KENTECH
Korea Institute of Energy Technology

Advanced Computer Vision
Week 13

Dec. 02, 2022

Seokju Lee

KENTECH
Korea Institute of Energy Technology

Final Presentation: Research Proposal (12/13, Tue)

Propose Your Own Research Goal (12/13, Tue)

Practice for **proposal & defense** progress of your **Ph.D. thesis**!

- **"Proposal"** mainly covers:

  Task definition + Data analysis + Baseline review + Your base solution

- 20 mins presentation (한국어) for each student
- **평가요소**

  1. Motivation + Reasons for choosing this topic?

     (why is it important?)
  2. Task definition (input & output)
  3. Review of previous related works (baseline)
  4. Your own solution (+ novel contributions)

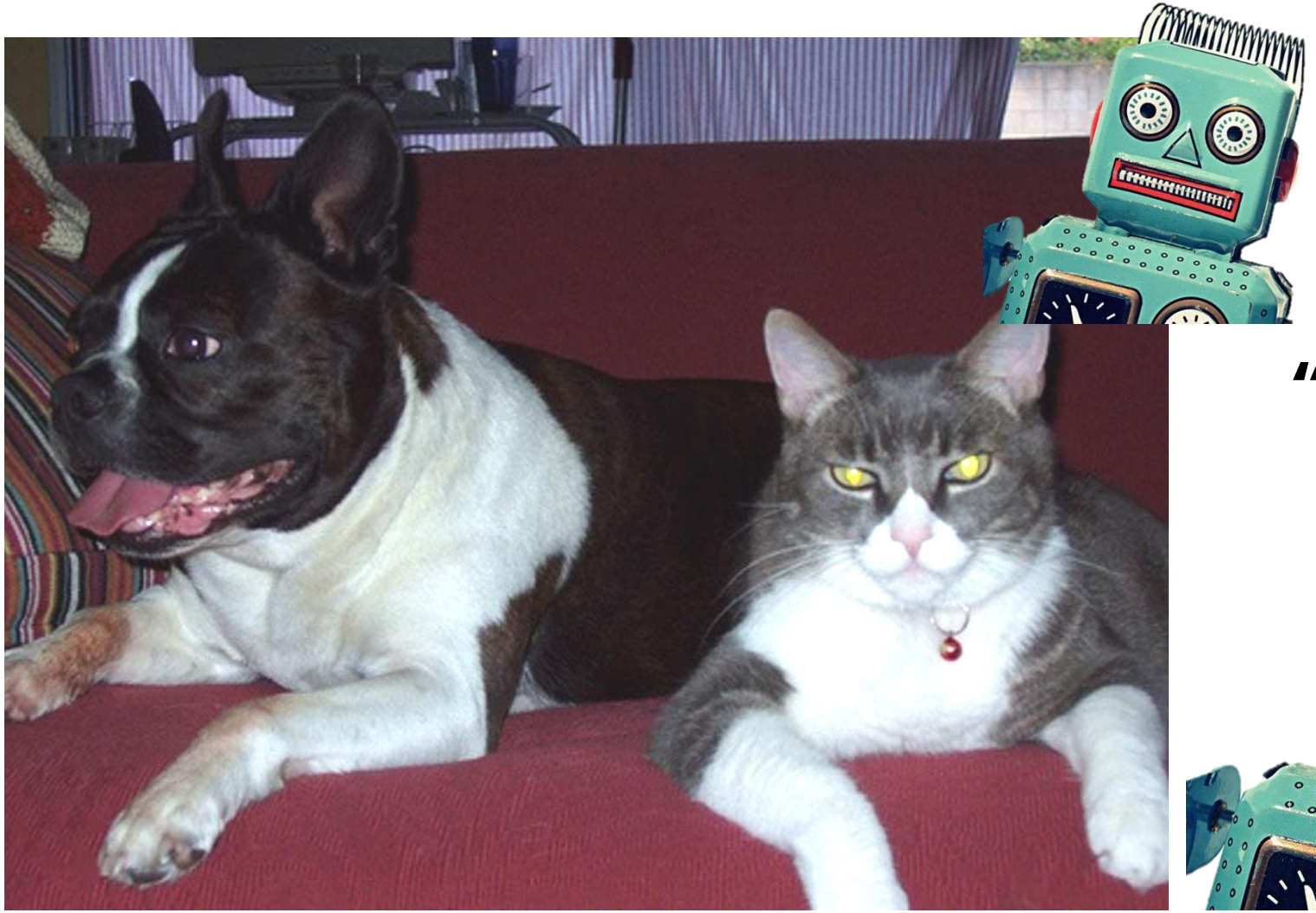
c.f.) Evaluation table for the paper review.

Cateogry	Sub-category (max 5 score each)
Background (10)	Why authors choose this topic?
	Overall motivations?
Purpose (5)	What are the authors trying to discover generally (top-down)?
Previous works (5)	Might have to be reviewed/discussed at the same time?
Methodology (10)	Novel contributions?
	Your own evaluation on them (will you accept it)?
Validation (25)	Which metrics?
	Quantitative evaluation & discussions?
	Qualitative evaluation & discussions?
	Dataset or benchmark analysis?
	Ablation study?
Discussion (25)	What did the authors address from this study (analysis in detail)?
	Generalization?
	Limitations of the work or failure cases?
	Future works?
	Possible applications?
발표퀄리티 (10)	전달력, 발표력, 질의 응답, 태도 등 (10)

KENTECH
Korea Institute of Energy Technology

Object Detection

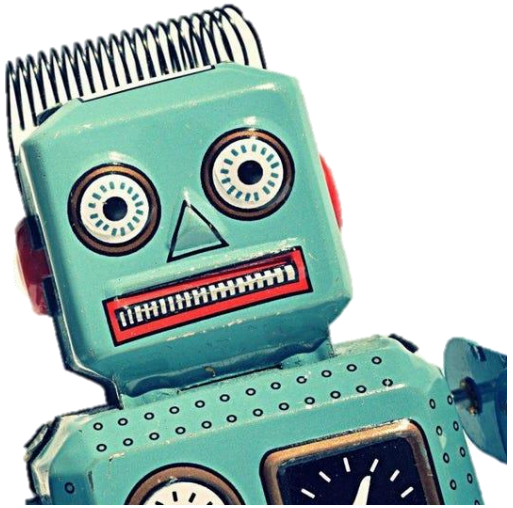
Limitation of Image Classification: Dog or Cat?

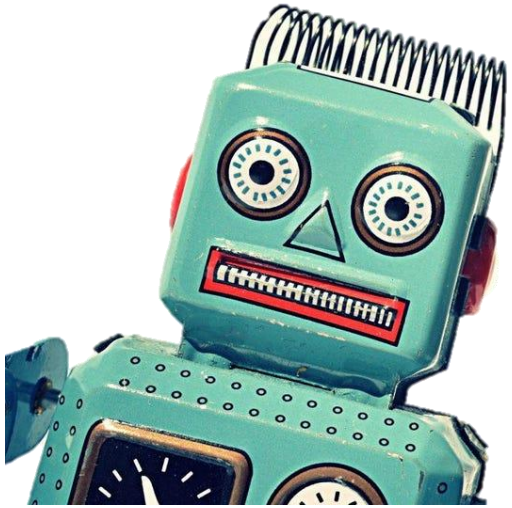


"Sofa?"

"Dog?"

