20.6	30.0	23.4	17.1	10.1
SGN [29]	fine + coarse	29.2	25.0	44.9	21.8	20.1	39.4	24.8	33.2	30.8	17.7	12.4
Mask R-CNN	fine	31.5	26.2	49.9	30.5	23.7	46.9	22.8	32.2	18.6	19.1	16.0

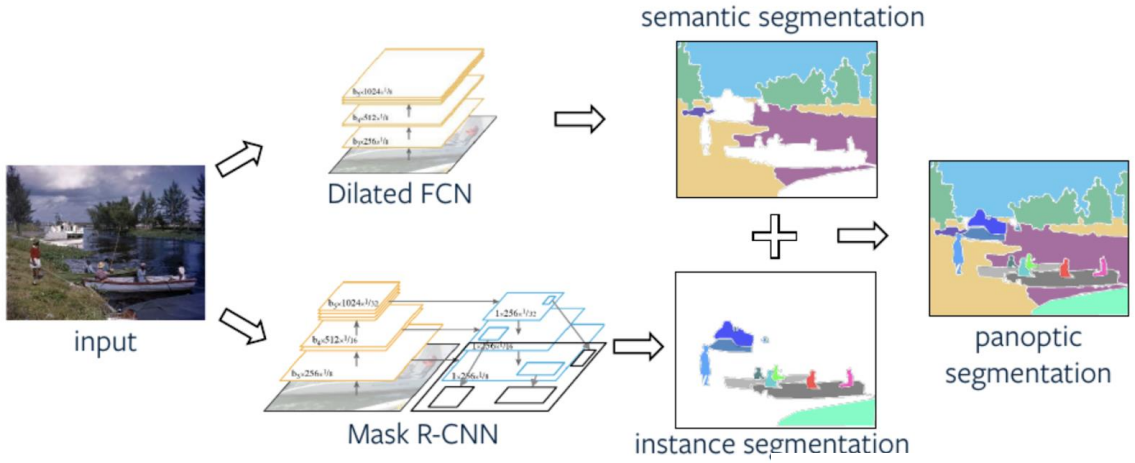
Сегментация

Panoptic segmentation

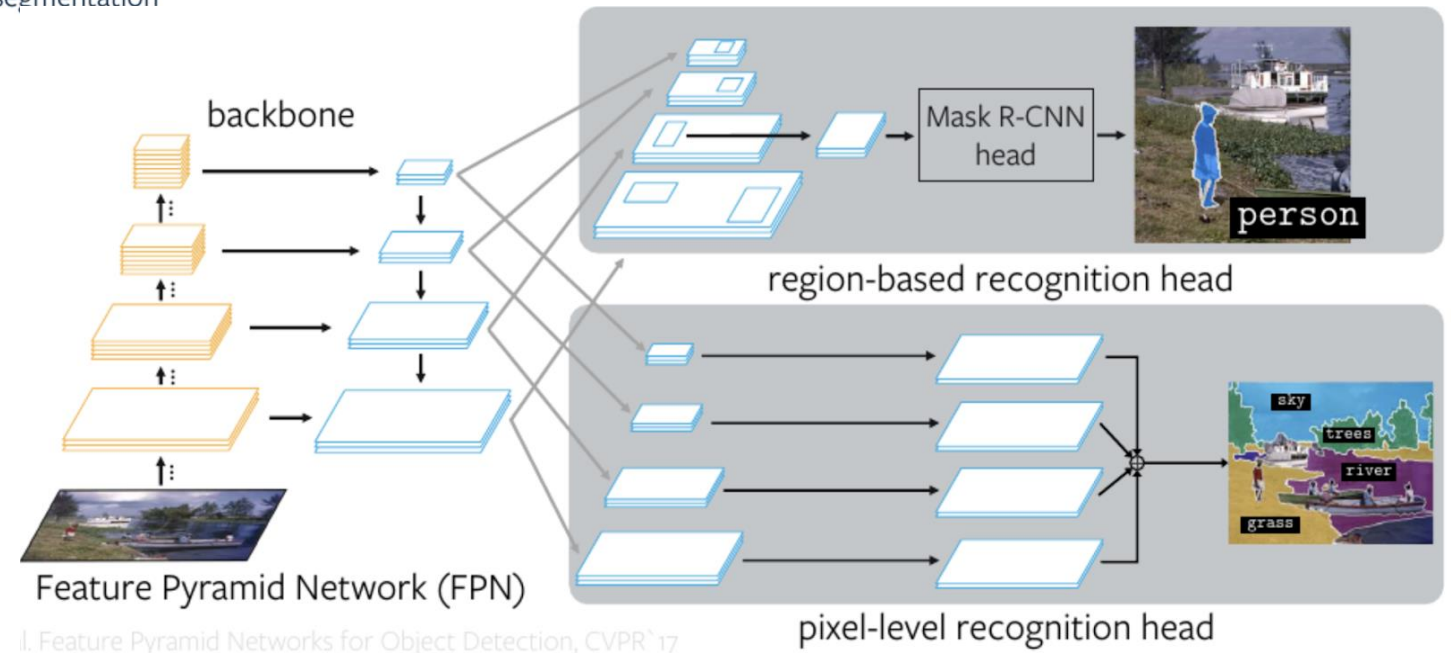
- Задача: совместить Semantic и Instance
- Требуется относить каждый пиксель к какому-либо классу и объекты одного и того же класса к разным
- Совмещение алгоритмов двух предшествующих сегментаций

Left: semantic segmentation, middle: instance segmentation, right: panoptic segmentation

