

Занятие № 10

# Свёрточные нейронные сети. Часть 2

# План занятия



1. Задачи детектирования и сегментации объектов
2. Модель RCNN
3. Модель Fast RCNN
4. Модель Faster RCNN
5. Модель Mask RCNN
6. Автоэнкодеры
7. SegNet
8. U-net

# Задачи обработки изображений

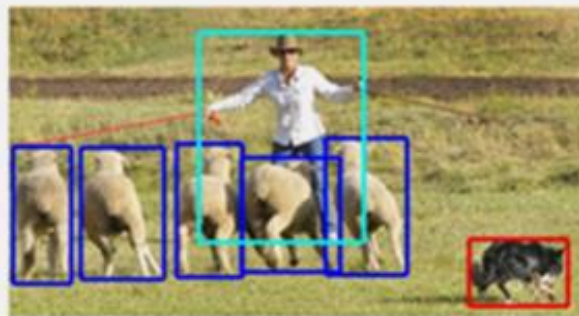


Основные задачи обработки изображений:

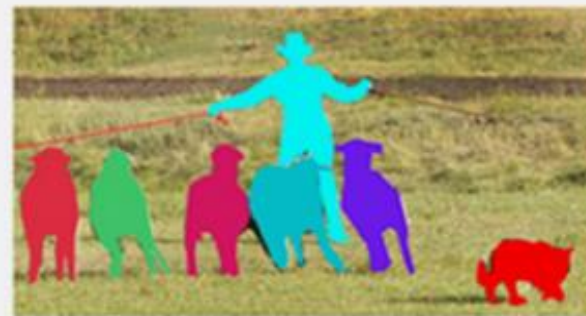
- а. Классификация объектов
- б. Детектирование объектов**
- с. Сегментация объектов



(a) classification



(b) detection



(c) segmentation

# Соревнование PASCAL Visual Object Classes



Цель соревнования:

Для 20 классов объектов надо:

1. Предсказать наличие объектов на изображении
2. Для каждого объекта предсказать рамку вокруг и метку класса

Объем данных:

~27k объектов на ~11k изображений

# Соревнование PASCAL Visual Object Classes

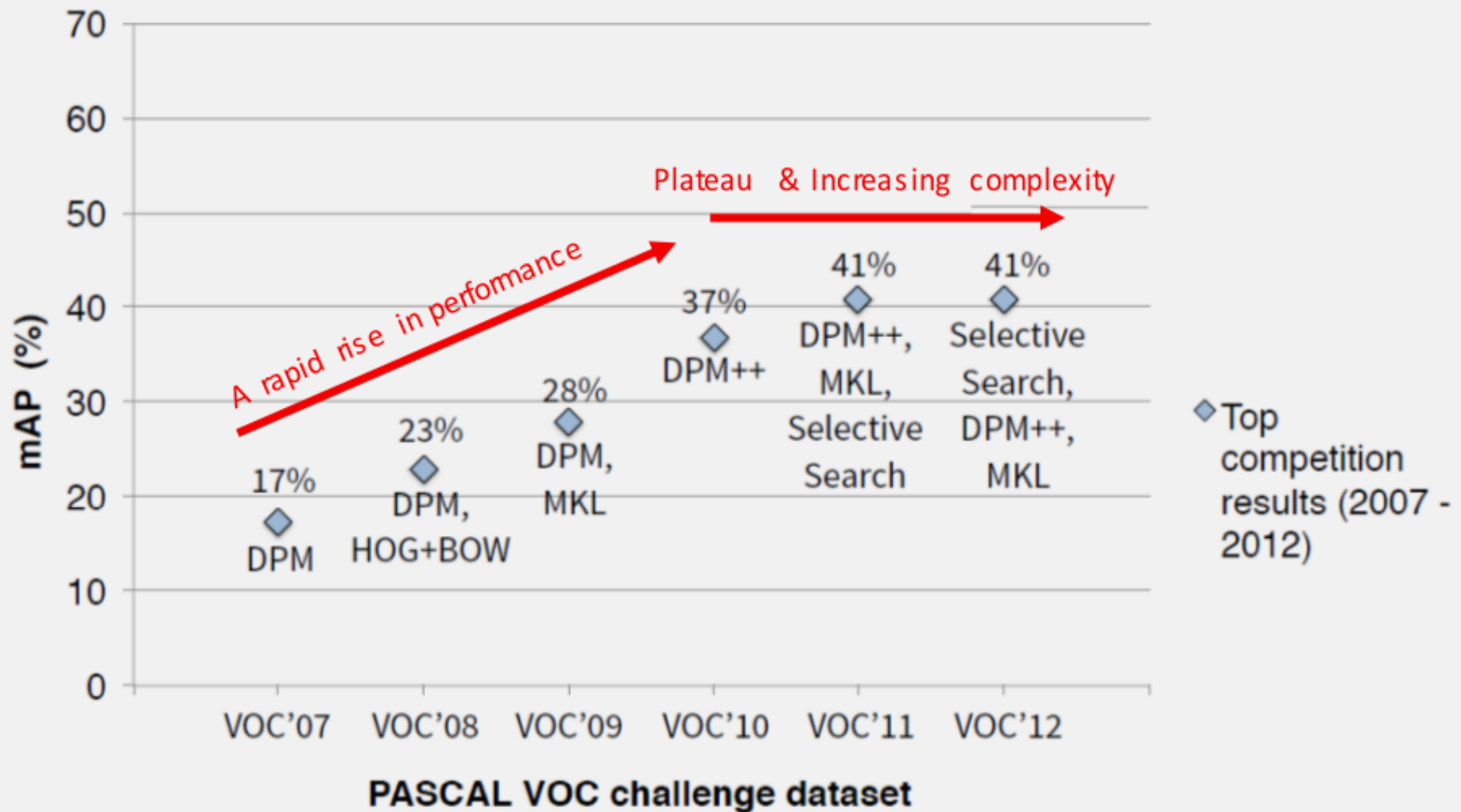


Классы объектов:

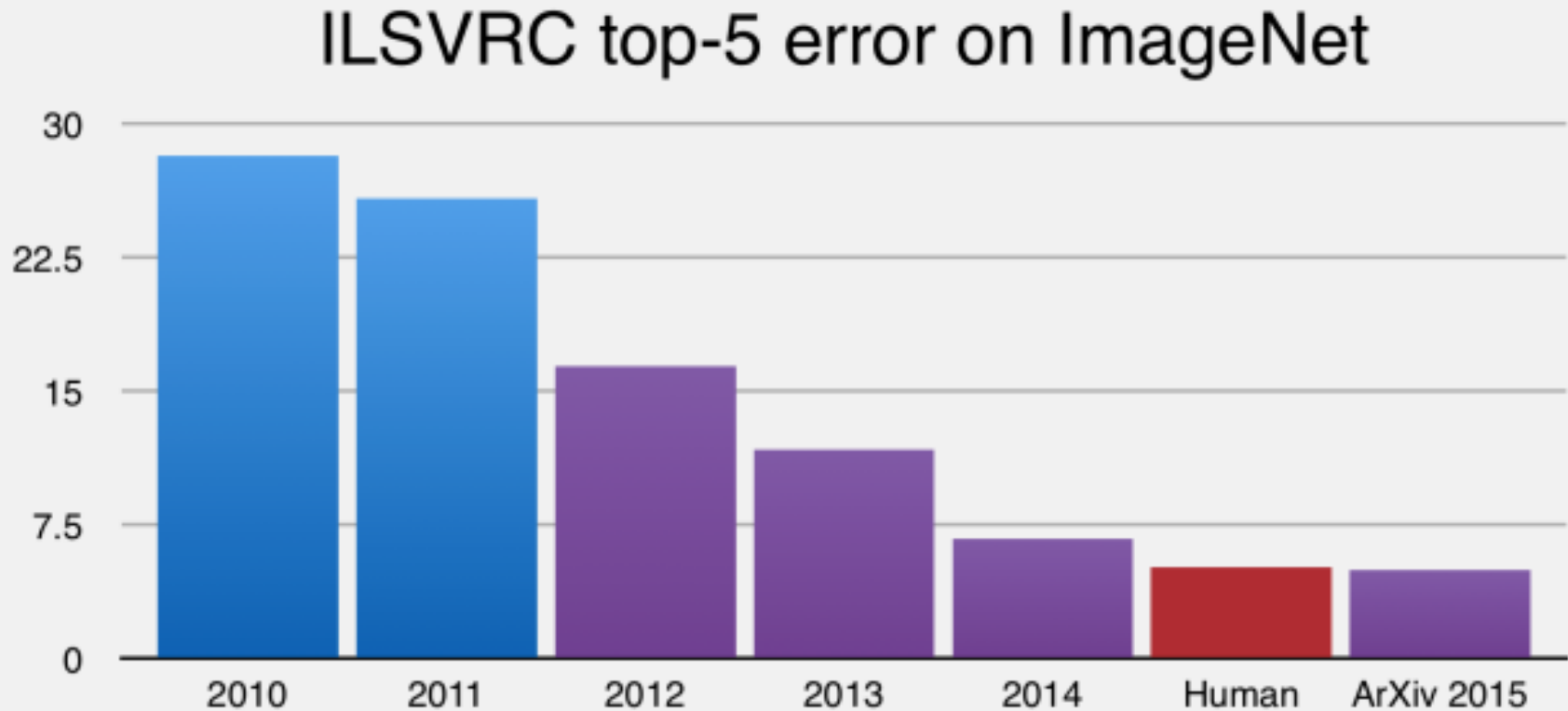
- Person: person
- Animal: bird, cat, cow, dog, horse, sheep
- Vehicle: airplane, bicycle, boat, bus, car, motorbike, train
- Indoor: bottle, chair, dining table, potted plant, sofa, TV/monitor

