Deep Learning: Convolutional Neural Networks - Part II

```
$ echo "Data Sciences Institute"
```

CNNs for computer Vision

[Krizhevsky 2012]

[Ciresan et al. 2013]

[Faster R-CNN - Ren 2015]

[NVIDIA dev blog]

Beyond Image Classification

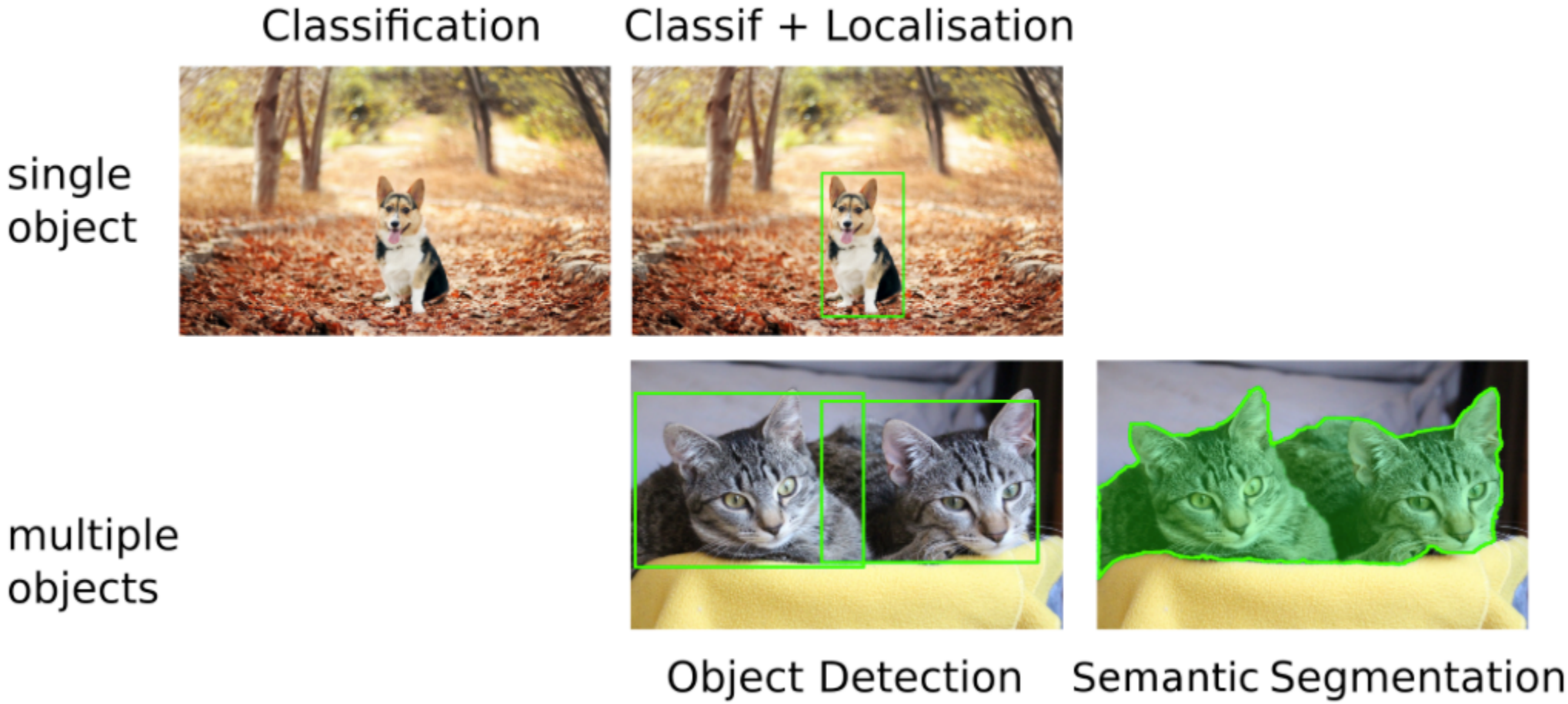
CNNs

- Previous lecture: image classification

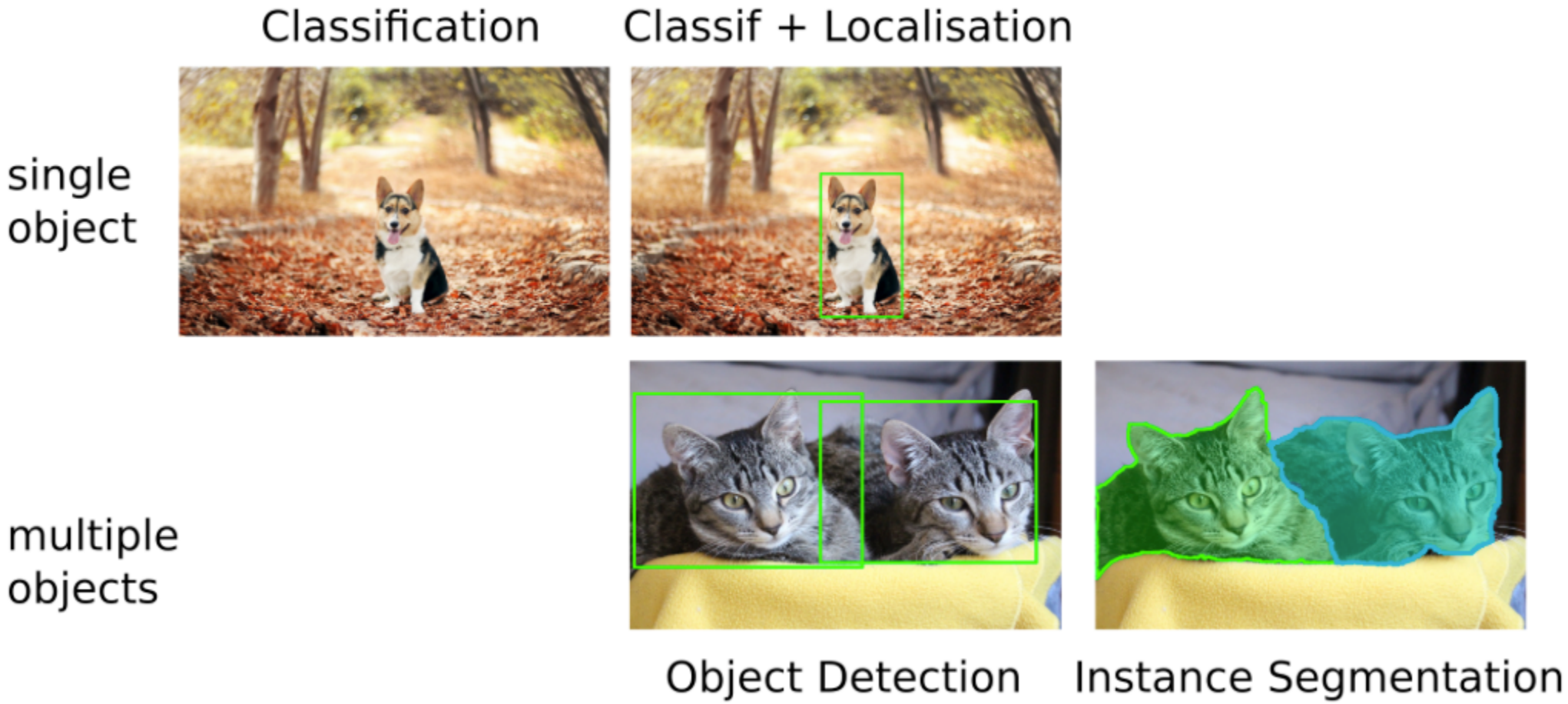