Model for Panoptic Segmentation



Комбинация Mask RCNN и FCN



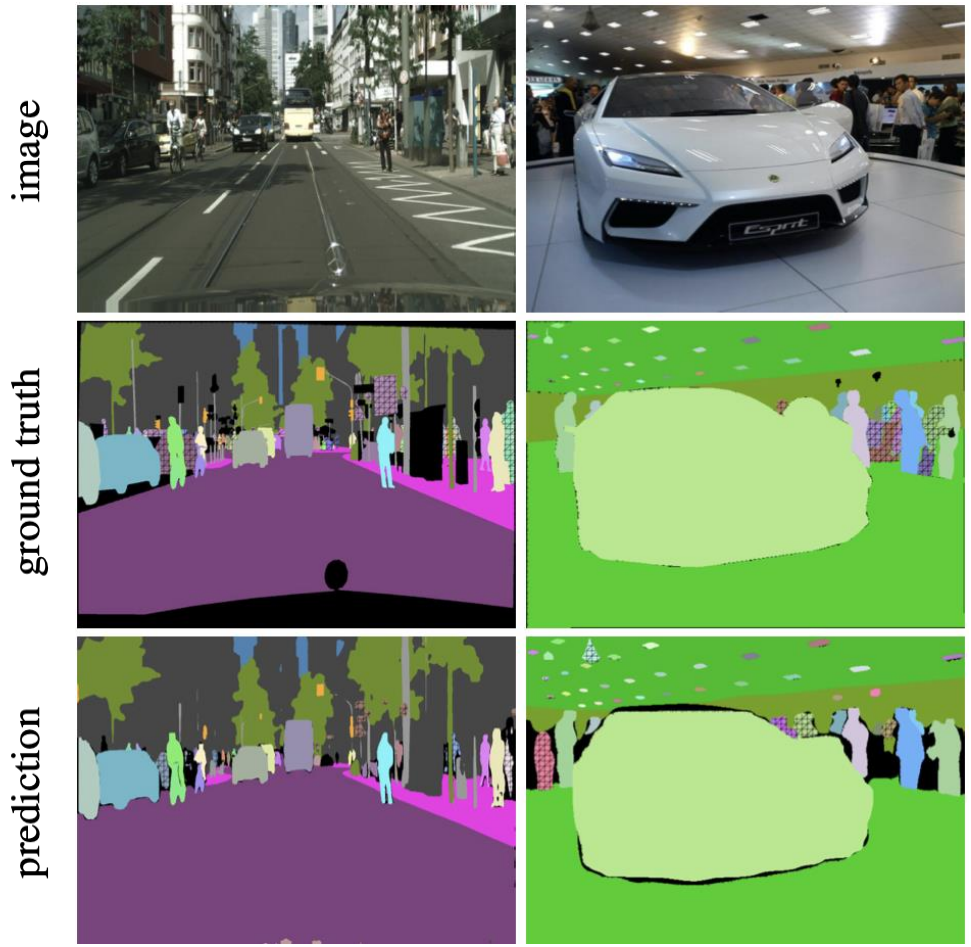
l. Feature Pyramid Networks for Object Detection, CVPR`17

Model for Panoptic Segmentation

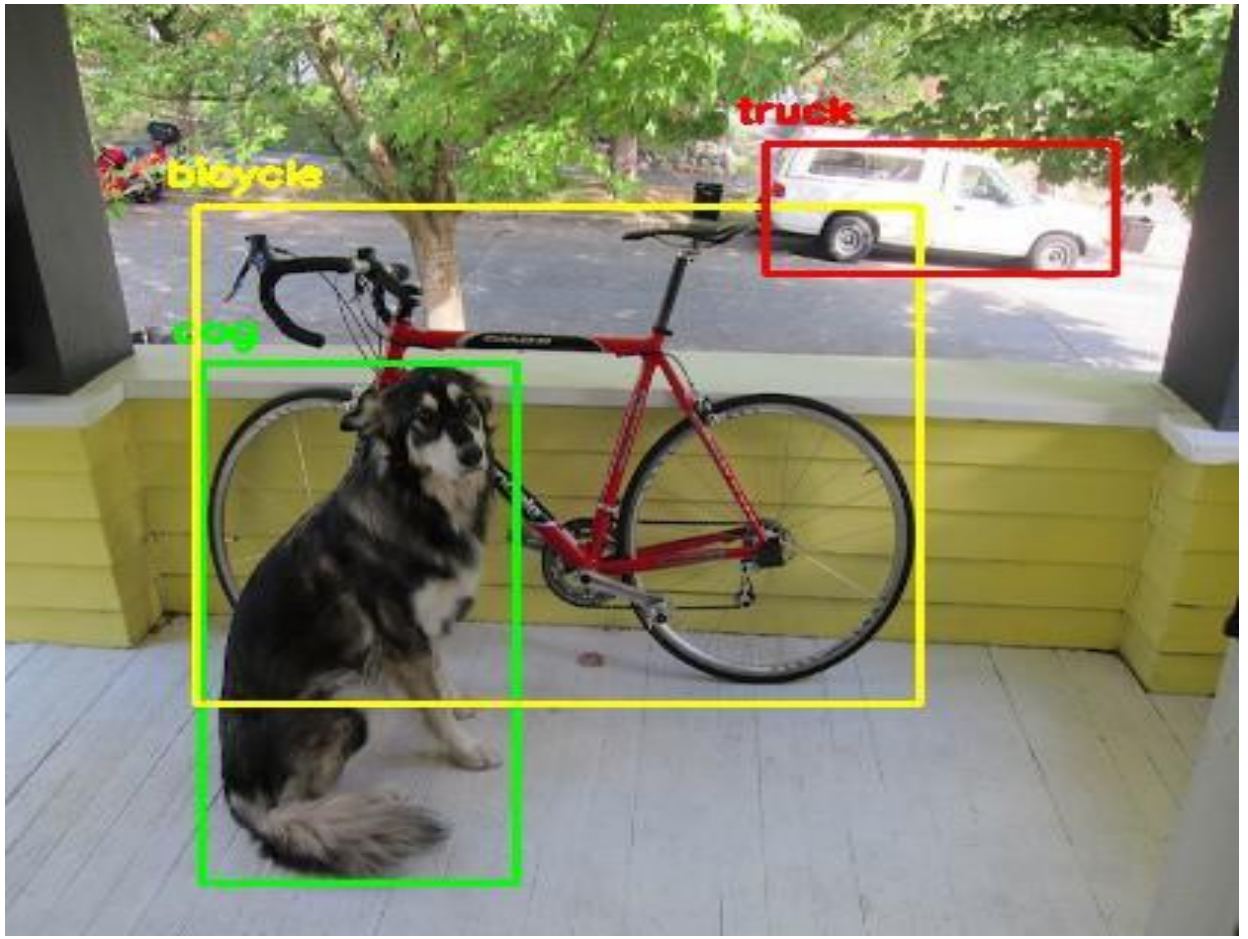
$$\mathrm{PQ} = \frac{\sum_{(p,g)\in TP} \mathrm{IoU}(p,g)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}.$$

$$\mathrm{PQ} = \underbrace{\frac{\sum_{(p,g)\in TP} \mathrm{IoU}(p,g)}{|TP|}}_{\text{segmentation quality (SQ)}} \times \underbrace{\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}_{\text{recognition quality (RQ)}}$$