Метрика соревнования: Mean Average Precision (mAP)

# Прогресс в соревновании PASCAL VOC

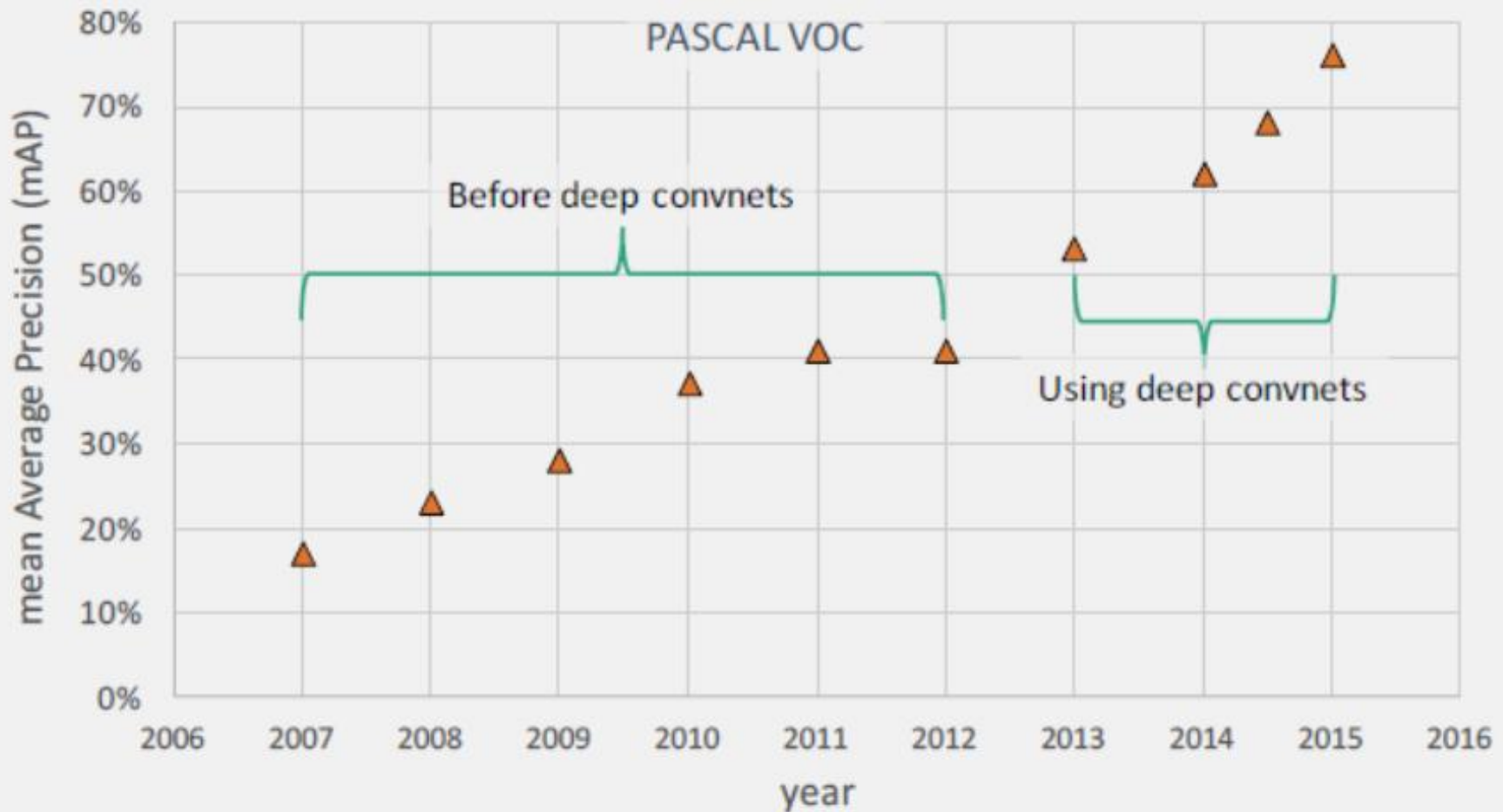


# Прогресс в соревновании ILSVRC



Фиолетовым отмечены модели на основе сверточных нейронных сетей

# Прогресс в соревновании PASCAL VOC после революции CNN



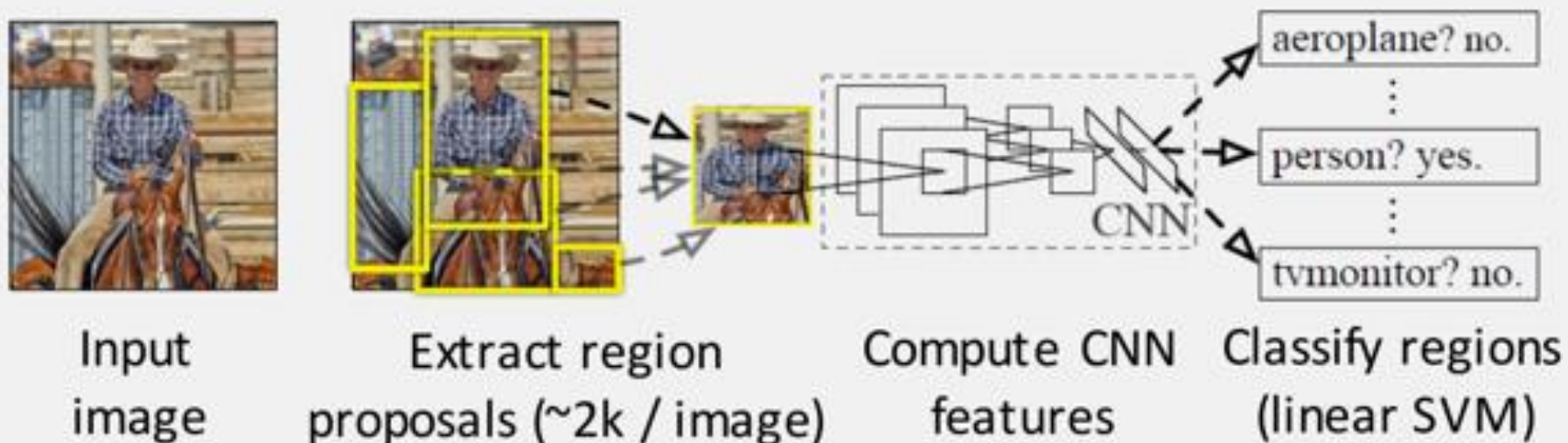


# RCNN (2014), идея



RCNN - Regions with CNN features

Идея: использование сверточной нейронной сети для детектирования объектов



# Выбор рамок потенциальных объектов на изображении



Проблема:

Проход скользящим окном на различных масштабах порождает очень много потенциальных рамок объектов

Решение:

Выбираем только те рамки, в которых потенциально могут быть объекты используя данные о цвете изображения