Limitations

- Mostly on centered images
- Only a single object per image
- Not enough for many real world vision tasks

Beyond Image Classification



Beyond Image Classification

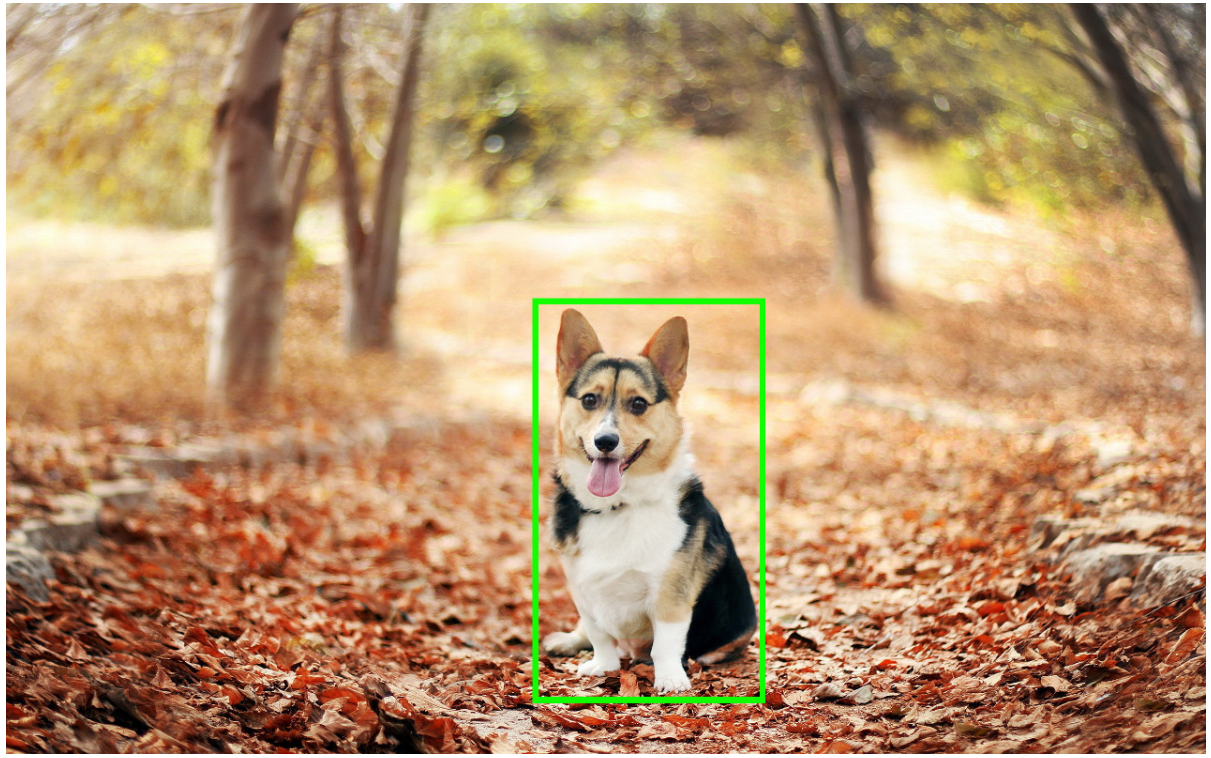


Outline

- Simple Localization as regression
- Detection Algorithms
- Fully convolutional Networks
- Semantic & Instance Segmentation