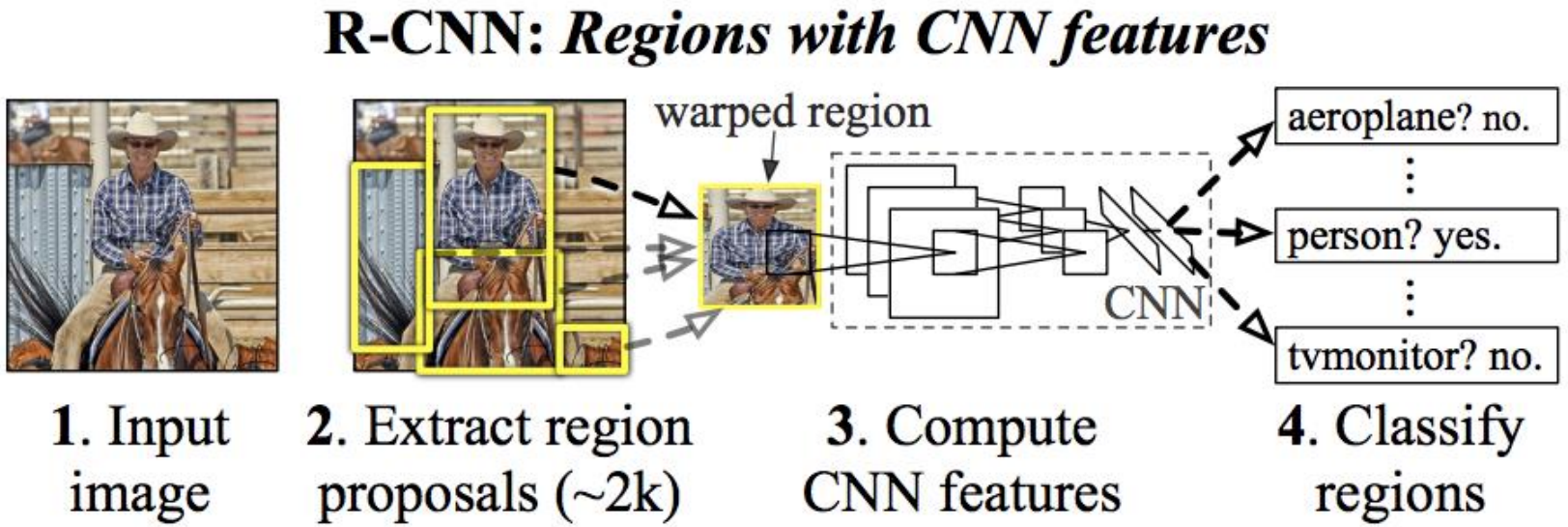
- Instance segmentation: уберем пересечения
- В semantic пересечений нет



Детекция



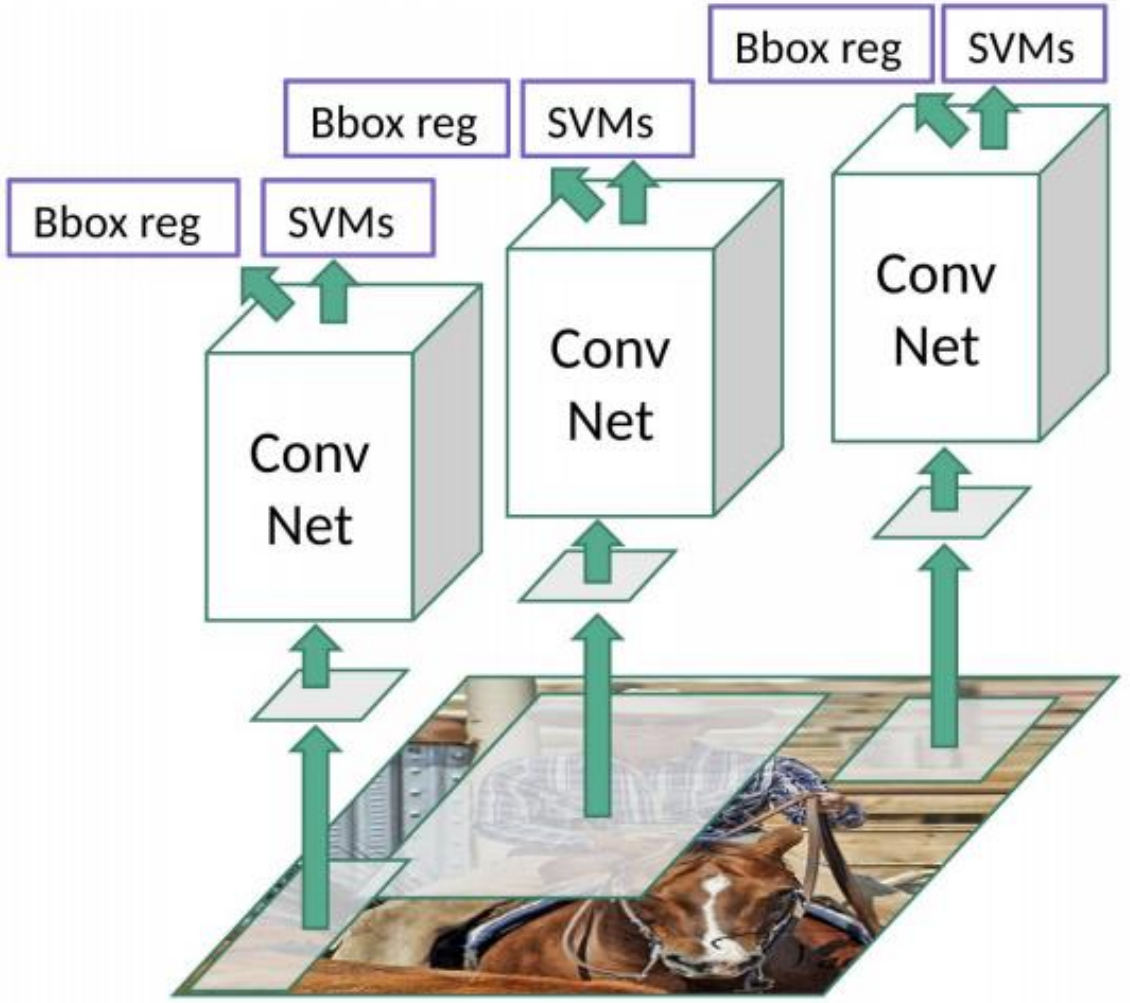
R-CNN



- На изображении выделяются, используя алгоритм **выборочного поиска** (**selective search**), регионы, которые предположительно содержат объект.
- Каждый регион-претендент (а вернее параллелепипед, ограничивающий его) при помощи аффинного преобразования отображается в квадрат 227x227. При этом параллелепипед перед преобразованием расширяется таким образом, чтобы после преобразования получить вокруг региона претендента кайму шириной 16 пикселей.
- Полученный прямоугольник подается на вход сети для классификации (например, AlexNet)