# Selective Search (2012)

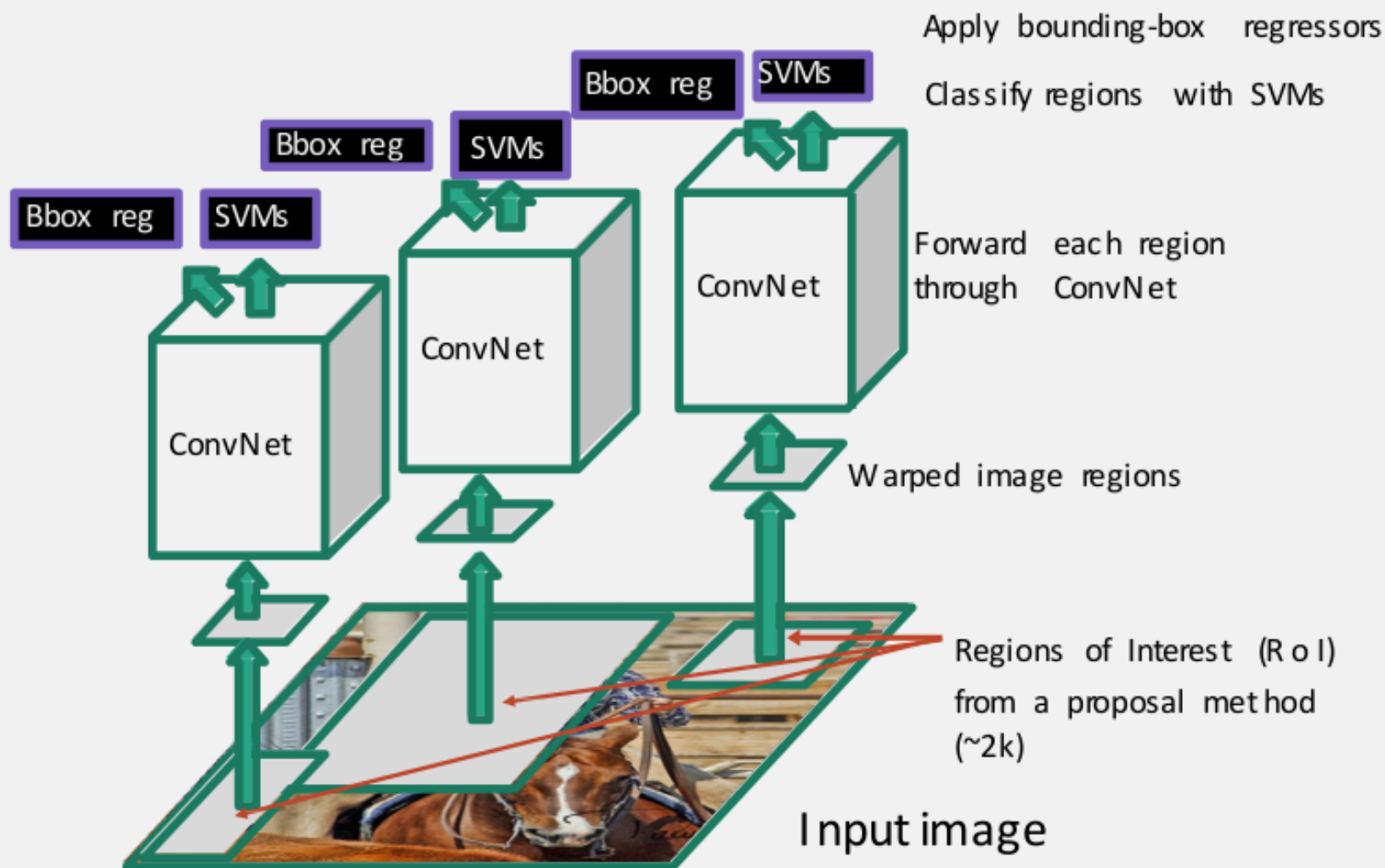


## Алгоритм Selective Search



Снижаем количество рамок с  $\sim 100k$  до  $\sim 2k$

# RCNN, архитектура



# Проблемы RCNN



- Используется предтренированная сверточная нейронная сеть
- Обучение экстрактора признаков, классификатора объектов и регрессора bbox-ов происходит раздельно
- Система работает медленно

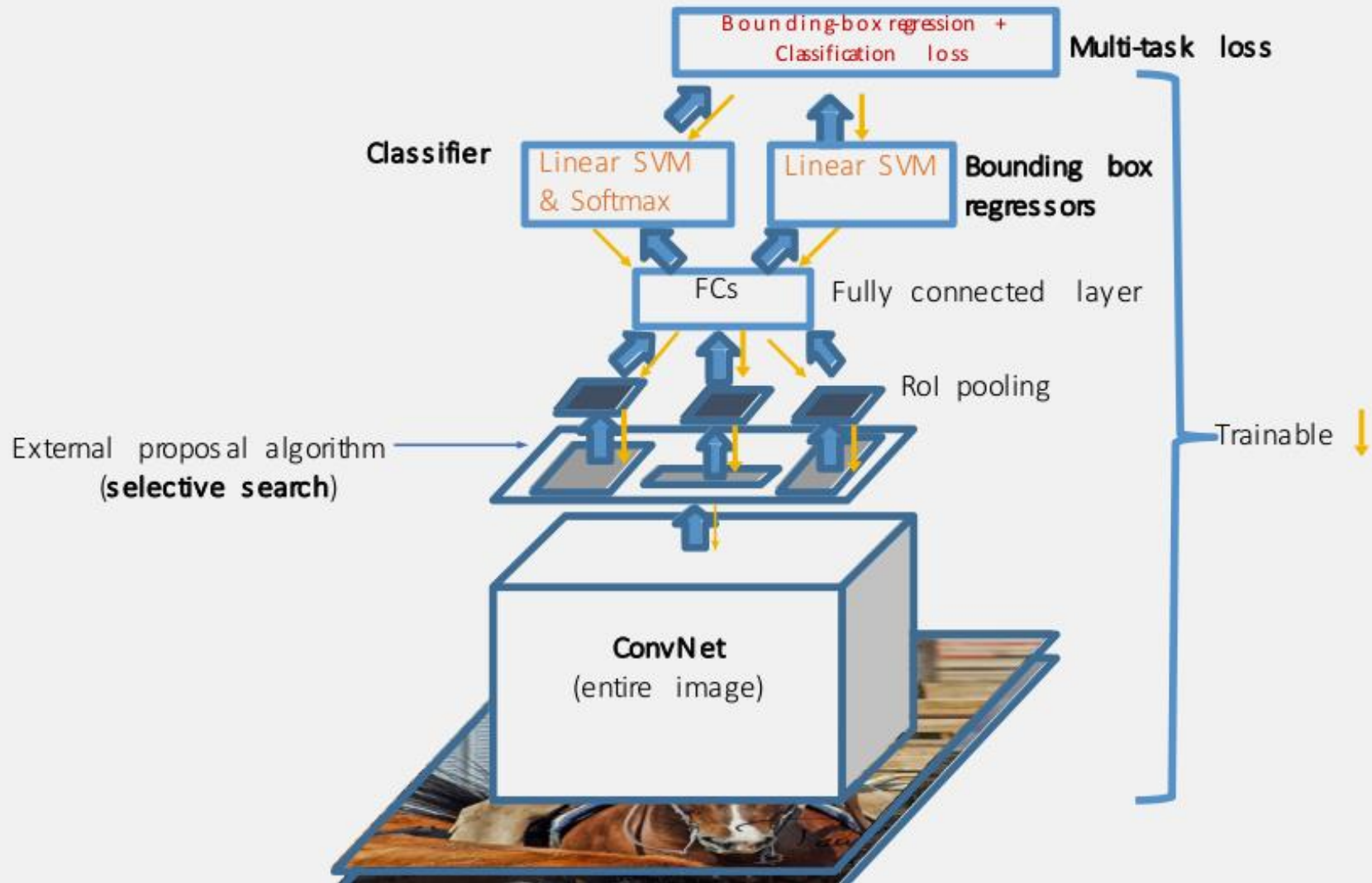
# Fast RCNN (2015), идея



Идея:

- Сделаем end-to-end систему
- Будем извлекать признаки из изображения только 1 раз