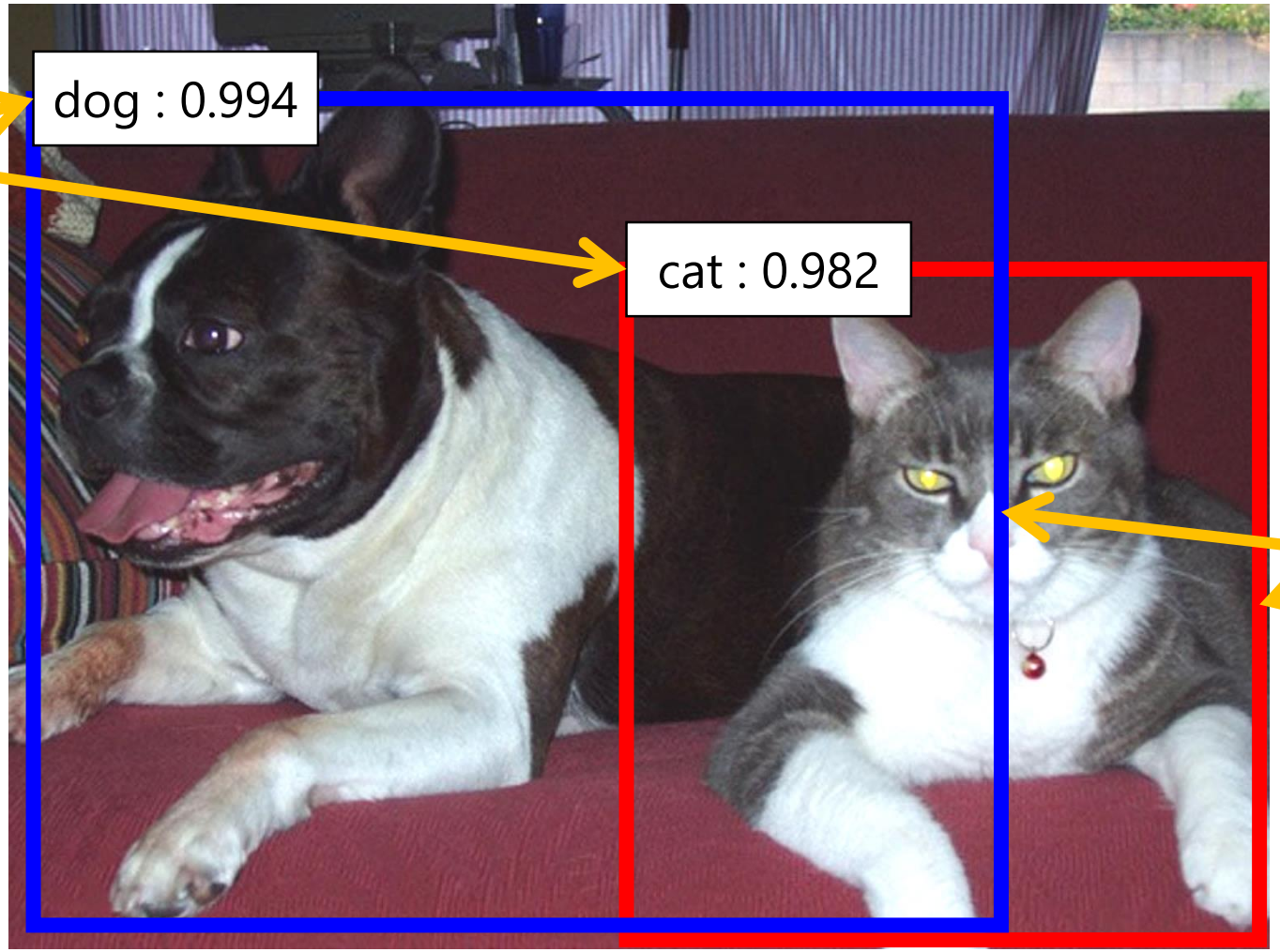
"Cat?"





[1] S. Ren, et al., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." *NIPS* (2015).

Object Detection: What and Where?

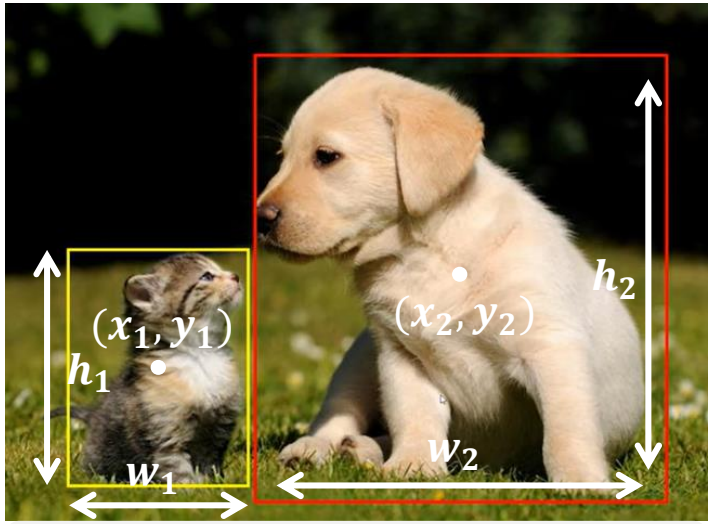


[1] S. Ren, et al., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." *NIPS* (2015).

Object Detection: Input & Output

: Task of assigning **labels** & **bounding boxes** to all objects in the image.

Classify **which class** the object belongs to.



Find the coordinates of the bounding box **where** the object is located, and classify **which class** it belongs to.

Images	Class (=label)
I_1	cat
I_2	cat
I_3	dog

Classification

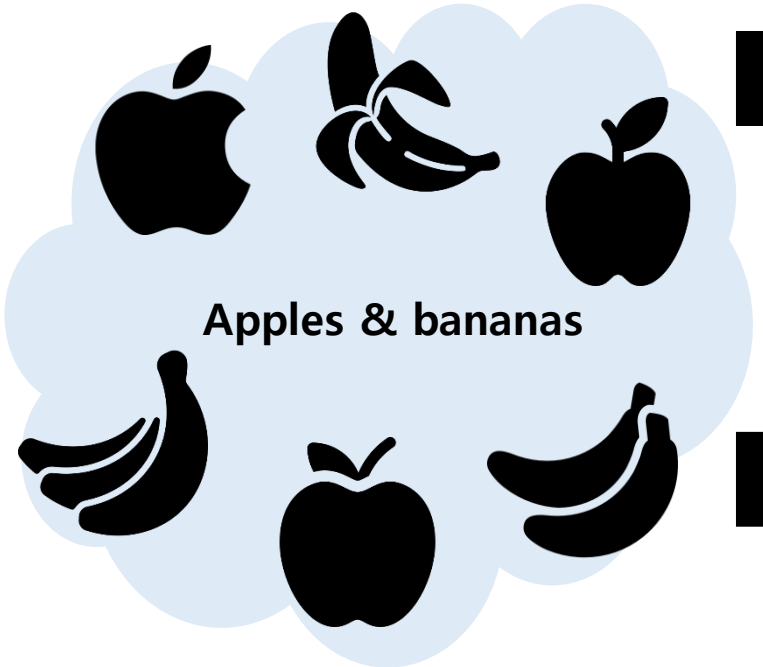
Images	Class (=label)	x	y	w	h
I_1	cat	60	210	100	180
I_1	dog	200	50	340	360
I_2	car	46	250	100	80

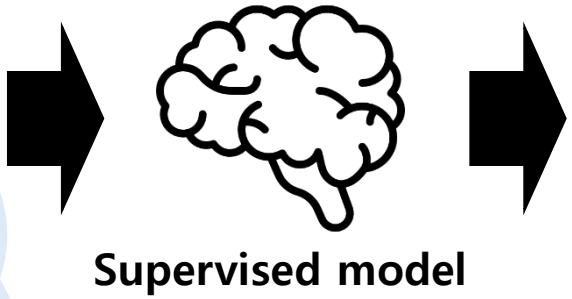
Classification + Regression

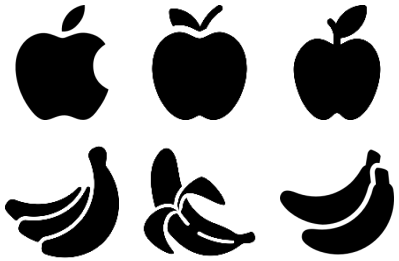
*Back to Basics: Supervised & Unsupervised Learning

Let's review – **Two** major approaches of machine learning?