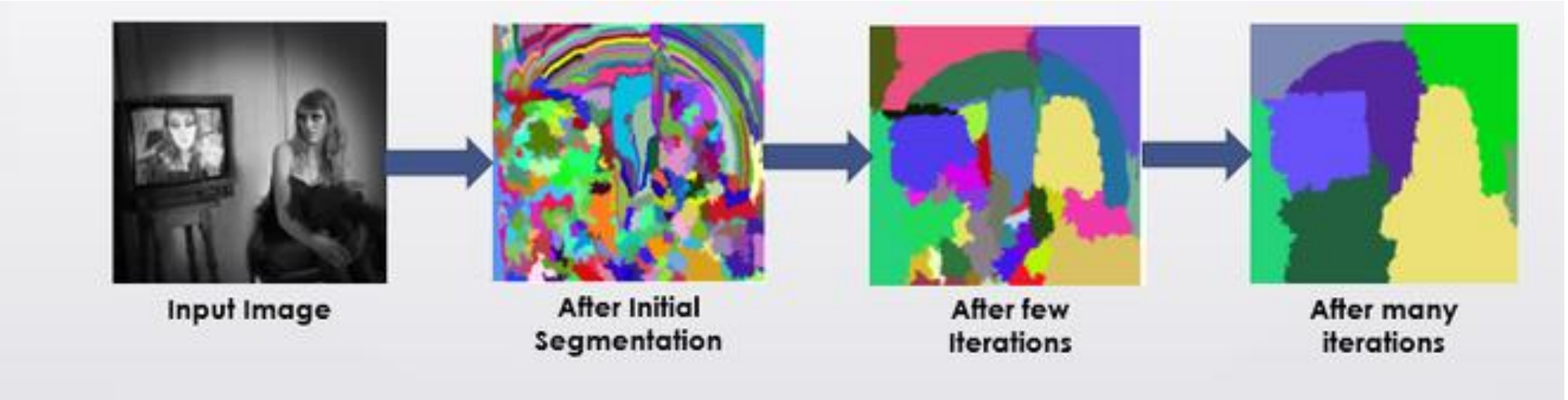
R-CNN

- С предпоследнего слоя выбранной для классификации сети снимается вектор особенностей (признаков) размерности 4096. Этот вектор используется в SVM, натренированной для каждого класса по схеме один против всех, для классификации.
- Тот же вектор особенностей подается в линейный регрессор соответствующий классу объекта для уточнение границ регионов.

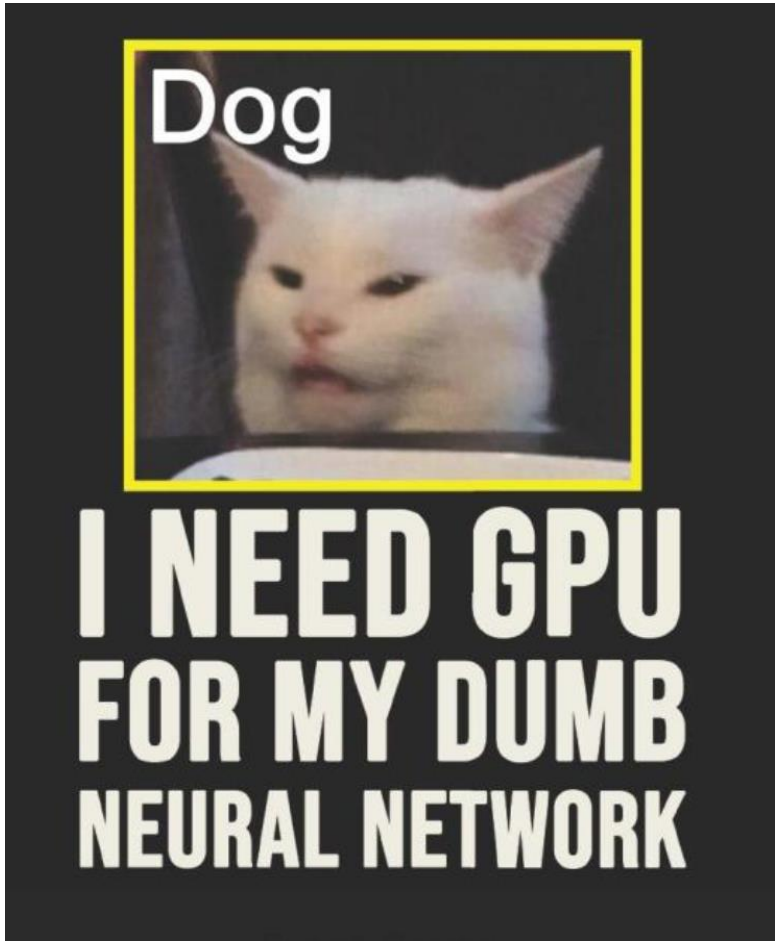


Selective Search (выборочный поиск)

1. Сгенерируйте начальную суб-сегментацию входного изображения, используя метод, описанный в статье «Efficient Graph-Based Image Segmentation».
2. Рекурсивно объединяйте более мелкие похожие области в более крупные с помощью жадного алгоритма.
3. Используйте полученные области для создания местоположений потенциальных объектов