# Fast RCNN, архитектура



# RCNN и Fast RCNN



Training time:	84 hours / 8.75 hours
VOC07 test mAP:	66.0% / 68.1%
Testing time per image:	49s / 2.32s



# Проблемы Fast RCNN



- Selective Search не тренируется
- Selective Search работает медленно

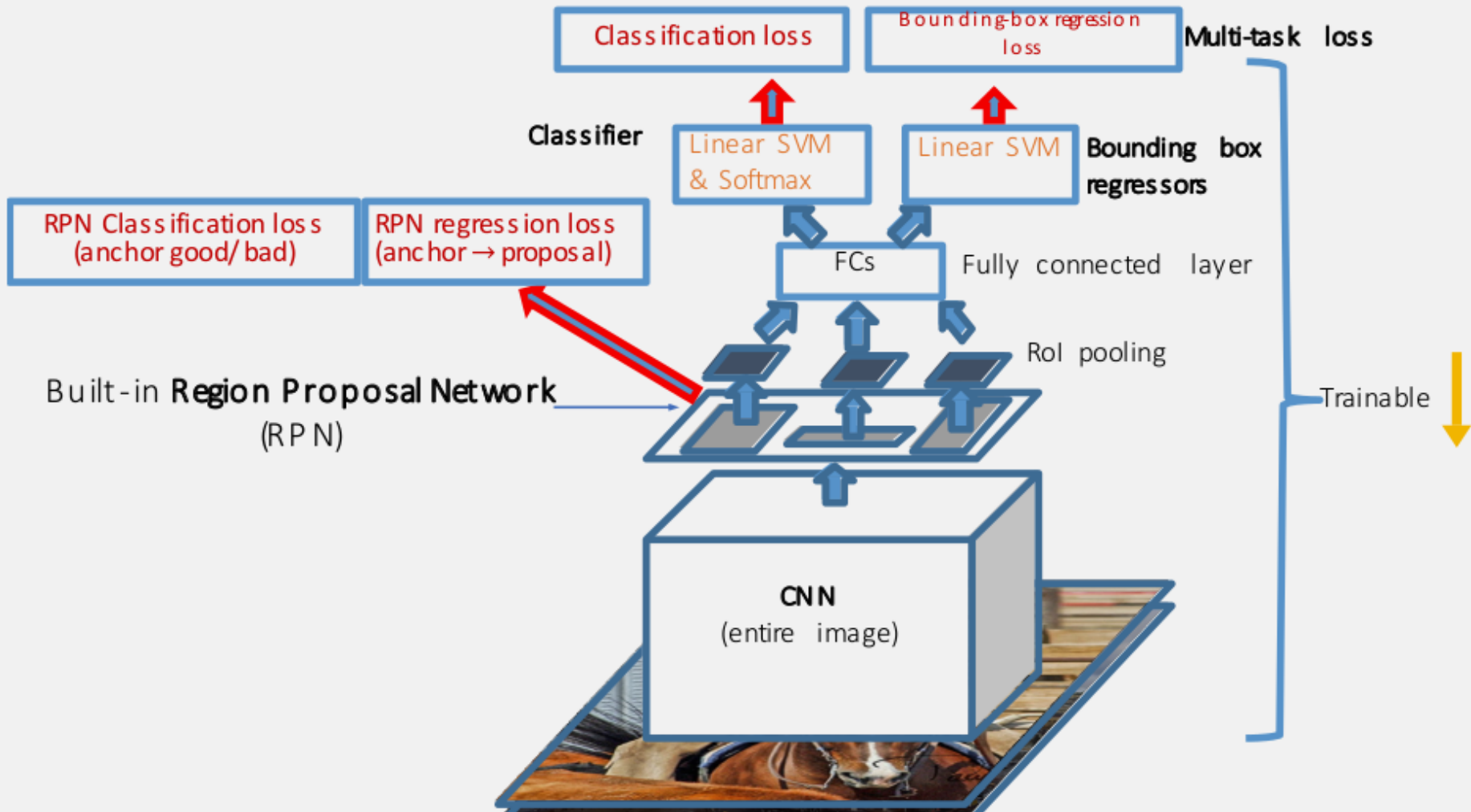
# Faster RCNN (2015), идея



Идея:

Заменим Selective Search на тренируемую Region Proposal Network

# Faster RCNN, архитектура



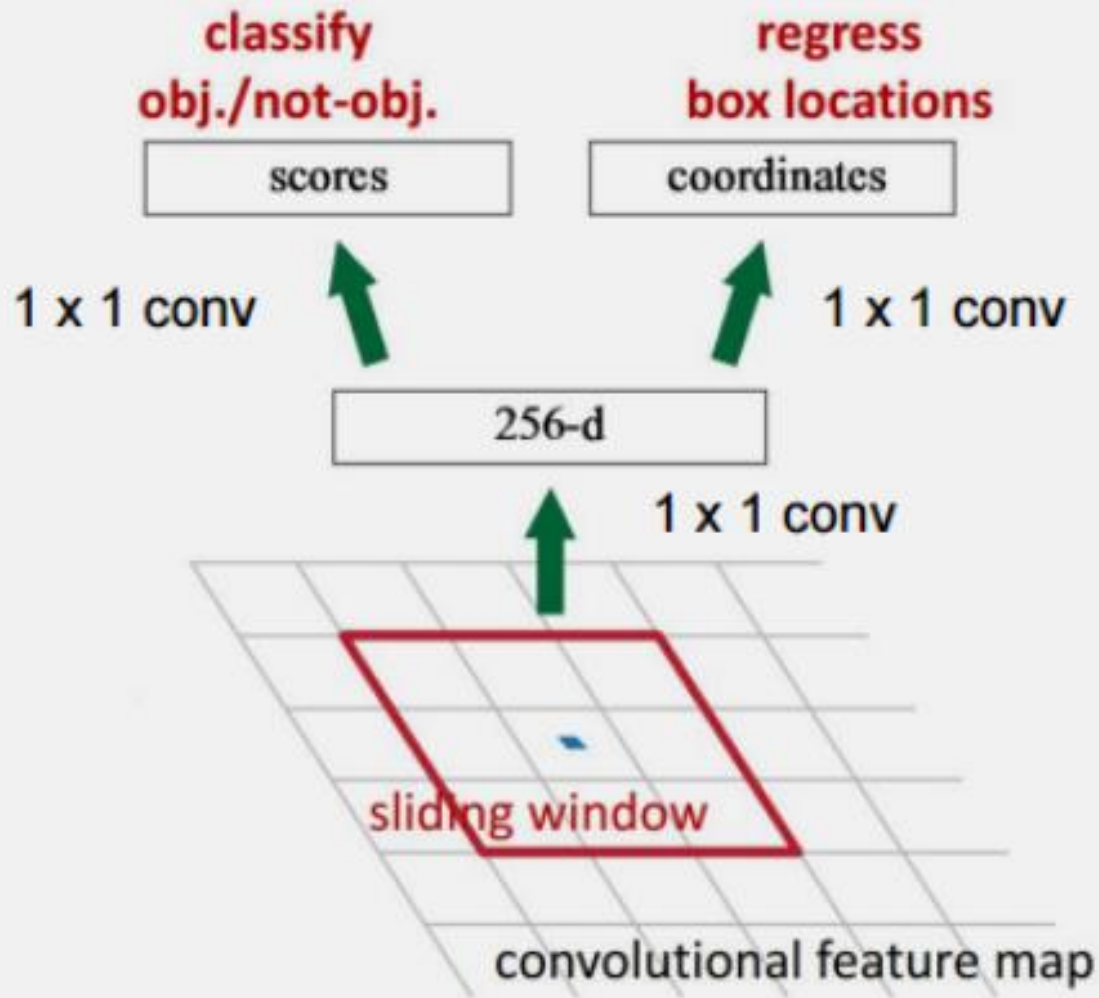
# Region Proposal Network



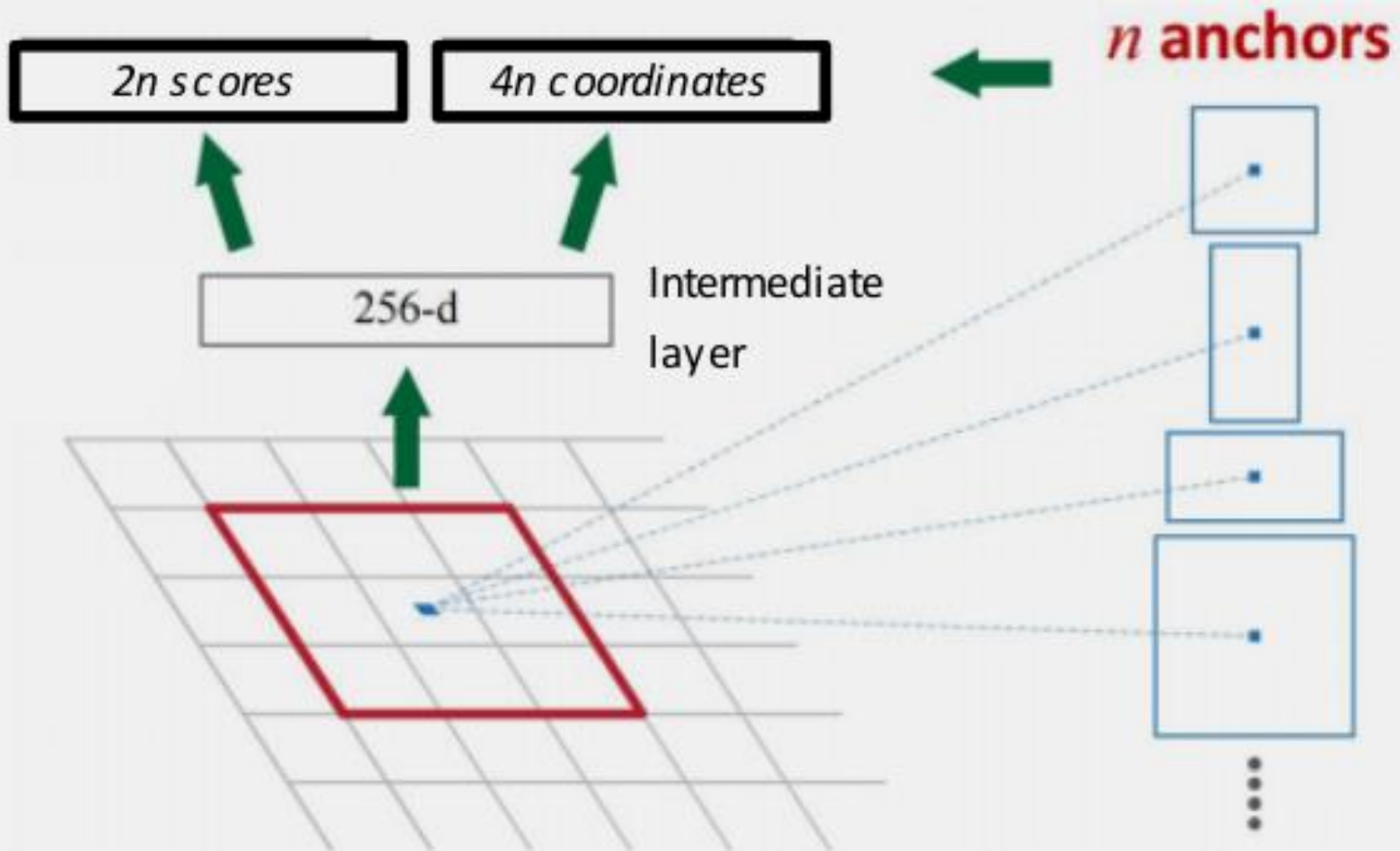
RPN - это небольшая нейронная сеть, которая на вход принимает карты признаков и предсказывает:

1. Есть ли в данной области объект
2. Уточняет локализацию Bbox-ов

# Region Proposal Network



# Region Proposal Network



# Сравниваем RCNN, Fast RCNN, Faster RCNN

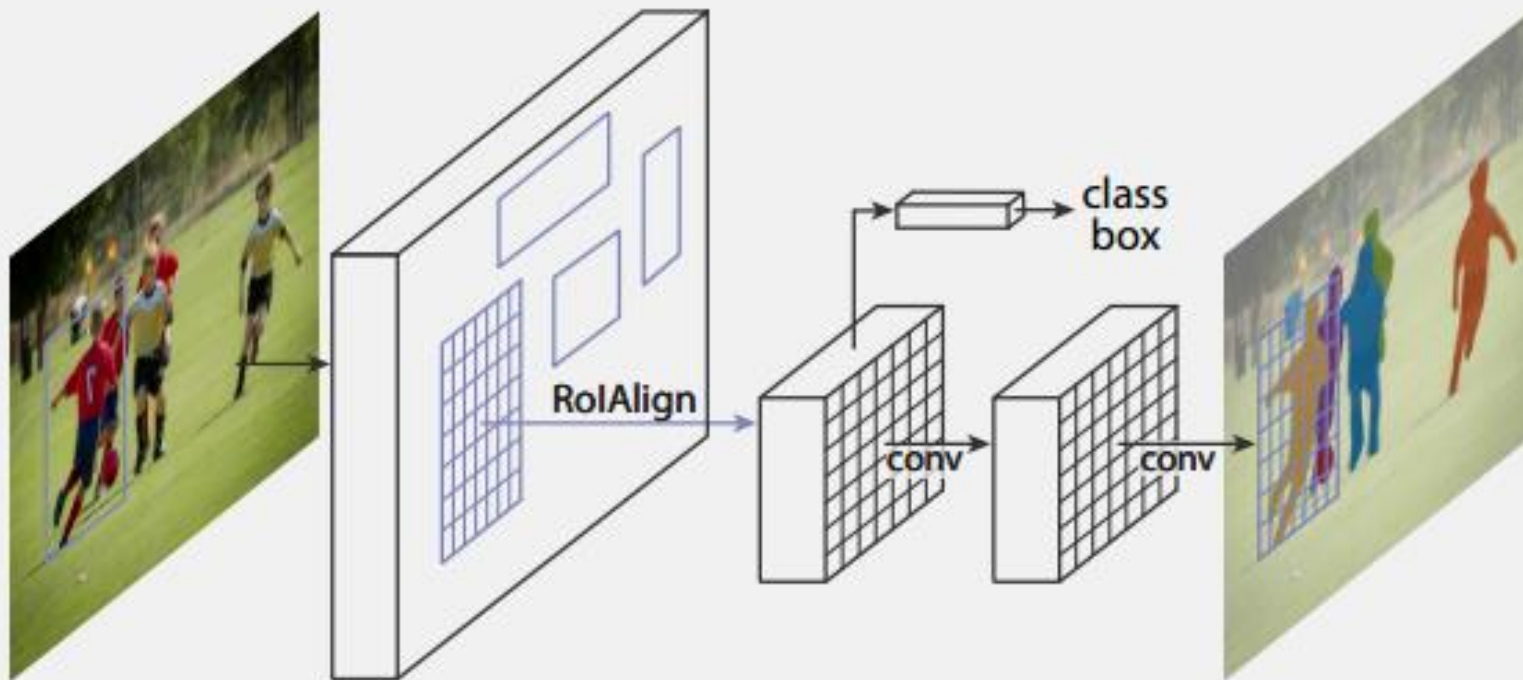


	RCNN	Fast RCNN	Faster RCNN
PASCAL VOC 2007 mAP	66.0	66.9	66.9
Время на предсказание	50 сек.	2 сек.	0.2 сек.
Ускорение	1x	25x	250x