1) **Supervised learning** requires **labels** for training
2) **Unsupervised learning** does **not** require **labels** for training

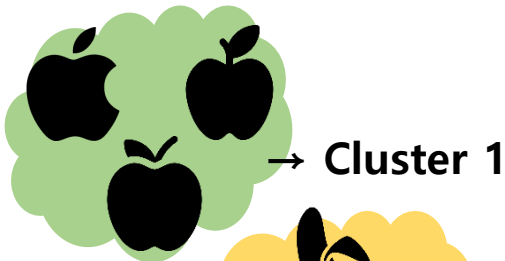




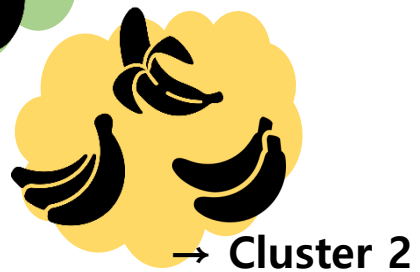


= "Apple"

= "Banana"

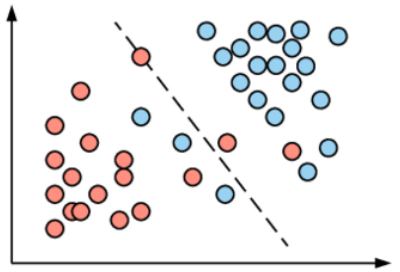


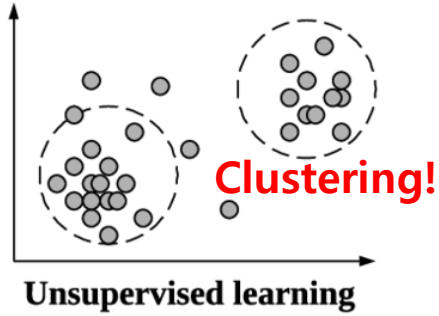
→ Cluster 1



→ Cluster 2

→ **Mathematical representation :**



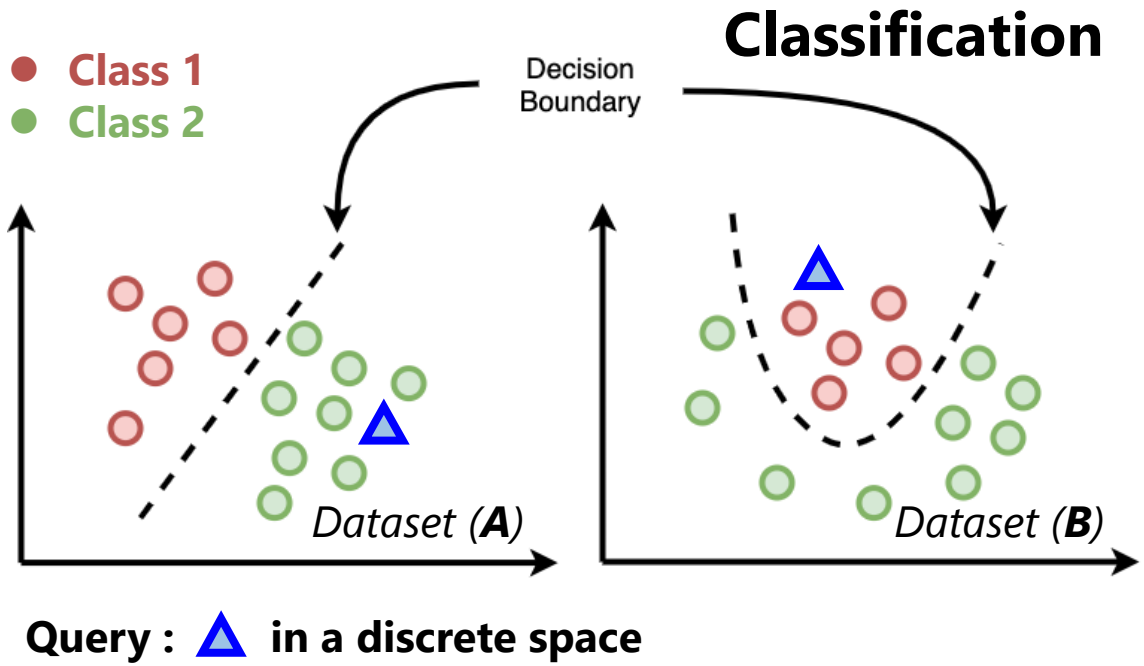


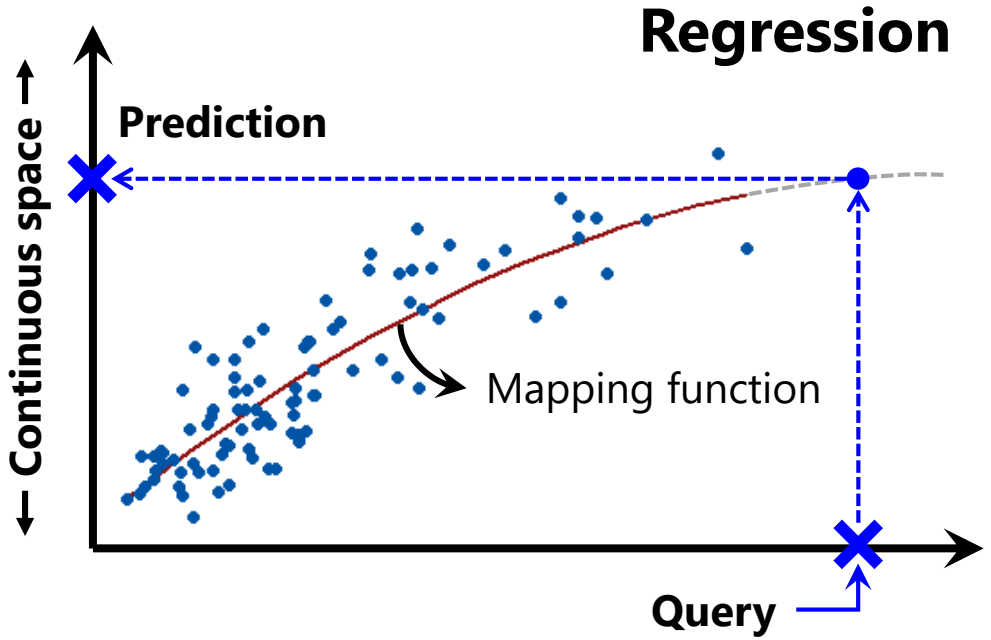
↑ Images from edX (IBM ML0101EN)

*Back to Basics: Classification & Regression

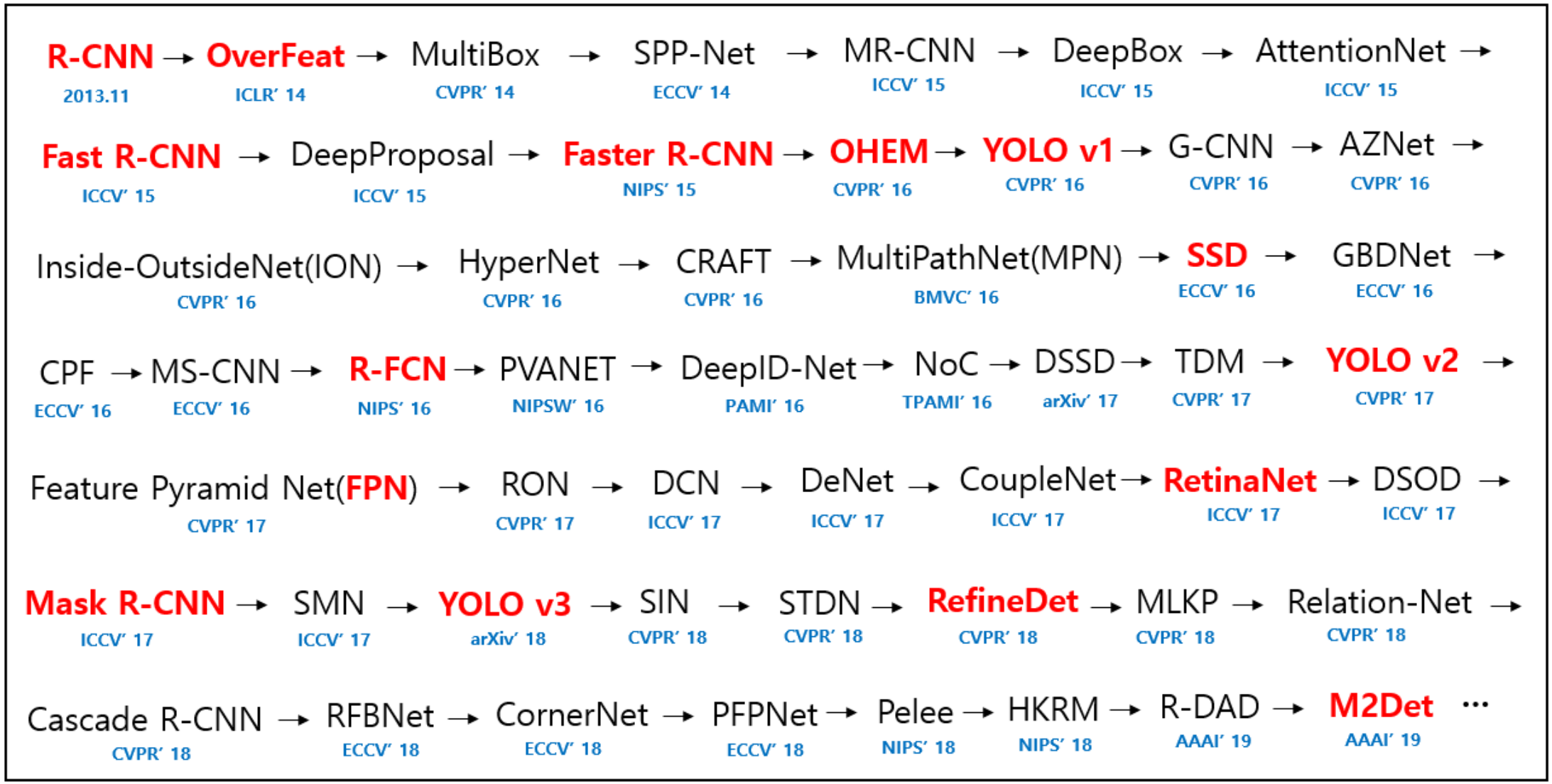
Supervised learning problem can be represented into **two tasks**.