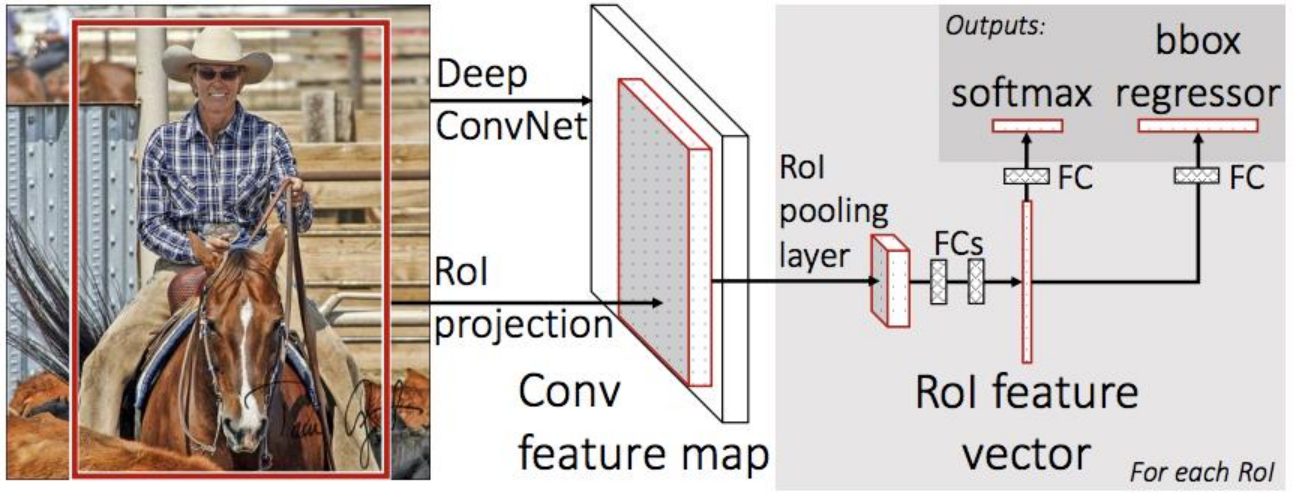


Проблемы R-CNN



- На обучение по-прежнему уходит огромное количество времени, поскольку придется классифицировать 2000 предложений регионов для каждого изображения.
- Применять R-CNN в реальном времени или для работы с видео невозможно (время работы составляет около 47 секунд на тестовых изображениях если брать сеть VGG16).
- Selective search не является алгоритмом машинного обучения (то есть, на этапе выборочного поиска обучение не происходит). Это может привести к появлению плохих предложений регионов-кандидатов.

Fast R-CNN



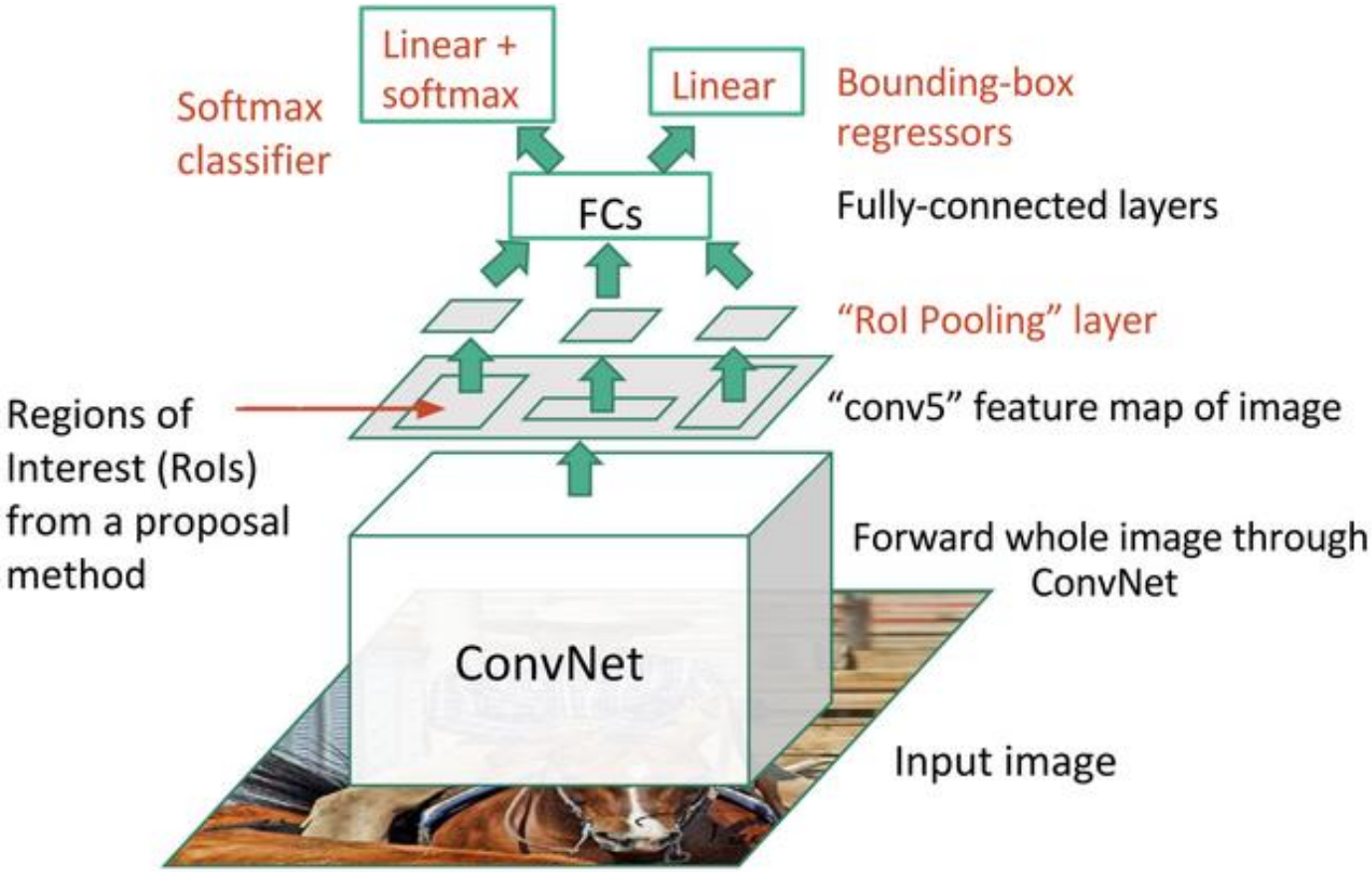
- Полное изображение подается на вход сети для классификации (но вместо последнего Max Pooling слоя используется слой нового типа RoI (Region-of-Interest) Pooling), и обрабатывается с помощью выборочного поиска.
- В итоге, до слоя RoI Pooling имеем карту признаков и регионы потенциальных объектов. Координаты регионов потенциальных объектов преобразуются в координаты на карте признаков.
- Полученная карта признаков с регионами передается слою RoI. Здесь на каждый регион накладывается сетка HxW с наперед заданными размерами (например, W = H = 7 для VGG16).