# Mask RCNN (2017), идея

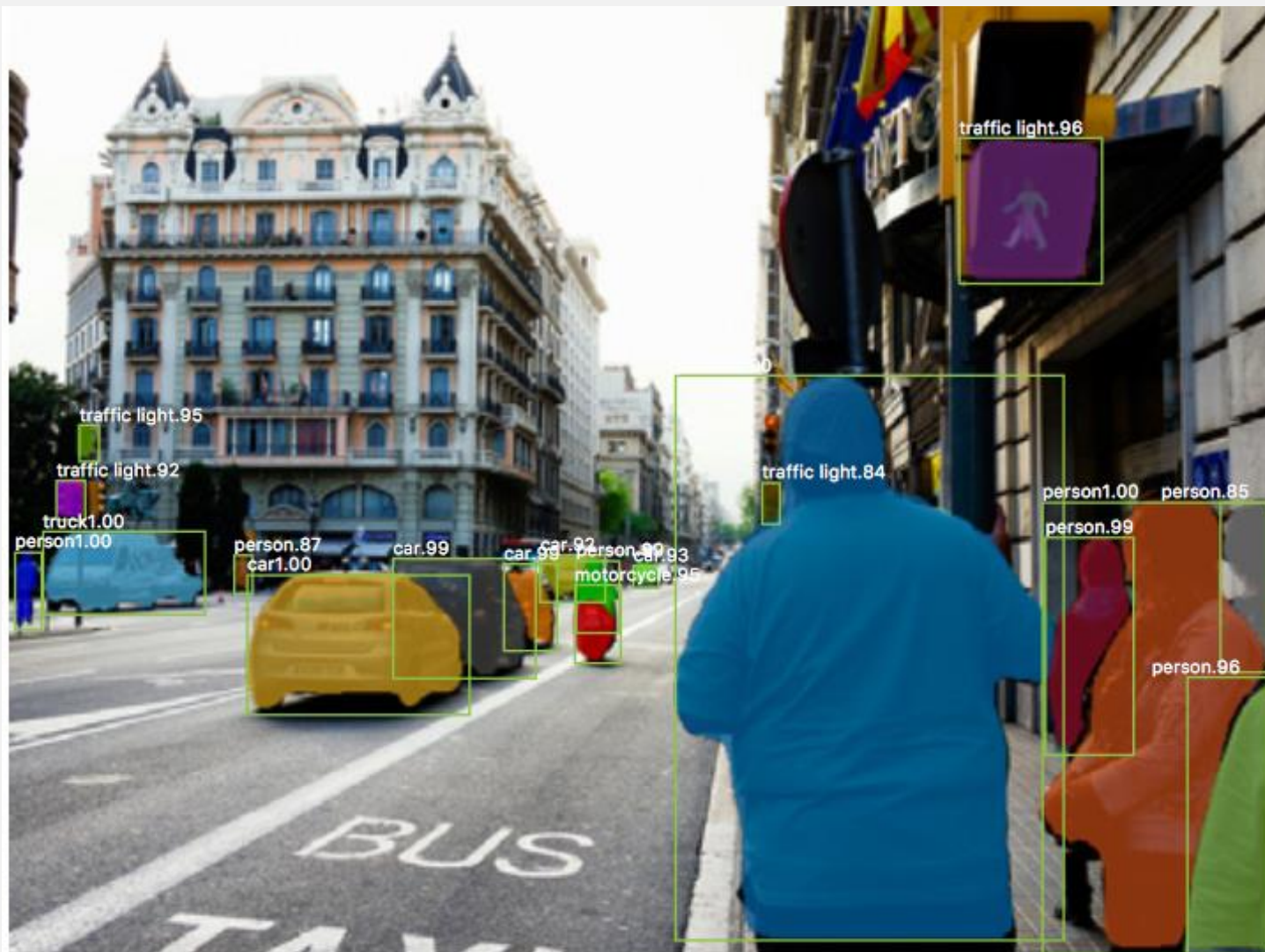


Добавим еще одну ветвь, которая на основе карты признаков будет предсказывать сегментацию