1) **Classification** is the task of predicting a **discrete** class label.
2) **Regression** is the task of predicting a **continuous** quantity.





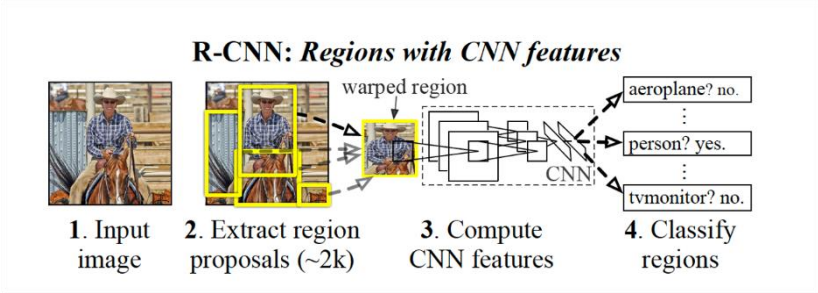
Object Detection: How to Design Models?



https://github.com/hoya012/deep_learning_object_detection

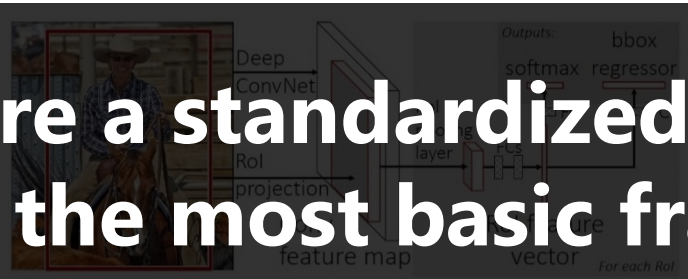
Object Detection: How to Design Models?

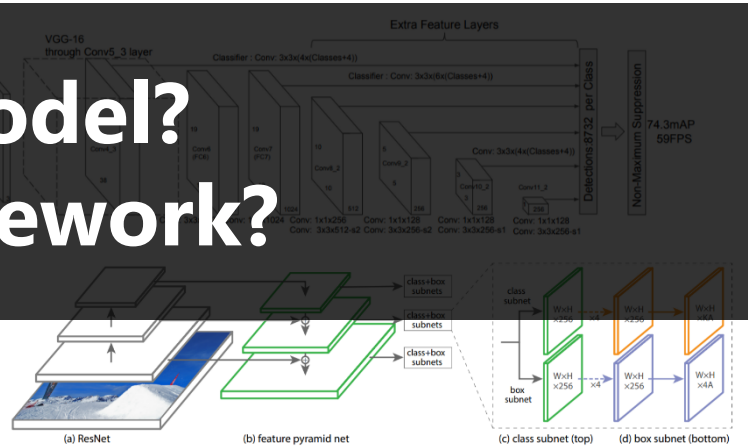
- R-CNN [Girshick et al. 2014]
- SPP-net [He et al. 2014]
- Fast R-CNN [Girshick. 2015]
- Faster R-CNN [Ren et al. 2015]
- R-FCN [Dai et al. 2016]
- YOLO [Redmon et al. 2016]
- SSD [Liu et al. 2016]
- Feature Pyramid Networks + Faster R-CNN [Lin et al. 2017]
- RetinaNet [Lin et al. 2017]
- Training with Large Minibatches (MegDet) [Peng, Xiao, Li, et al. 2017]
- Cascade R-CNN [Cai & Vasconcelos 2018]
- DETR [Carion et al. 2020]
- …

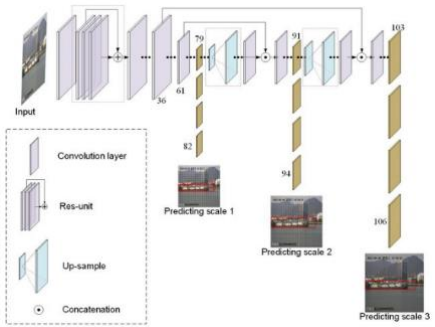


→ **Is there a standardized model?**

→ **What is the most basic framework?**







Object Detection: What is the Meta-Architecture?

Meta- means?

: High-level or symbolic abstraction.

Meta-architecture means?

: Abstract of deep neural architecture. Representative/generic structure of networks.

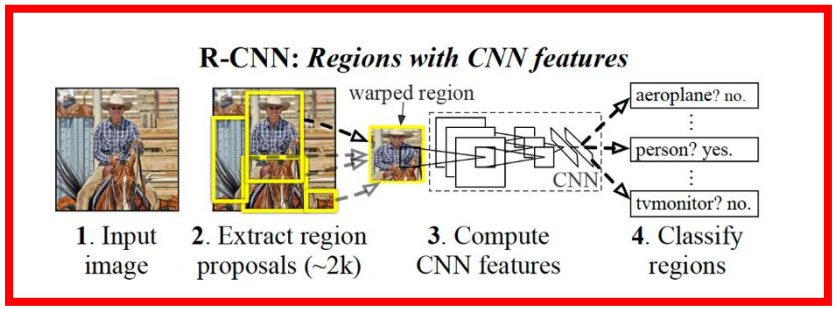
Meta-architecture for object detection?

→ Two-stage detector

→ One-stage detector

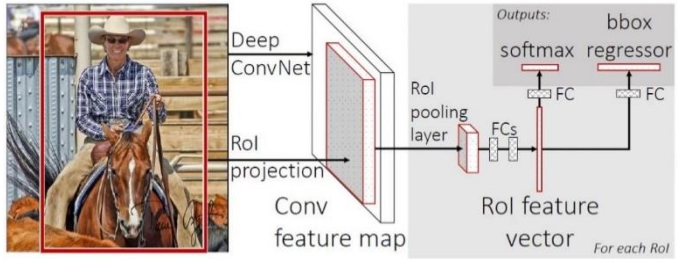
Object Detection: Two-Stage vs. One-Stage

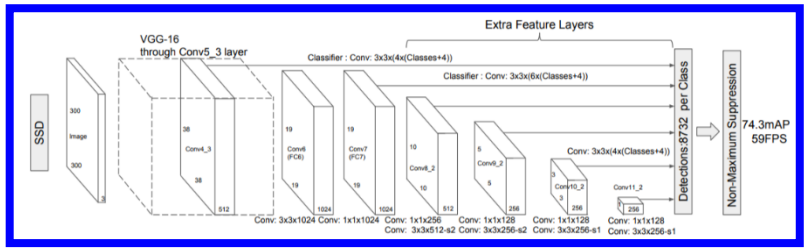
- **R-CNN [Girshick et al. 2014]**
- SPP-net [He et al. 2014]
- Fast R-CNN [Girshick. 2015]
- Faster R-CNN [Ren et al. 2015]
- R-FCN [Dai et al. 2016]
- YOLO [Redmon et al. 2016]
- **SSD [Liu et al. 2016]**
- Feature Pyramid Networks + Faster R-CNN [Lin et al. 2017]
- RetinaNet [Lin et al. 2017]
- Training with Large Minibatches (MegDet) [Peng, Xiao, Li, et al. 2017]
- Cascade R-CNN [Cai & Vasconcelos 2018]
- DETR [Carion et al. 2020]
- …