Fast R-CNN

- После RoI Pooling данные через два полносвязных слоя подаются параллельно на:

а. softmax слой для оценки принадлежности данного претендента одному из классов объектов

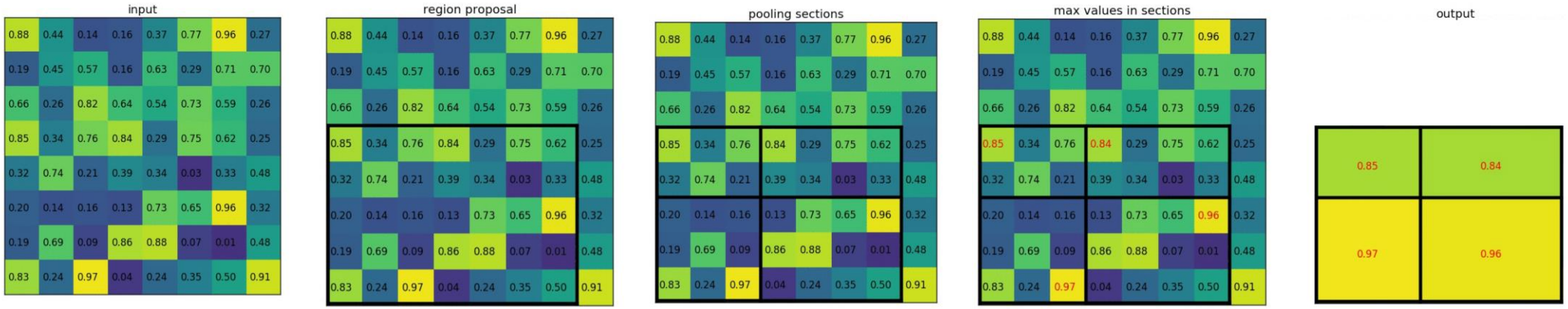
б. слой, реализующий регрессию, которая уточняет границы региона потенциального объекта.



RoI Pooling

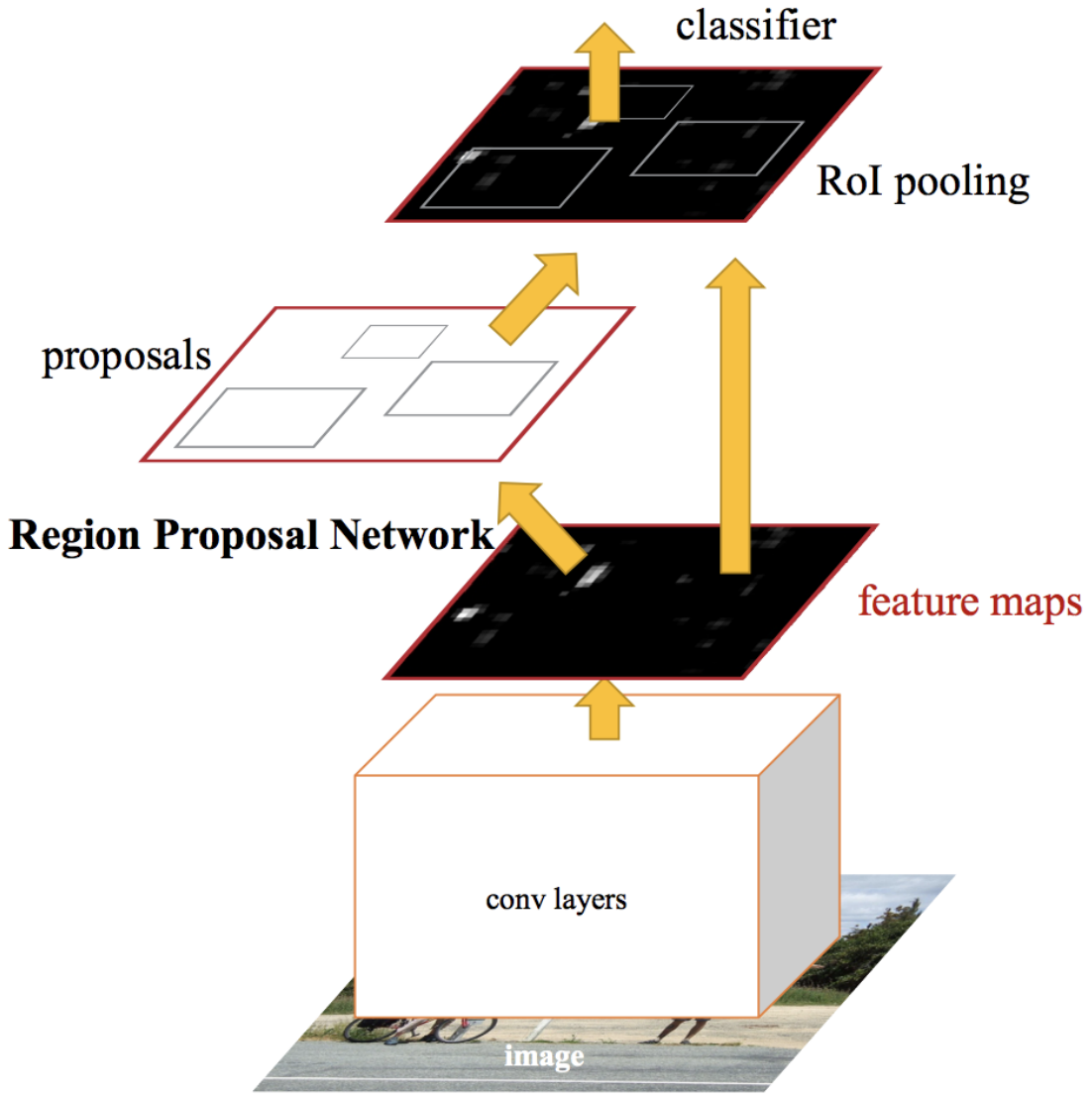
- Разделение предложенного региона на секции равного (если это возможно) размера, количество которых равно HxW.
- Нахождение максимума в каждой секции

В чем преимущество пула RoI? Один из них - скорость обработки. Для каждогог региона используется одна и та же карта признаков.