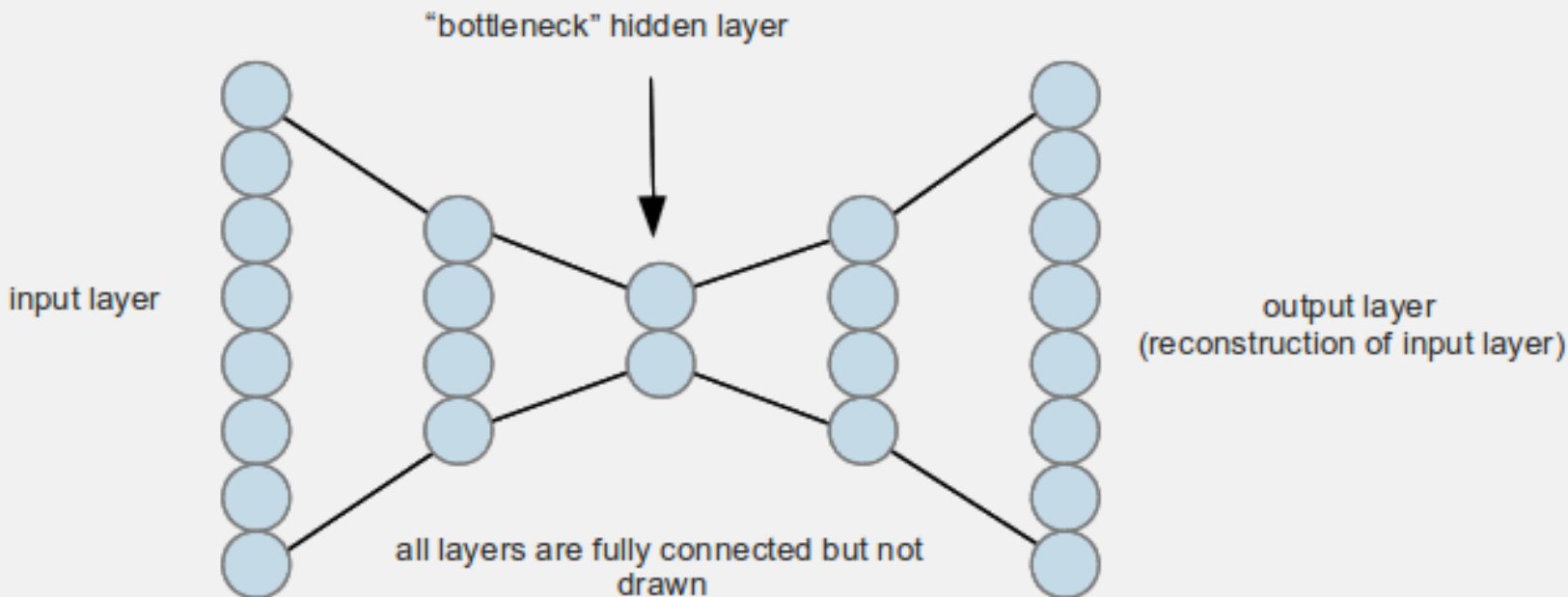
# Mask RCNN, сегментация



# Автоэнкодеры

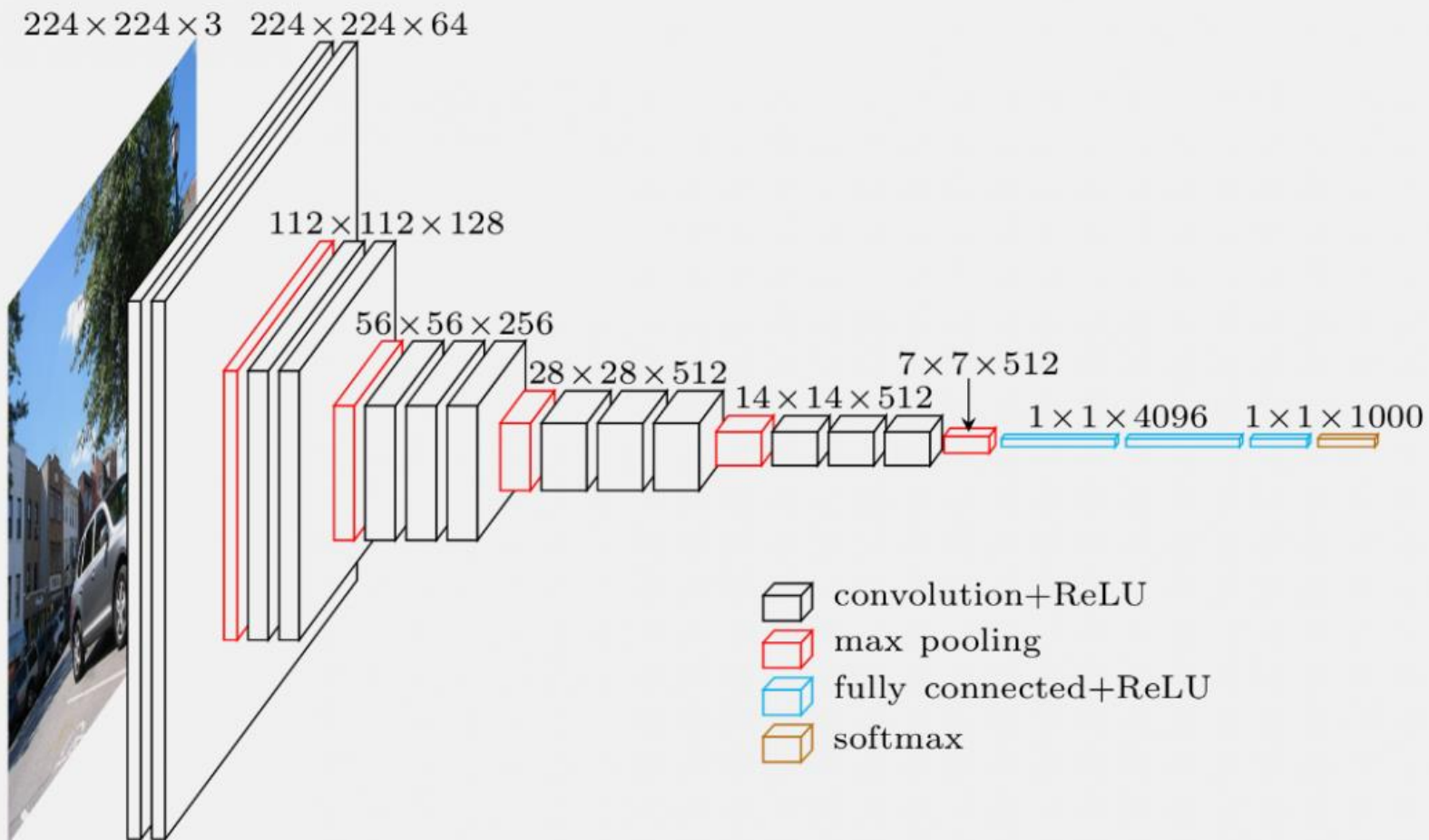


Автоэнкодеры - это нейронные сети, которые восстанавливают входной сигнал на выходе. Автоэнкодеры конструируются таким образом, чтобы не иметь возможность точно скопировать вход на выходе, сеть вынуждена учиться отбирать наиболее важные признаки.





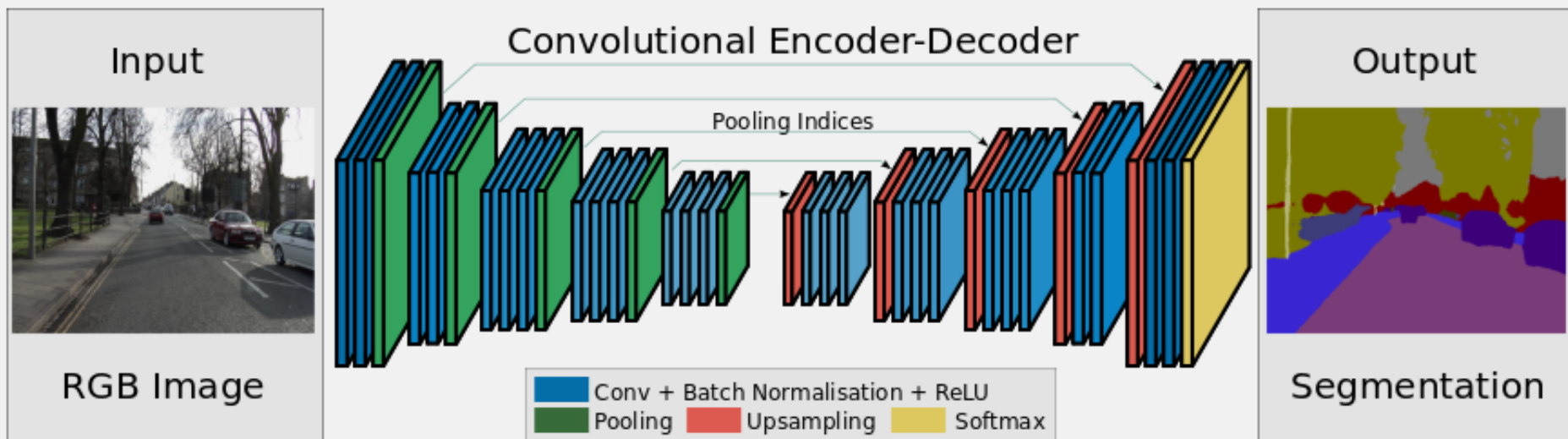
## Архитектура сети VGG-16 (победитель ILSVRC 2014)



# Архитектура SegNet (2015)



SegNet по сути сверточный автоэнкодер с хитрой реализацией операции unpooling



# Архитектура SegNet, unpooling



0.1	0.5	<b>1.2</b>	-0.7
<b>0.8</b>	-0.2	-0.5	0.3
0.4	<b>0.9</b>	-0.1	-0.2
-0.6	0.1	<b>0.5</b>	0.3