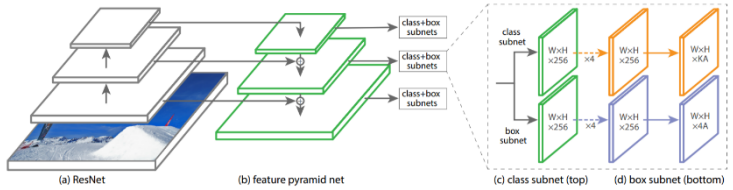


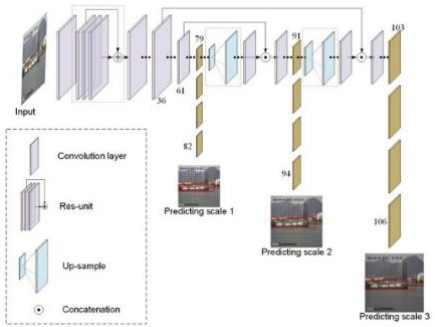
(1) Two-stage detector

(2) One-stage detector









Two-Stage Detector: R-CNN Series

References

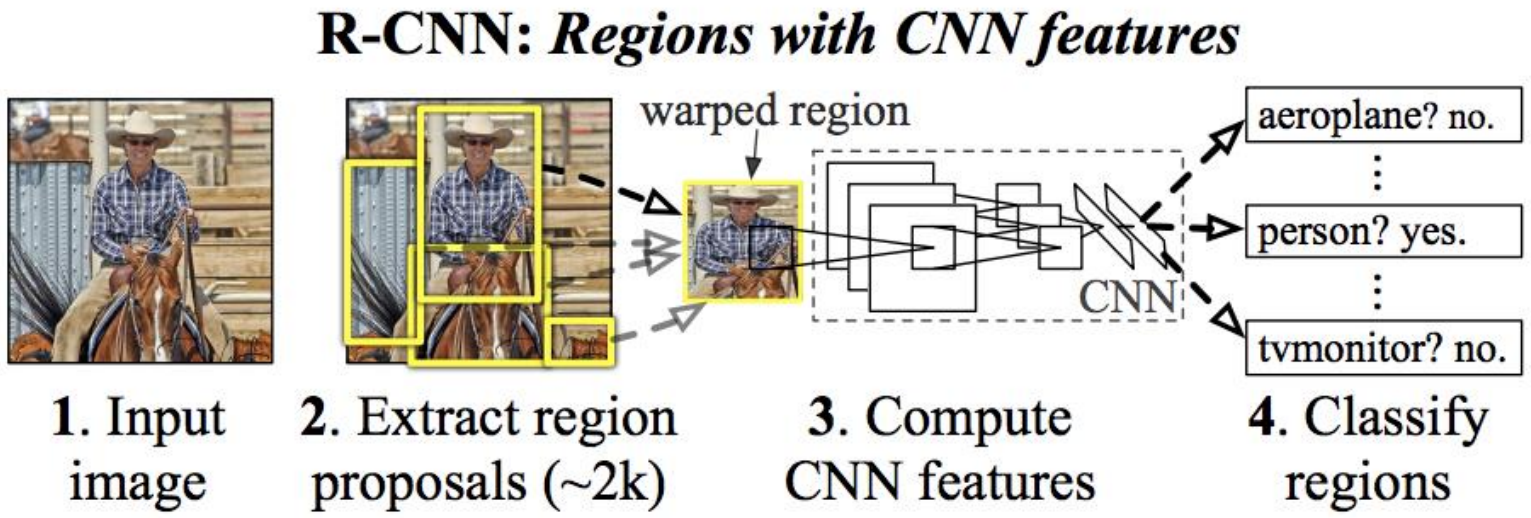
- B. Alexe, T. Deselaers, and V. Ferrari. "**Measuring the objectness of image windows.**" *T-PAMI*, 2012.
- I. Endres and D. Hoiem. "**Category independent object proposals.**" *ECCV*, 2010.
- J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. "**Selective search for object recognition.**" *IJCV*, 2013.
- A. Krizhevsky, I. Sutskever, and G. Hinton. "**ImageNet classification with deep convolutional neural networks.**" *NIPS*, 2012.
- X. Wang, M. Yang, S. Zhu, and Y. Lin. "**Regionlets for generic object detection.**" *ICCV*, 2013.
- R. Girshick, J. Donahue, T. Darrell, and J. Malik. "**Rich feature hierarchies for accurate object detection and semantic segmentation.**" *CVPR*, 2014.

R-CNN: Region-Based Convolutional Neural Networks

Key idea

→ Exploit strong **CNN representation** for **accurate** region classification

Hybrid approach: Traditional region proposal + CNN feature extraction



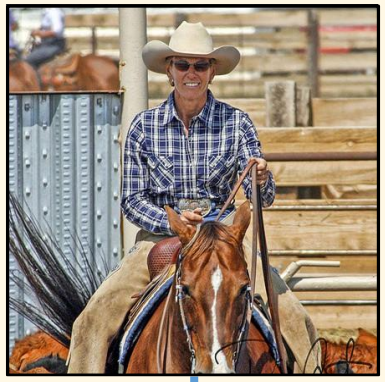
[1] R. Girshick, et al., "Rich feature hierarchies for accurate object detection and semantic segmentation." *CVPR* (2014).

R-CNN: Region-Based Convolutional Neural Networks

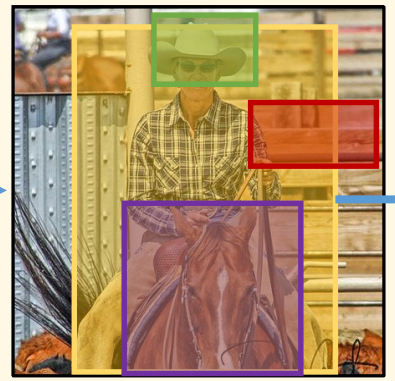
Per-image computation

Per-region computation for each $r_i \in r(I)$

I:



Selective search



Crop & warp

ConvNet(r_i)

1-vs-rest SVMs

Box regressor

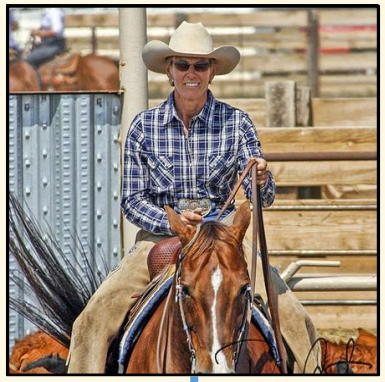
Figure credit: Ross Girshick

R-CNN: Region-Based Convolutional Neural Networks

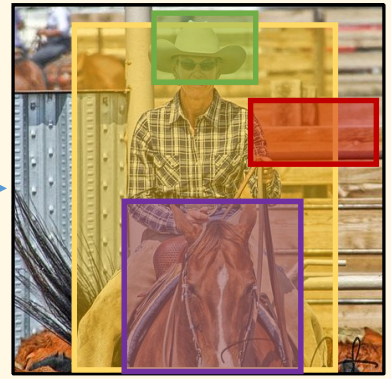
Per-image computation

Per-region computation for each $r_i \in r(I)$

I:



Selective search



1

Use an off-the-shelf region/object/detection proposal algorithm (~2k proposals per image)

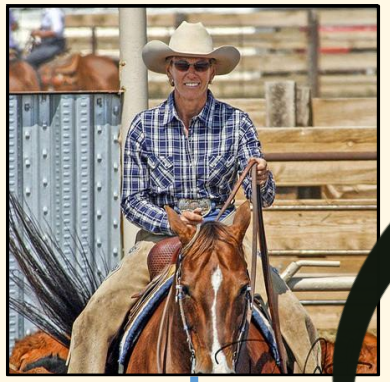
Figure credit: Ross Girshick

R-CNN: Region-Based Convolutional Neural Networks

Per-image computation

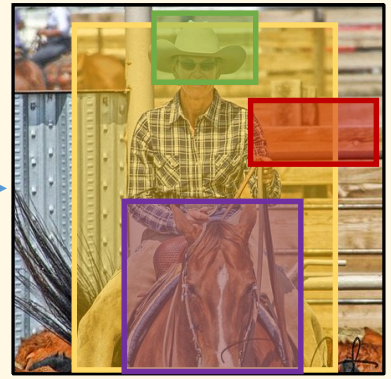
Per-region computation for each $r_i \in r(I)$