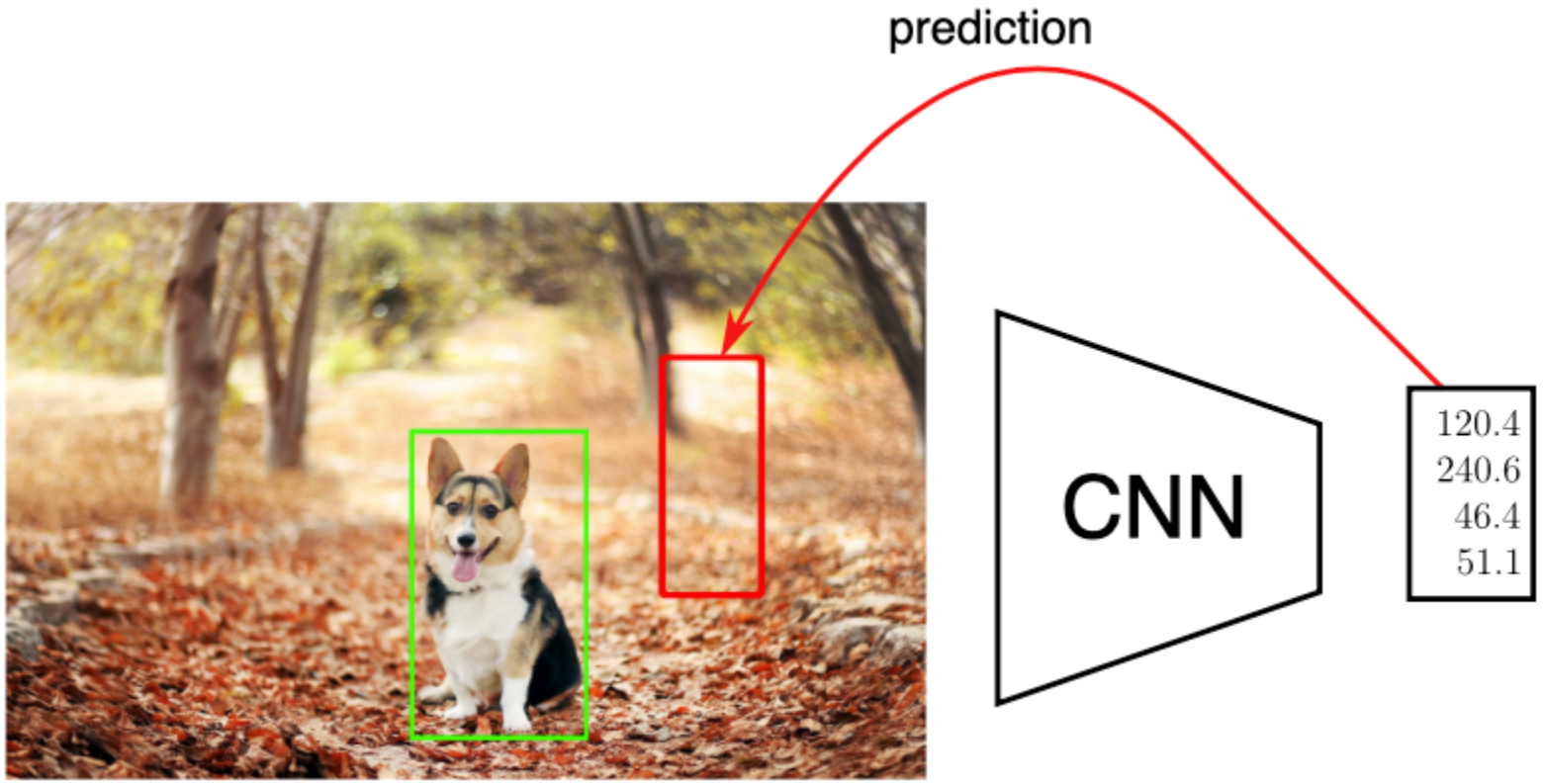
Localization

Localization

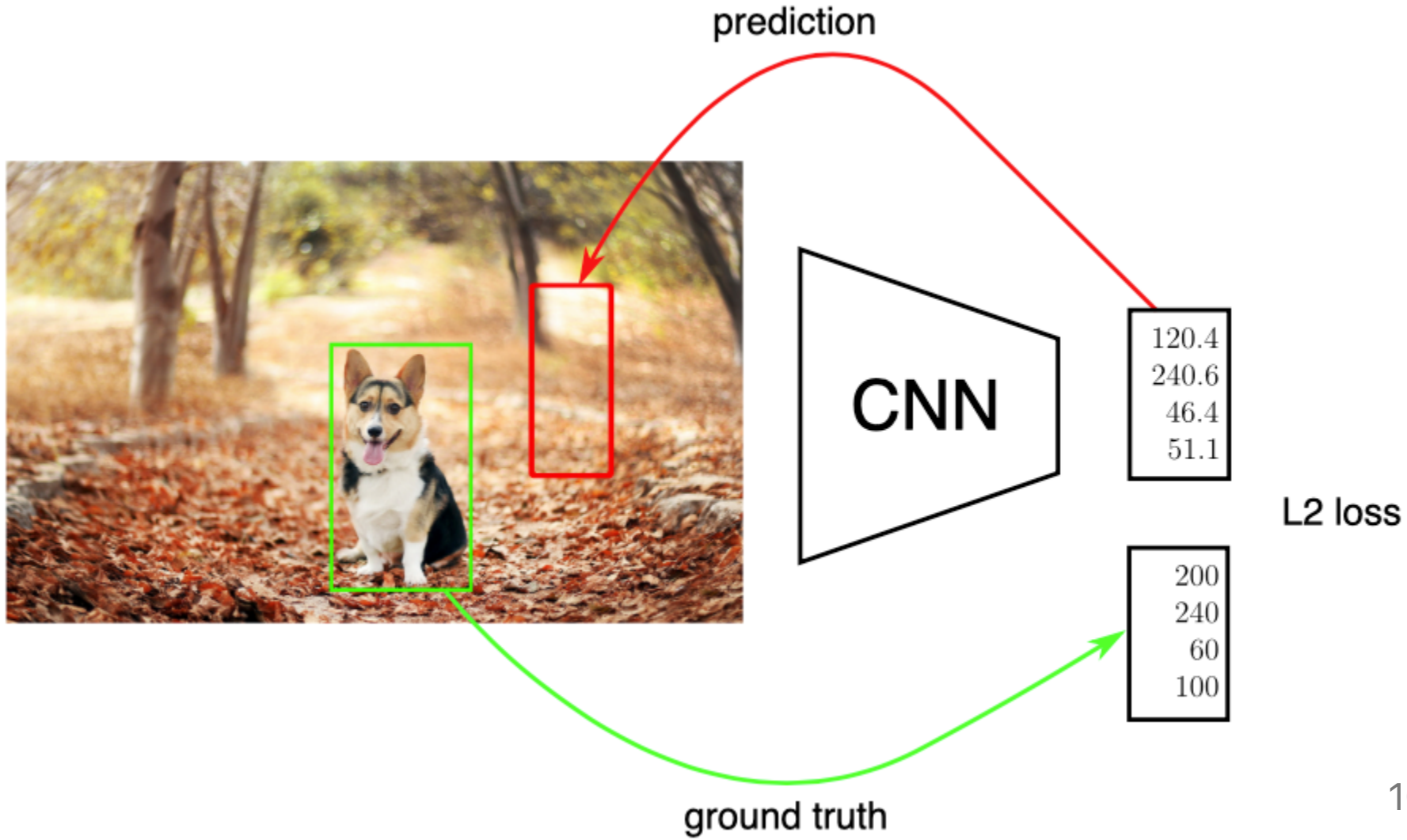


- Single object per image
- Predict coordinates of a bounding box `(x, y, w, h)`
- Evaluate via Intersection over Union (IoU)

Localization as regression

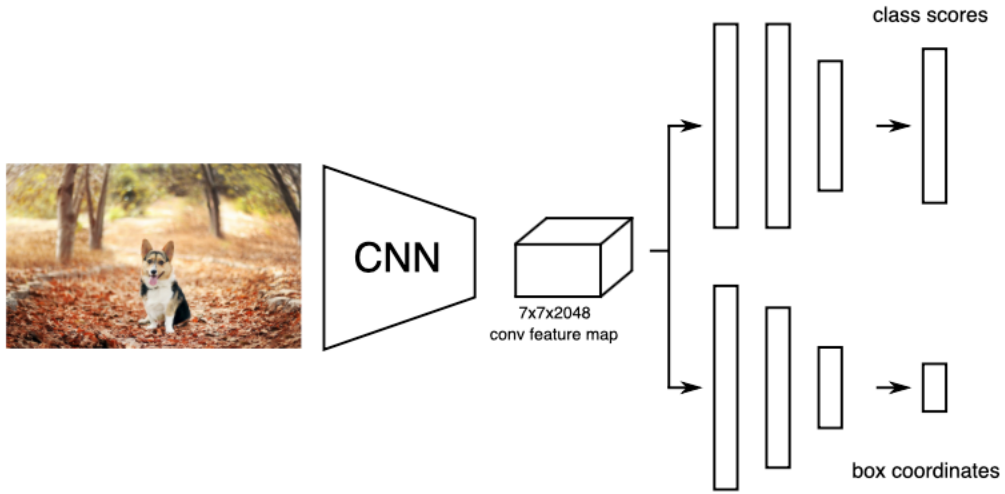