max-pooling

0.8	1.2
0.9	0.5

		x	
x			
	x		
		x	

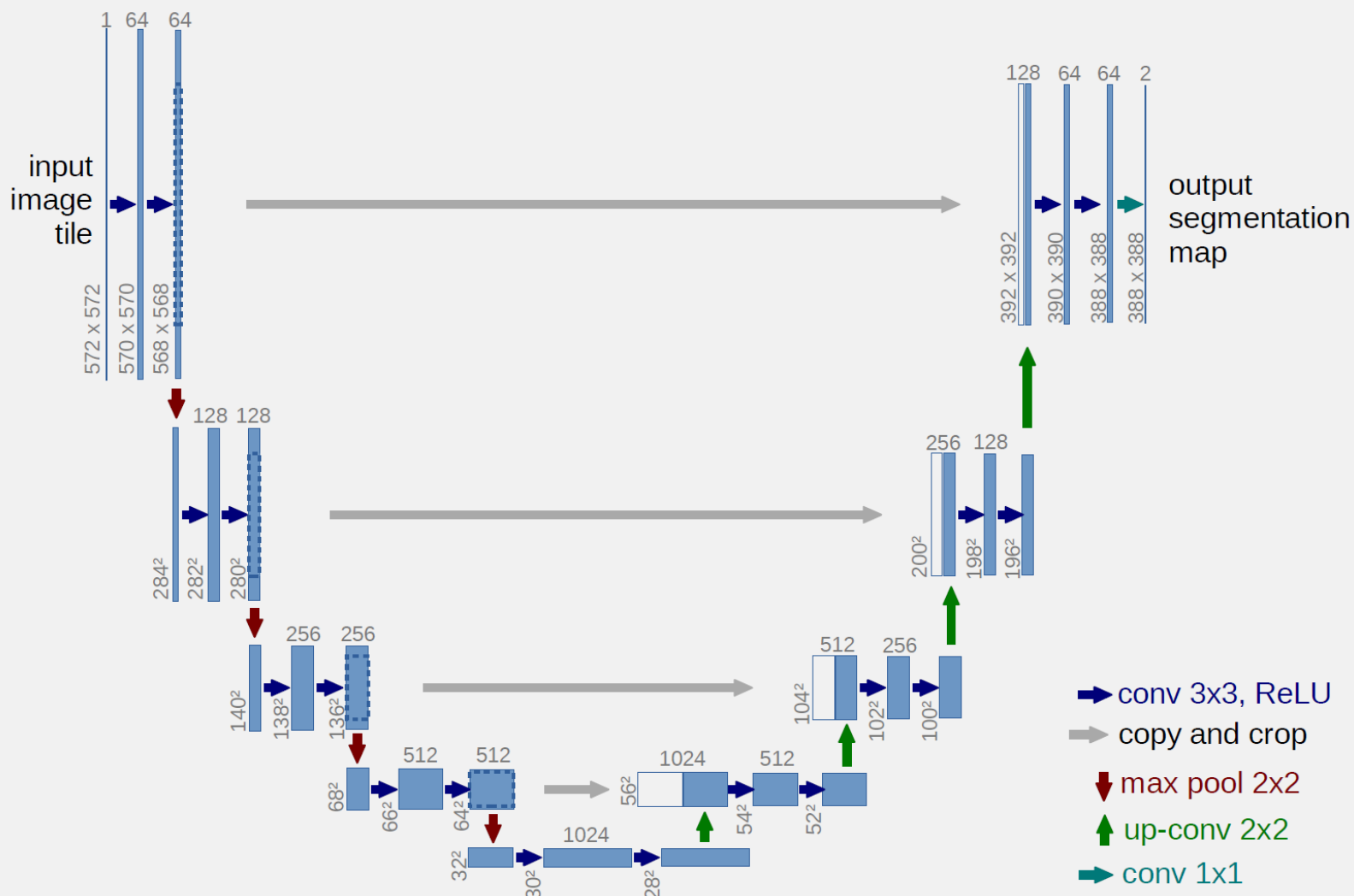
max locations

0	0	<b>0.5</b>	0
<b>1.3</b>	0	0	0
0	<b>0.4</b>	0	0
0	0	<b>0.1</b>	0

unpooling

1.3	0.5
0.4	0.1

# Архитектура U-net (2015)

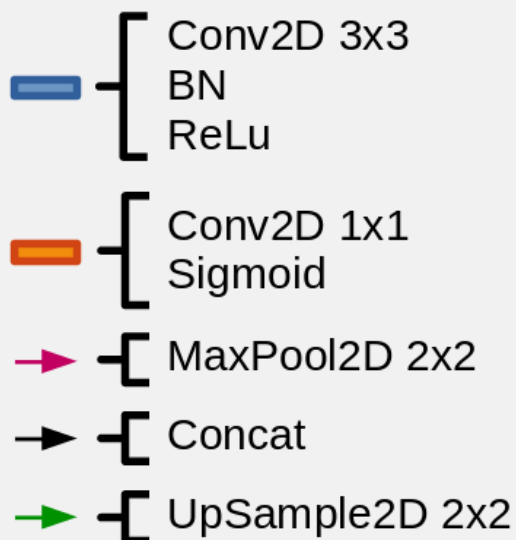




Задача:

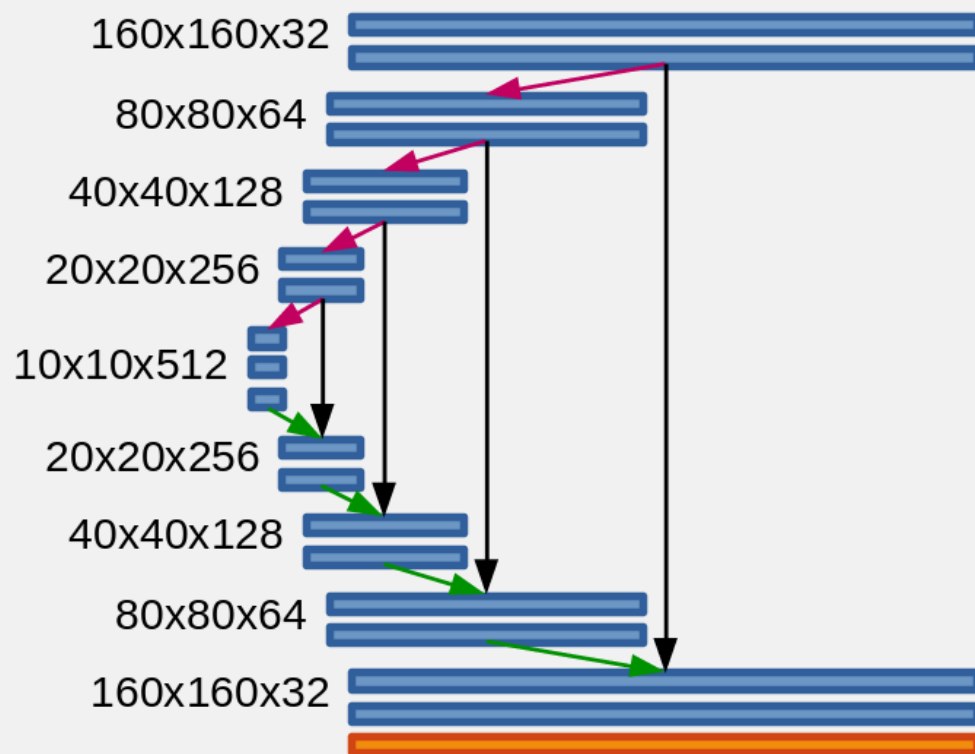
- 25 спутниковых изображений в Train
- 425 спутниковых изображений в Test
- 10 классов объектов
- Изображения представляют участки поверхности земли 1x1 км
- Для каждого участка 1x1 км даны 4 файла tiff с разных приборов

# U-net, пример



Input: 160x160x20

Output: 160x160x7 or 160x160x2





# Сегментация Train



# Сегментация Train

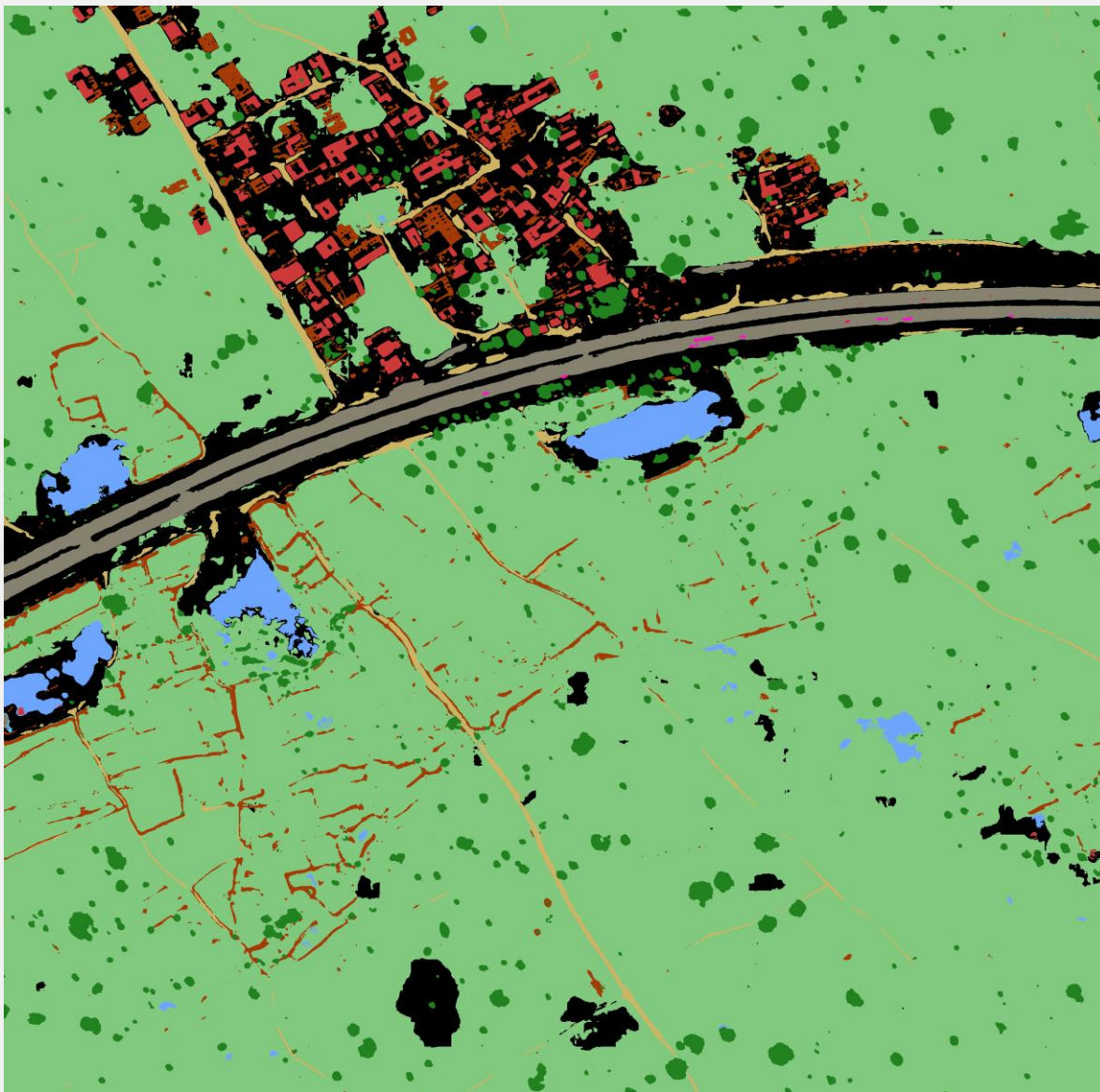




# Сегментация Test



# Сегментация Test





**Спасибо за  
внимание!**

Евгений Некрасов

[e.nekrasov@corp.mail.ru](mailto:e.nekrasov@corp.mail.ru)