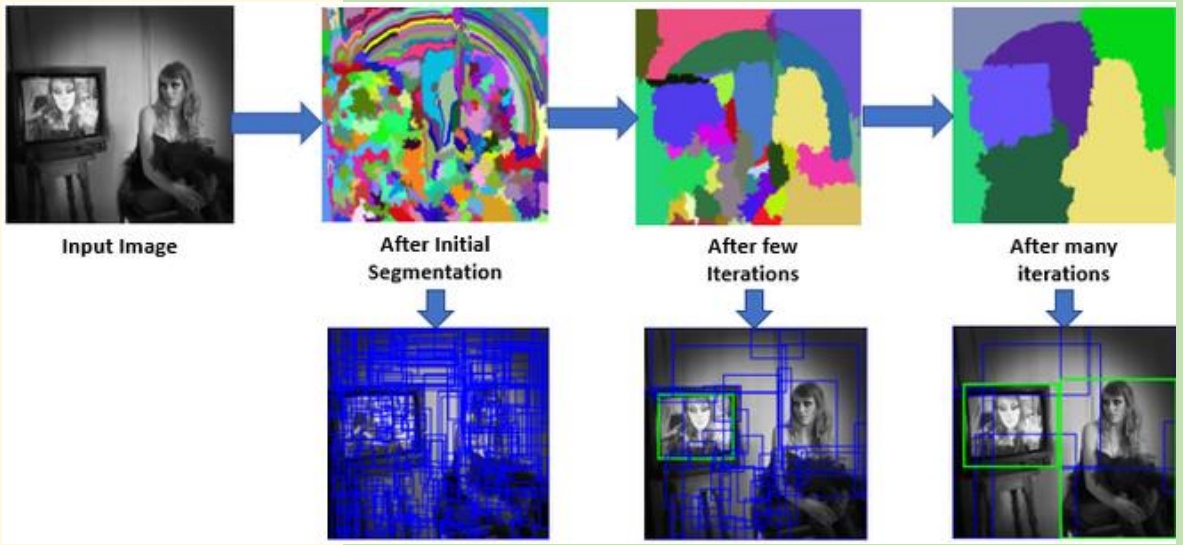
I:



Selective search

1





Selective search?
→ Region proposal based on different **similarity** metrics. (e.g., color, texture, size, and shape)

Region-of-Interest (RoI)

Use an off-the-shelf region/object/detection proposal algorithm (~2k proposals per image)

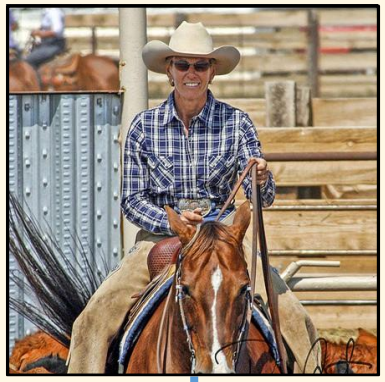
Figure credit: Ross Girshick

R-CNN: Region-Based Convolutional Neural Networks

Per-image computation

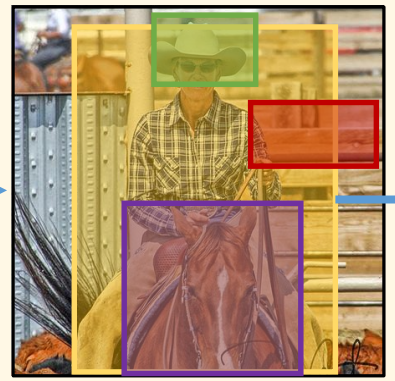
Per-region computation for each $r_i \in r(I)$

I:



Selective search

1



Crop & warp

2

Crop and warp each proposal image window to obtain a fixed-size network input

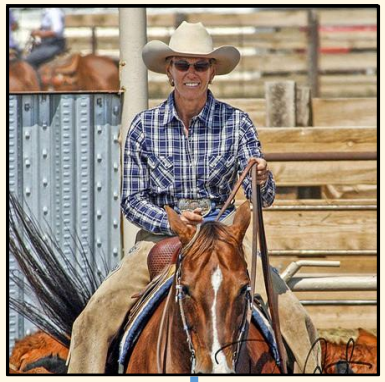
Figure credit: Ross Girshick

R-CNN: Region-Based Convolutional Neural Networks

Per-image computation

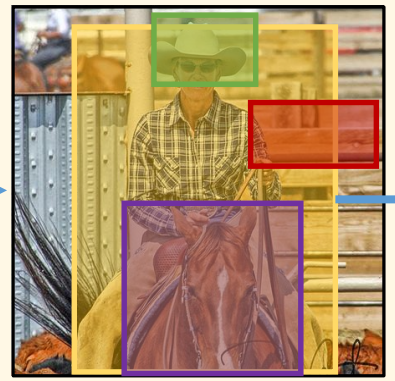
Per-region computation for each $r_i \in r(I)$

I:



Selective search

1



Crop & warp

2

ConvNet(r_i)

3

Forward propagate the fixed-size network input to get a feature representation

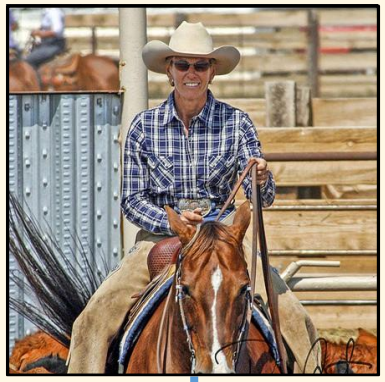
Figure credit: Ross Girshick

R-CNN: Region-Based Convolutional Neural Networks

Per-image computation

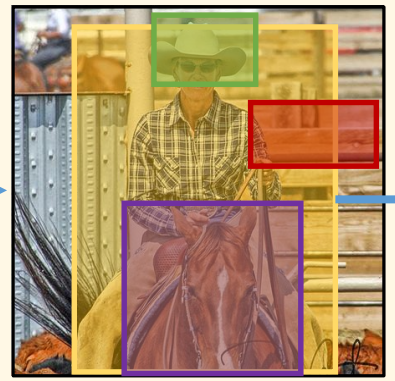
Per-region computation for each $r_i \in r(I)$

I:



Selective search

1



Crop & warp

2

ConvNet(r_i)

3

Classification

4

Object classification

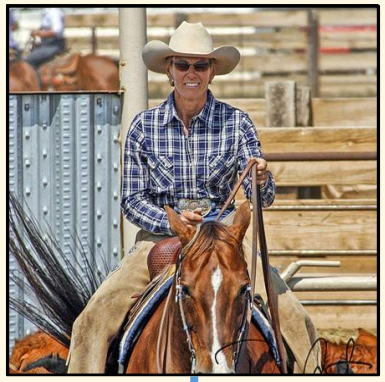
Figure credit: Ross Girshick

R-CNN: Region-Based Convolutional Neural Networks

Per-image computation

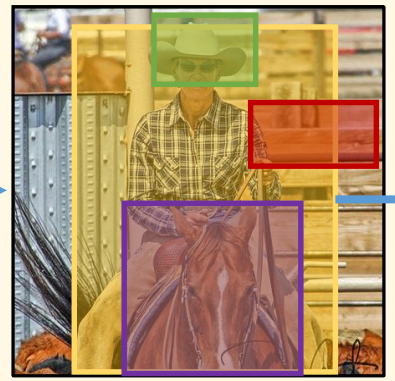
Per-region computation for each $r_i \in r(I)$

I:



Selective search

1



Crop & warp

2

ConvNet(r_i)

3

Classification

4

Box regressor

5

Refine proposal localization with bounding-box regression

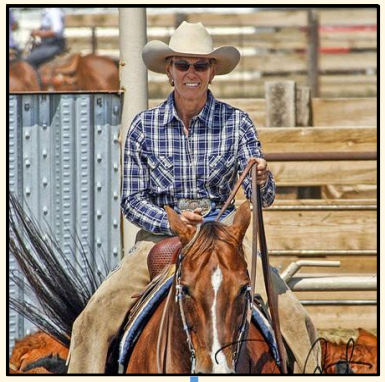
Figure credit: Ross Girshick

The Problem with R-CNN: "Slow"

Per-image computation

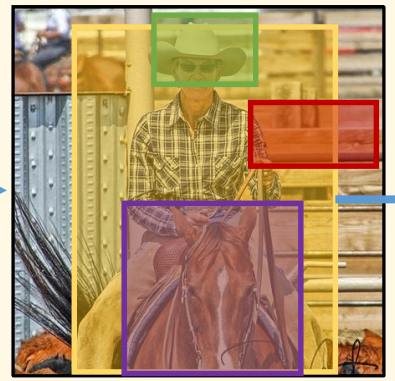
Per-region computation for each $r_i \in r(I)$

I:



Selective search

1



Crop & warp

2

ConvNet(r_i)