Localization as regression



Classification + Localization

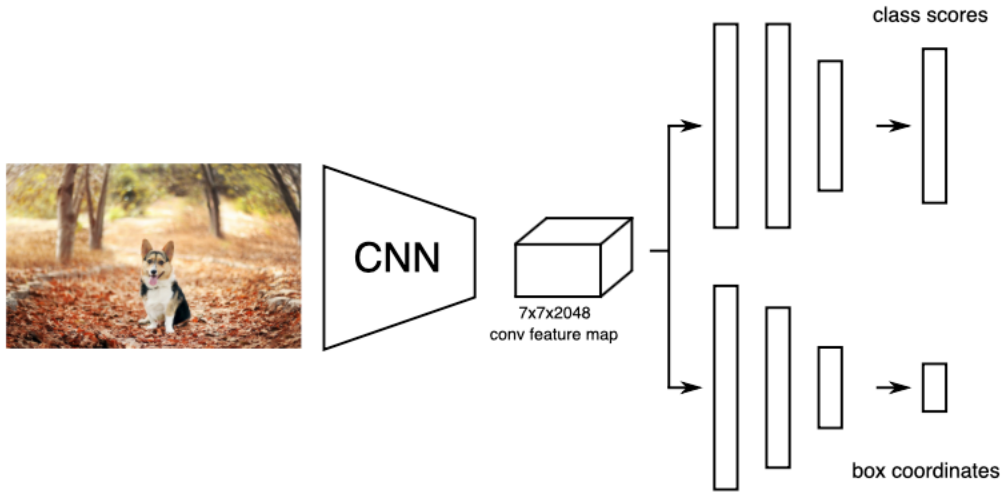
- Use a pre-trained CNN on ImageNet (ex. ResNet)
- The "localization head" is trained separately with regression
- Possible end-to-end fine tuning of both tasks
- At test time, use both heads