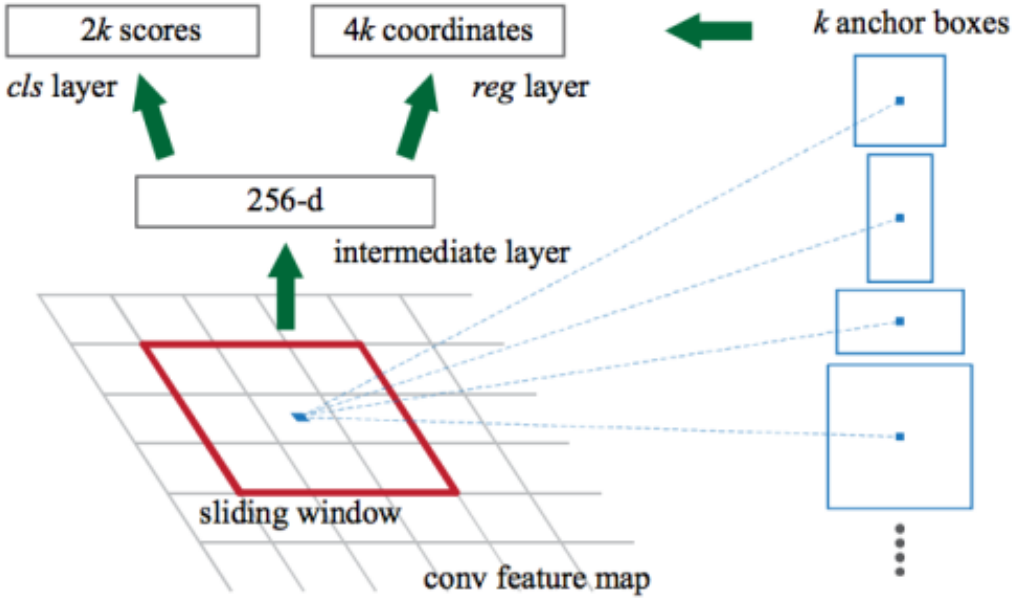
Faster R-CNN

- Полное изображение подается на вход сети для классификации (до последнего сверточного слоя включительно)
- Последний сверточный слой формирует карту признаков.
- Карта признаков обрабатывается слоем RPN (Region Proposal Network), который формирует регионы.
- Получаем карту признаком с регионами и действуем аналогично Fast R-CNN.



Region Proposal Network

- По извлечённым CNN признакам скользят «мини-нейросетью» с небольшим окном (3x3).
- Полученные с её помощью значения передаются в два параллельных полносвязанных слоя: **box-regression layer (reg)** и **box-classification layer (cls)**.
- Выходы этих слоёв базируются на **якорях** (**anchor**): k рамках для каждого положения скользящего окна, имеющих разные размеры и соотношения сторон.
- **Reg**-слой для каждого якоря выдаёт по 4 координаты, корректирующие положение охватывающей рамки.
- **Cls**-слой выдаёт два числа – вероятности того, что рамка содержит хоть какой-то объект или что не содержит.



Перенос стиля

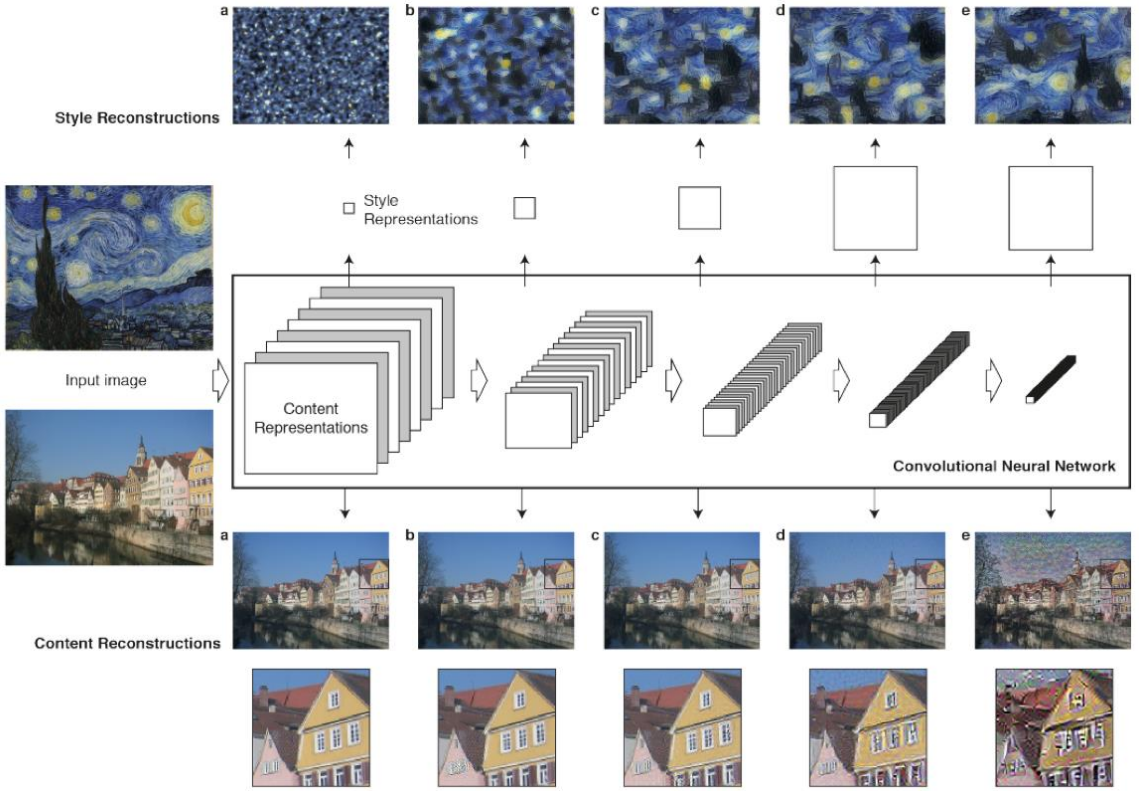
Задача: Взяв контент с одного изображения и стиль от второго, нейронная сеть объединяет их в одно художественное произведение

Способы решения:

- Сверточное нейронные сети
- Генеративные модели

Neural Style Transfer

- Основана на VGG19
- Не изменяем веса, а изменяем значения пикселей на третье картинке, созданной из шума
- Принцип переноса стиля состоит в том, чтобы определить две функции расстояния, одна из которых описывает, насколько различается содержимое двух изображений, Lcontent, а другая описывает разницу между двумя изображениями с точки зрения их стиля, Lstyle
- Реконструируем изображение из карт признаков поданных на обработку изображений



- Вычисляем корреляцию между различными функциями на разных уровнях CNN. Мы реконструируем стиль входного изображения из признаков, построенных на различных подмножествах слоев CNN.

Neural Style Transfer Losses

- Заменили max-polling на average
- Contented и style суммируются с различными весами
- A - изображение, которое мы подбираем. F^l - выход l-го слоя для изо
- A ^ l - это представление исходного изображения, а G ^ l - представление сгенерированного изображения в слое l
- P ^ l - это представление исходного изображения, а F ^ l - представление сгенерированного изображения на картах характеристик слоя l.