3

Classification

4

Box regressor

5

~ 50 secs for one image

Very heavy *per-region* computation
E.g., 2,000 full network iterations

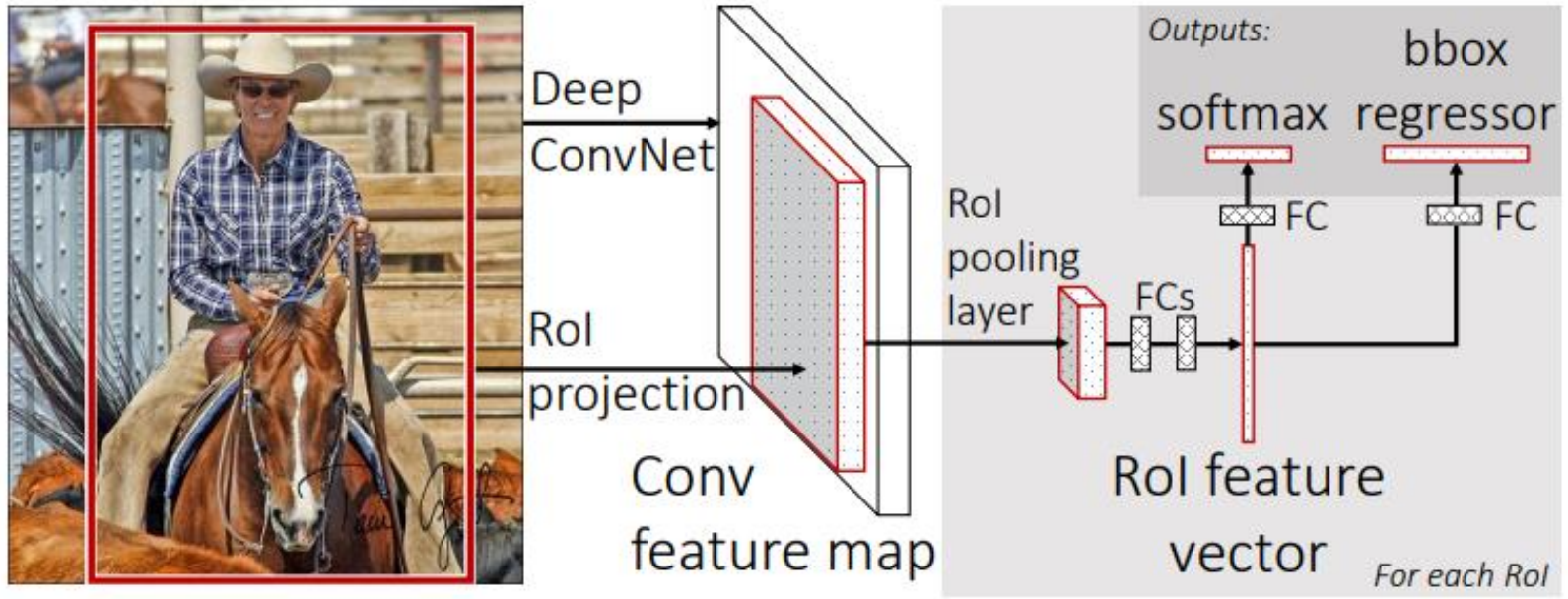
Figure credit: Ross Girshick

Solution: Fast R-CNN

Key idea

→ **Sharing** heavy per-region computation for **fast** inference (2 *fps*, x100 times to the R-CNN)

Region proposal + CNN feature extraction for the entire input image in the first stage



[1] R. Girshick, et al., "Fast R-CNN." *ICCV* (2015).

One-Stage Detectors

Key idea

→ Object detection from **single feedforward** of CNN

→ Eliminate proposal generation & per-region feature resampling and classification

→ Extremely fast (40~90 FPS with single GPU)

References

- YOLO: You Only Look Once [Redmon et al., CVPR'16]
- YOLO-v2 [Redmon et al., CVPR'17]
- YOLO-v3 [Redmon et al., arXiv'18]
- **SSD [Liu et al., ECCV'16]**
- RetinaNet [Lin et al., ICCV'17]

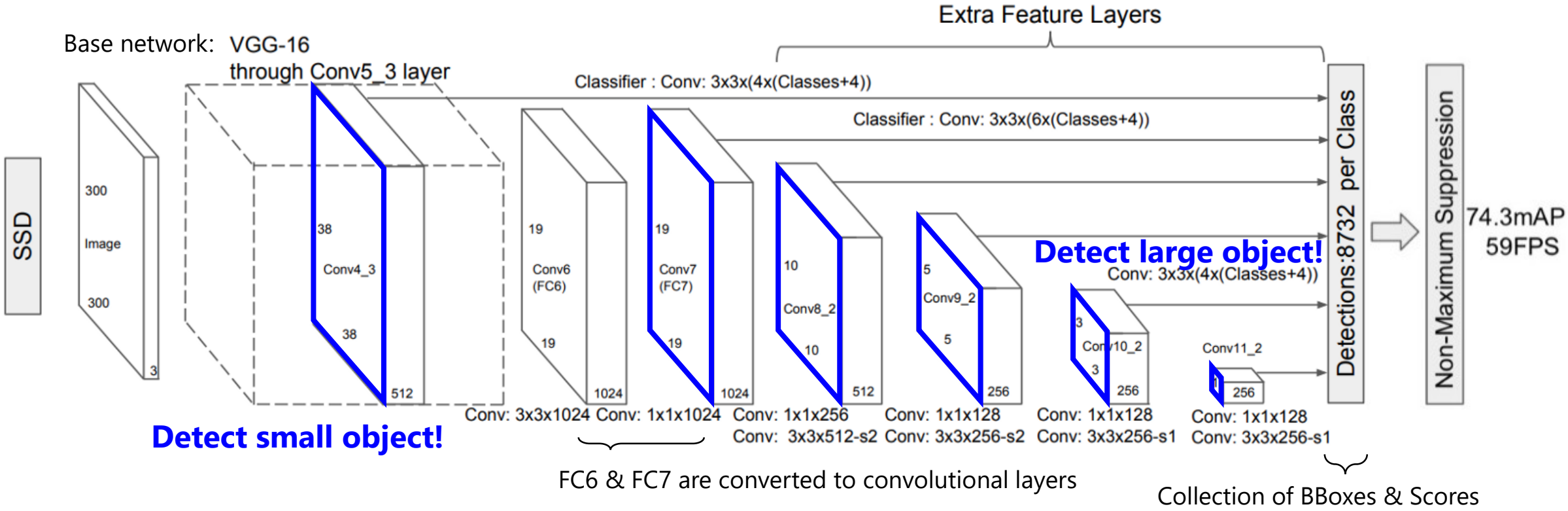
SSD: Single Shot Multibox Detector

Key idea

→ Fully convolutional layers to extract **spatial** features.

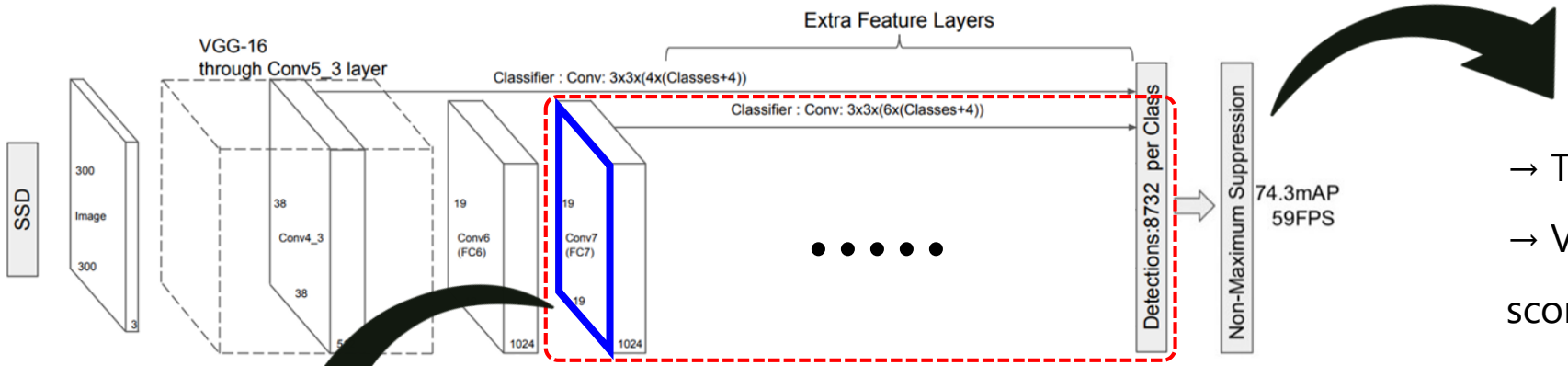
→ Aggregates **multi-scale** spatial features.