Classification + Localization

C classes, 4 output dimensions (1 box)

Predict exactly N objects: predict $(N \times 4)$ coordinates and $(N \times K)$ class scores



Object detection

We don't know in advance the number of objects in the image. Object detection relies on *object proposal* and *object classification*

Object proposal: find regions of interest (RoIs) in the image

Object classification: classify the object in these regions

Two main families:

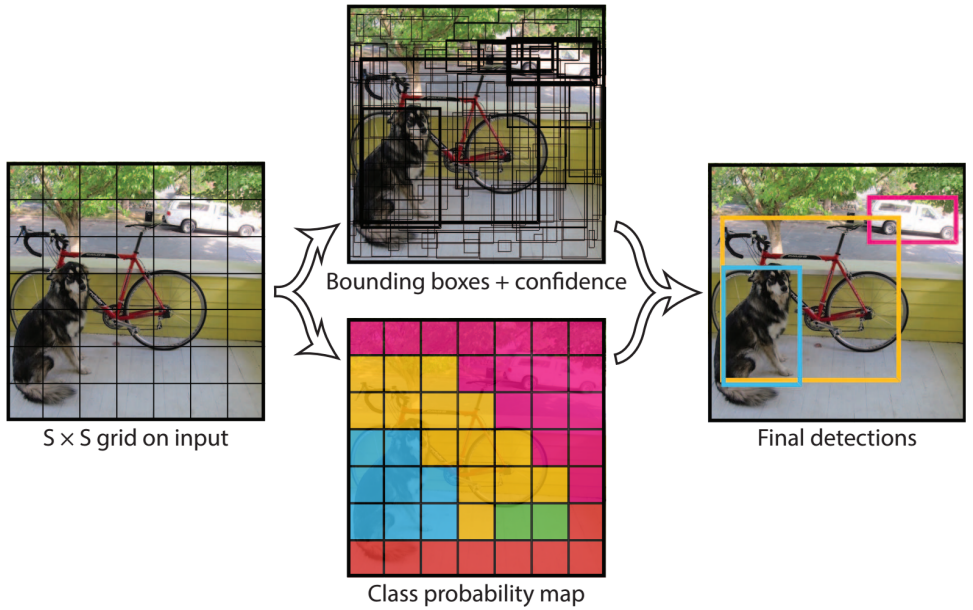
- Single-Stage: A grid in the image where each cell is a proposal (SSD, YOLO, RetinaNet)
- Two-Stage: Region proposal then classification (Faster-RCNN)

YOLO

For each cell of the $S \times S$ predict:

- B **boxes** and **confidence scores** C ($5 \times B$ values) + **classes** c

Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." CVPR (2016)

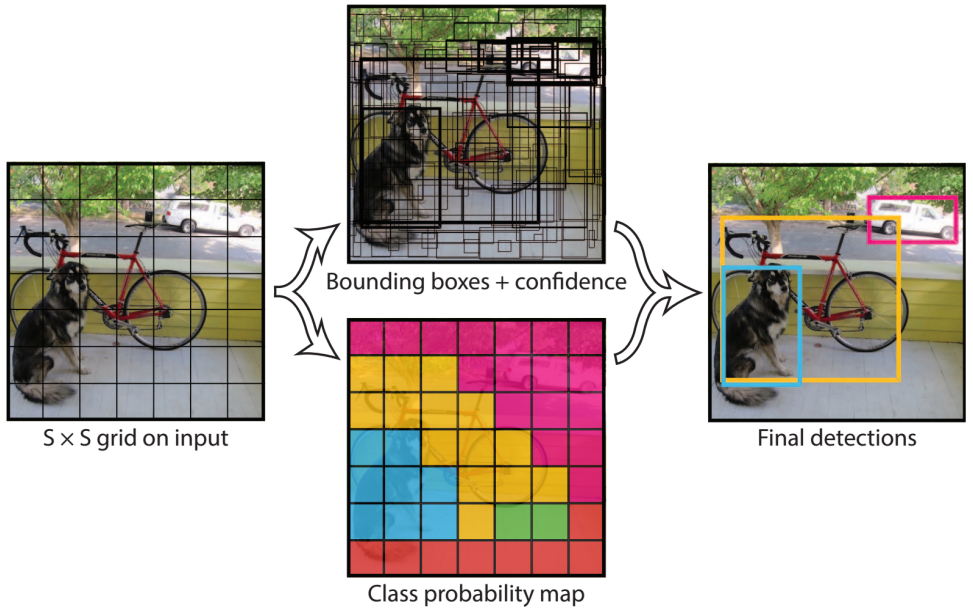


YOLO

Final detections:

$$C_j * prob(c) > \text{threshold}$$

Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." CVPR (2016)



YOLO

- After ImageNet pretraining, the whole network is trained end-to-end
- The loss is a weighted sum of different regressions

$$
\begin{aligned}
&\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
&+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \qquad (3)
\end{aligned}
$$

Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." CVPR (2016)

Box Proposals

Instead of having a predefined set of box proposals, find them on the image:

- **Selective Search** – from pixels (not learnt, no longer used)
- **Faster - RCNN** – Region Proposal Network (RPN)

Crop-and-resize operator (**RoI-Pooling**):

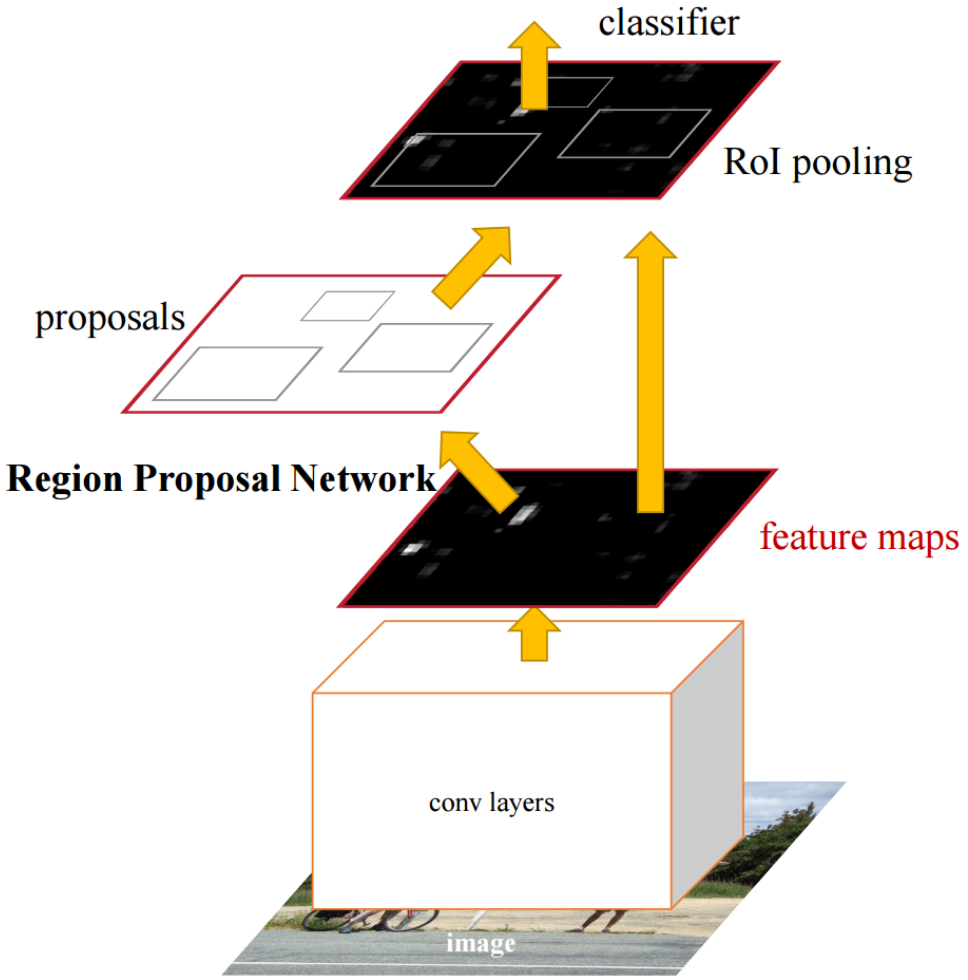
- Input: convolutional map + N regions of interest
- Output: tensor of $N \times 7 \times 7 \times \text{depth}$ boxes
- Allows to propagate gradient only on interesting regions, and efficient computation

Girshick, Ross, et al. "Fast r-cnn." ICCV 2015

Faster-RCNN

- Train jointly **RPN** and other head
- 200 box proposals, gradient propagated only in positive boxes
- Region proposal is translation invariant, compared to YOLO

Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." NIPS 2015



Measuring performance

method	test size shorter edge/max size	feature pyramid	align	mAP@[0.5:0.95]	AP_s	AP_m	AP_l
R-FCN [17]	600/1000			32.1	12.8	34.9	46.1
Faster R-CNN (2fc)	600/1000			30.3	9.9	32.2	47.4
Deformable [3]	600/1000		√	34.5	14.0	37.7	50.3
G-RMI [13]	600/1000			35.6	-	-	-
FPN [19]	800/1200	√		36.2	18.2	39.0	48.2
Mask R-CNN [7]	800/1200	√	√	38.2	20.1	41.1	50.2
RetinaNet [20]	800/1200	√		37.8	20.2	41.1	49.2
RetinaNet ms-train [20]	800/1200	√		39.1	21.8	42.7	50.2
Light head R-CNN	800/1200		√	**39.5**	21.8	43.0	50.7
Light head R-CNN ms-train	800/1200		√	**40.8**	22.7	44.3	52.8
Light head R-CNN	800/1200	√	√	**41.5**	25.2	45.3	53.1

Measures: mean Average Precision **mAP** at given **IoU** thresholds

- AP @0.5 for class "cat": average precision for the class, where $IoU(box^{pred}, box^{true}) > 0.5$

Zeming Li et al. Light-Head R-CNN: In Defense of Two-Stage Object Detector 2017

State-of-the-art

- Larger image sizes, larger and better models, better augmented data
- https://paperswithcode.com/sota/object-detection-on-coco