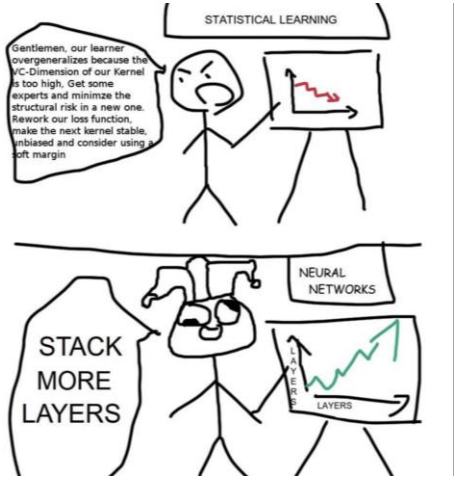
Content loss

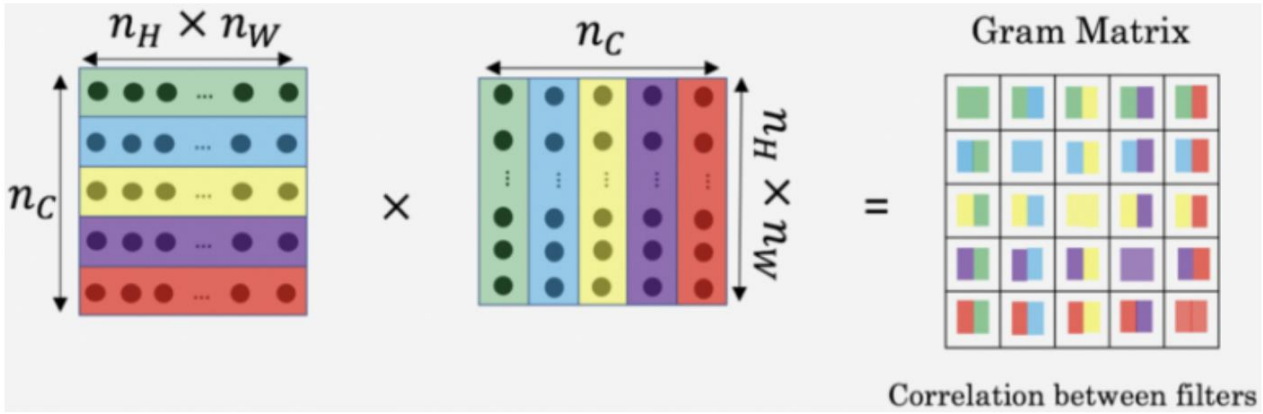
$$\mathcal{L}_{content}(\vec{p}, \vec{x}, l) = \frac{1}{2} \sum_{i,j} \left(F_{ij}^l - P_{ij}^l\right)^2$$

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l.$$

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} \left(G_{ij}^l - A_{ij}^l\right)^2$$

$$\mathcal{L}_{style}(\vec{a}, \vec{x}) = \sum_{l=0}^{L} w_l E_l$$





Итоги

Какие задачи и алгоритмы CV можно найти в презентации?

- Классификация + локализация: LeNet, AlexNet, VGG, Inception, ResNet, DenseNet
- Сегментация: FCN (Semantic), Mask R-CNN (Instance), Panoptic
- Детекция: R-CNN, Fast R-CNN, Faster R-CNN
- Перенос стиля: Neural Style Transfer

Что дальше?

Управление беспилотниками, распознавание лиц, прогнозирование заболеваний по снимкам рентген, мрт и тд.

Источники

Статьи:

- **CNN:** https://arxiv.org/ftp/arxiv/papers/1901/1901.06032.pdf
- **LeNet:** http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf
- **AlexNet:** https://www.researchgate.net/publication/221361415_ImageNet_a_Large-Scale_Hierarchical_Image_Database
- **VGG:** https://arxiv.org/pdf/1409.1556.pdf
- **Inception:** https://static.googleusercontent.com/media/research.google.com/ru//pubs/archive/44903.pdf
- **ResNet:** https://arxiv.org/pdf/1512.03385.pdf
- **DenseNet:** https://arxiv.org/pdf/1608.06993.pdf
- **Detection:**
- **R-CNN, Fast R-CNN, Faster R-CNN:** https://arxiv.org/pdf/1311.2524.pdf
  https://arxiv.org/pdf/1504.08083.pdf
  https://arxiv.org/pdf/1506.01497.pdf
- **Fully Convolutional Networks for Semantic Segmentation** https://arxiv.org/abs/1411.4038
- **Mask R_CNN** https://arxiv.org/pdf/1703.06870.pdf
- **Panoptic segmentation** https://arxiv.org/pdf/1801.00868.pdf
- **Neural Style Transfer** https://arxiv.org/pdf/1508.06576.pdf

Источники

Ссылки для лучшего понимания:

- **CNN:** https://cs231n.github.io/convolutional-networks/
- https://www.kaggle.com/getting-started/171198
- https://neurohive.io/ru/osnovy-data-science/glubokaya-svertochnaja-nejronnaja-set/
- **LeNet:** https://engmrk.com/lenet-5-a-classic-cnn-architecture/
- https://www.kaggle.com/blurredmachine/lenet-architecture-a-complete-guide
- **ImageNet:** https://machinelearningmastery.com/introduction-to-the-imagenet-large-scale-visual-recognition-challenge-ilsvrc/
- **AlexNet:** https://neurohive.io/ru/vidy-nejrosetej/alexnet-svjortochnaja-nejronnaja-set-dlja-raspoznavanija-izobrazhenij/
- https://towardsdatascience.com/alexnet-the-architecture-that-challenged-cnns-e406d5297951
- **VGG:** https://neurohive.io/en/popular-networks/vgg16/
- **Inception:** https://habr.com/ru/post/302242/
- https://sh-tsang.medium.com/review-inception-v3-1st-runner-up-image-classification-in-ilsvrc-2015-17915421f77c
- **ResNet:** https://neurohive.io/ru/vidy-nejrosetej/resnet-34-50-101/
- https://blog.paperspace.com/popular-deep-learning-architectures-resnet-inceptionv3-squeezenet/
- https://www.machinelearningmastery.ru/understanding-and-coding-a-resnet-in-keras-446d7ff84d33/
- **DenseNet:** https://habr.com/ru/post/498168/#DenseNet
- https://towardsdatascience.com/review-densenet-image-classification-b6631a8ef803
- **Localization:** http://datahacker.rs/deep-learning-object-localization/
- **R-CNN, Fast R-CNN, Faster R-CNN:** https://habr.com/ru/post/421299/
- https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e
- https://leonardoaraujosantos.gitbook.io/artificial-inteligence/machine_learning/deep_learning/object_localization_and_detection