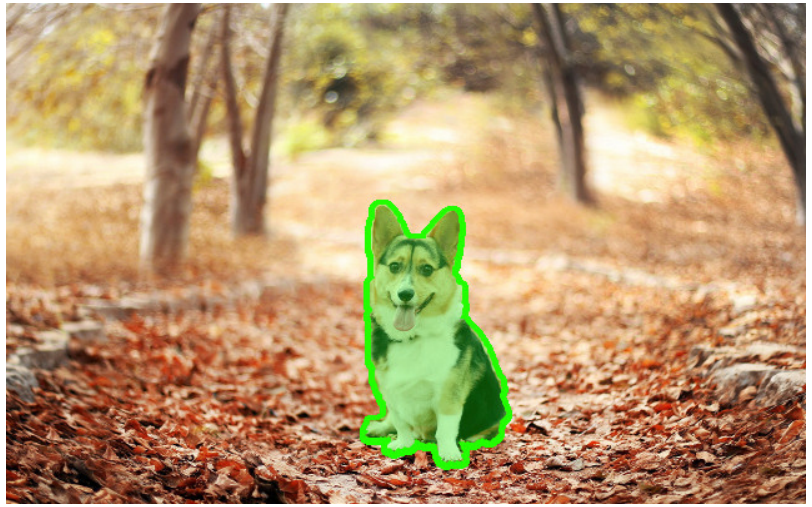
Model	FLOPs	# Params	AP_{val}	$AP_{test\text{-}dev}$
SpineNet-190 (1536) [11]	2076B	176.2M	52.2	52.5
DetectoRS ResNeXt-101-64x4d [43]	—	—	—	55.7†
SpineNet-190 (1280) [11]	1885B	164M	52.6	52.8
SpineNet-190 (1280) w/ self-training [71]	1885B	164M	54.2	54.3
EfficientDet-D7x (1536) [56]	410B	77M	54.4	55.1
YOLOv4-P7 (1536) [60]	—	—	—	55.8†
Cascade Eff-B7 NAS-FPN (1280)	1440B	185M	54.5	54.8
w/ Copy-Paste	1440B	185M	(+1.4) **55.9**	(+1.2) **56.0**
w/ self-training Copy-Paste	1440B	185M	(+2.5) **57.0**	(+2.5) **57.3**

Ghiasi G. et al. Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation, 2020

Segmentation

Segmentation

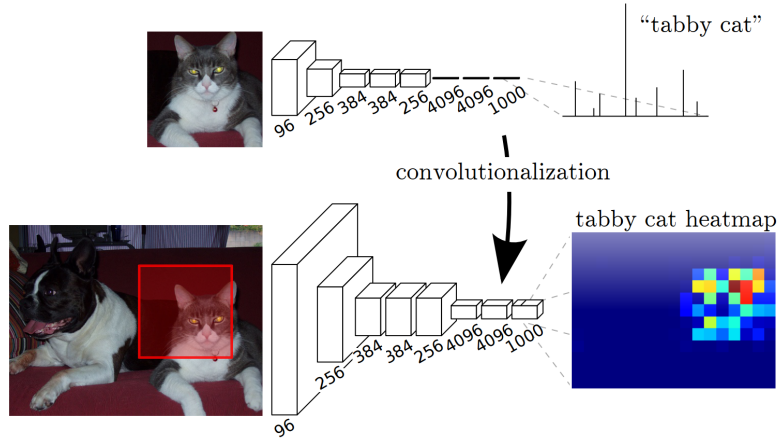
Output a class map for each pixel (here: dog vs background)



- **Instance segmentation**: specify each object instance as well (two dogs have different instances)
- This can be done through **object detection + segmentation**

Convolutionize

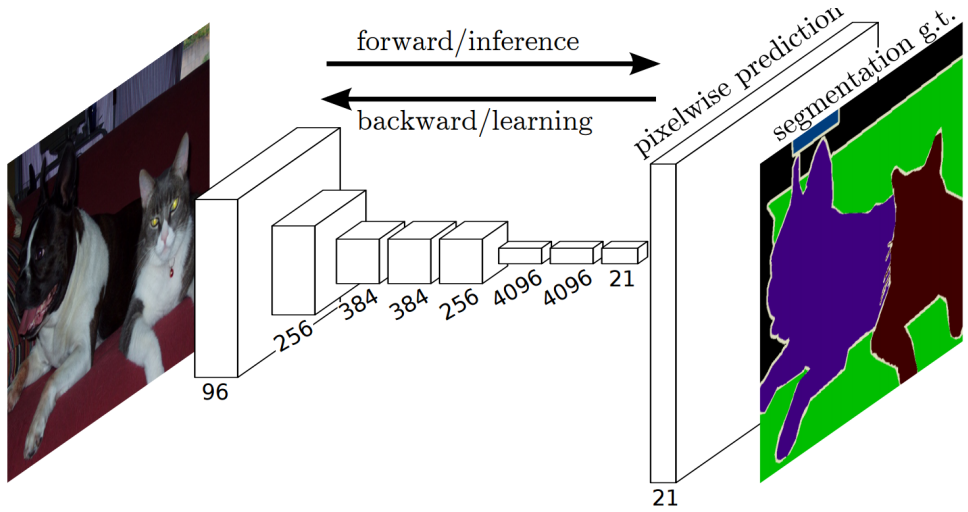
- Slide the network with an input of `(224, 224)` over a larger image. Output of varying spatial size
- **Convolutionize**: change Dense `(4096, 1000)` to 1×1 Convolution, with `4096, 1000` input and output channels
- Gives a coarse **segmentation** (no extra supervision)



`Long, Jonathan, et al. "Fully convolutional networks for semantic segmentation." CVPR 2015`

Fully Convolutional Network

- Predict / backpropagate for every output pixel
- Aggregate maps from several convolutions at different scales for more robust results

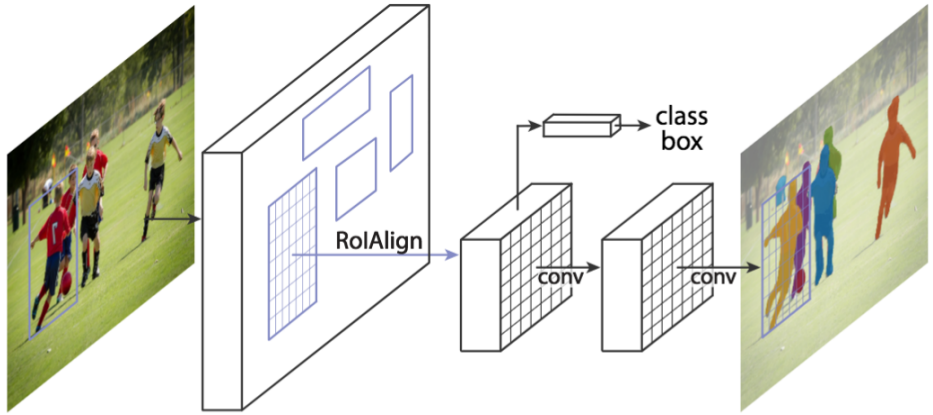


Long, Jonathan, et al. "Fully convolutional networks for semantic segmentation." CVPR 2015

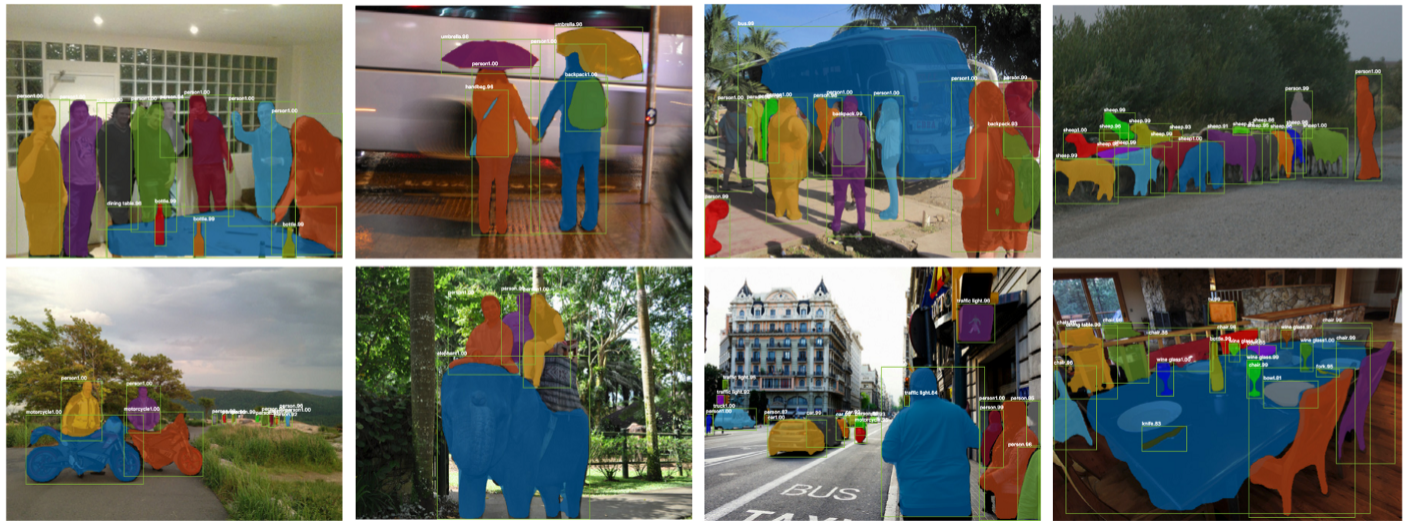
Mask-RCNN

Faster-RCNN architecture with a third, binary mask head

K. He and al. Mask Region-based Convolutional Network (Mask R-CNN) NIPS 2017



Results



- Mask results are still coarse (low mask resolution)
- Excellent instance generalization

`K. He and al. Mask Region-based Convolutional Network (Mask R-CNN) NIPS 2017`

Results

He, Kaiming, et al. "Mask r-cnn." Internal Conference on Computer Vision (ICCV), 2017.

State-of-the-art & links

Most benchmarks and recent architectures are reported here:

https://paperswithcode.com/area/computer-vision

Tensorflow

object detection API

Pytorch

Detectron https://github.com/facebookresearch/Detectron

- Mask-RCNN, Retina Net and other architectures
- Focal loss, Feature Pyramid Networks, etc.

Take away NN for Vision

Pre-trained features as a basis

- ImageNet: centered objects, very broad image domain
- 1M+ labels and many different classes resulting in very **general** and **disentangling** representations
- Better Networks (i.e. ResNet vs VGG) have **a huge impact**

Fine tuning

- Add new layers on top of convolutional or dense layer of CNNs
- **Fine tune** the whole architecture end-to-end
- Make use of a smaller dataset but with richer labels (bounding boxes, masks...)

Next: Lab 5!