→ Meta-architecture for one-stage detector.



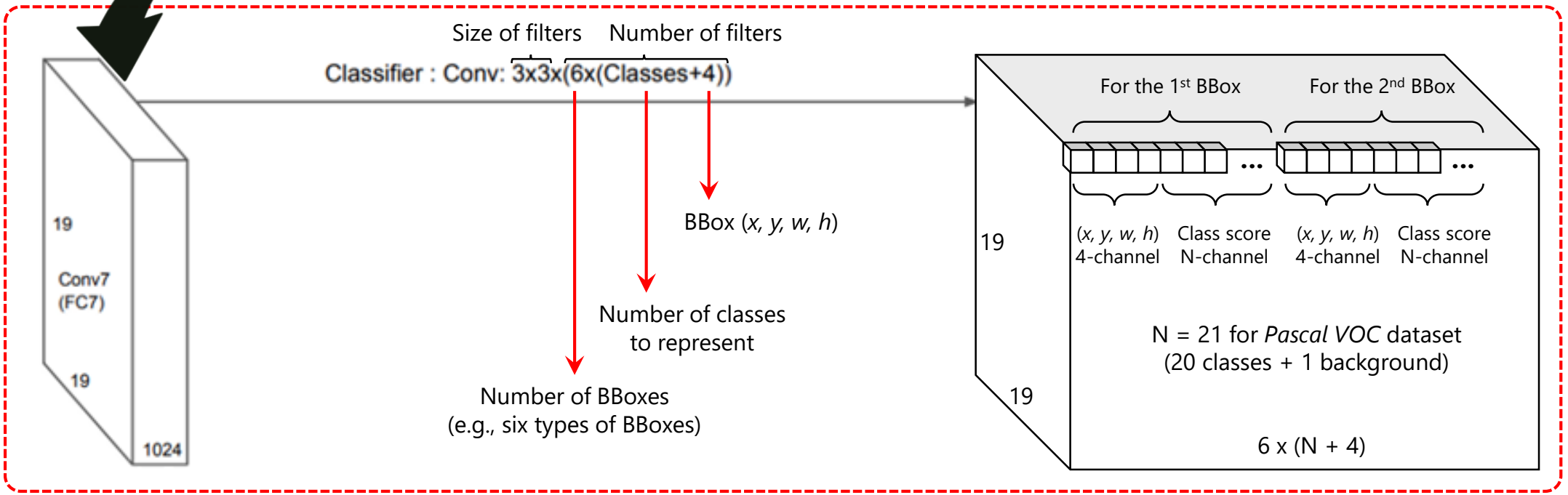
[1] Liu, et al., "Ssd: Single shot multibox detector." *ECCV* (2016).

SSD: Single Shot Multibox Detector



→ Total 8,732 BBoxes for each class

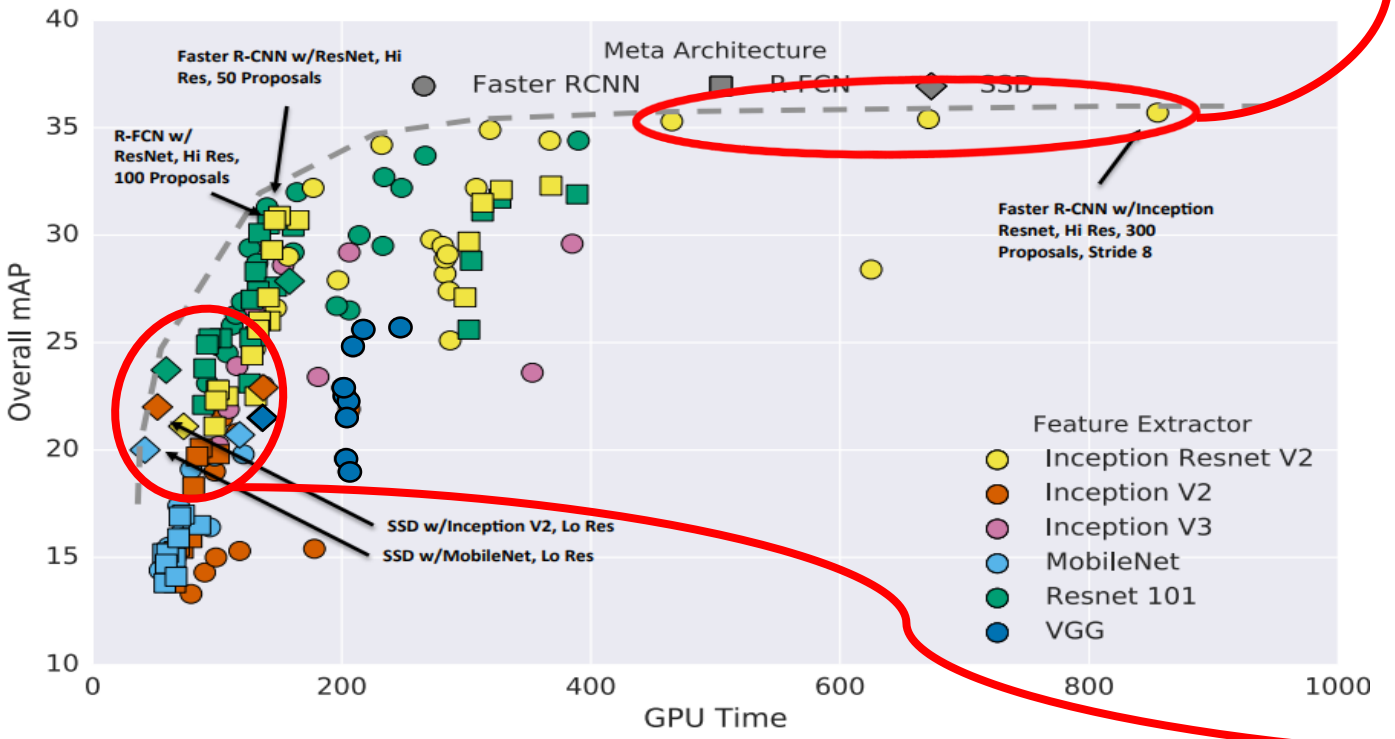
→ Visualize the BBoxes with the class score over 0.5 (threshold)

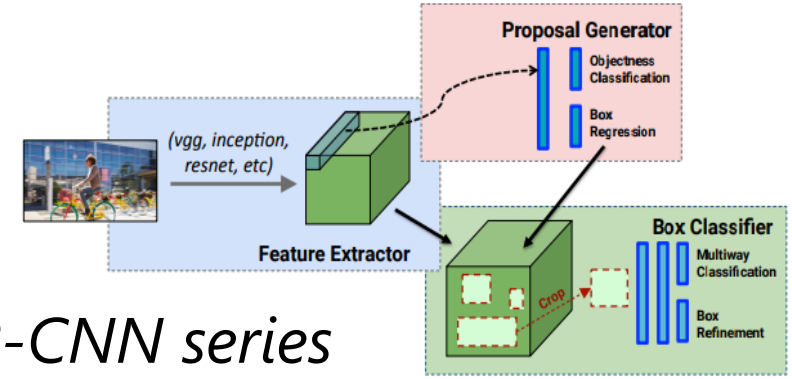


[1] Liu, et al., "Ssd: Single shot multibox detector." *ECCV* (2016).

Summary: What is the Best Model?

Speed & accuracy trade-offs





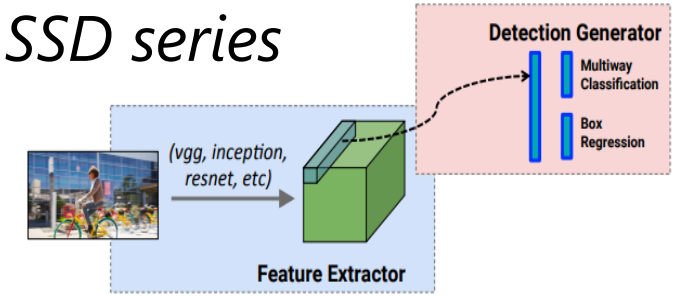
R-CNN series

Two-stage detectors

- Complex & Slow (~5FPS)
- More accurate

---------- vs. ----------

SSD series



One-stage detectors

- Simple & Fast (~55FPS)
- Less accurate

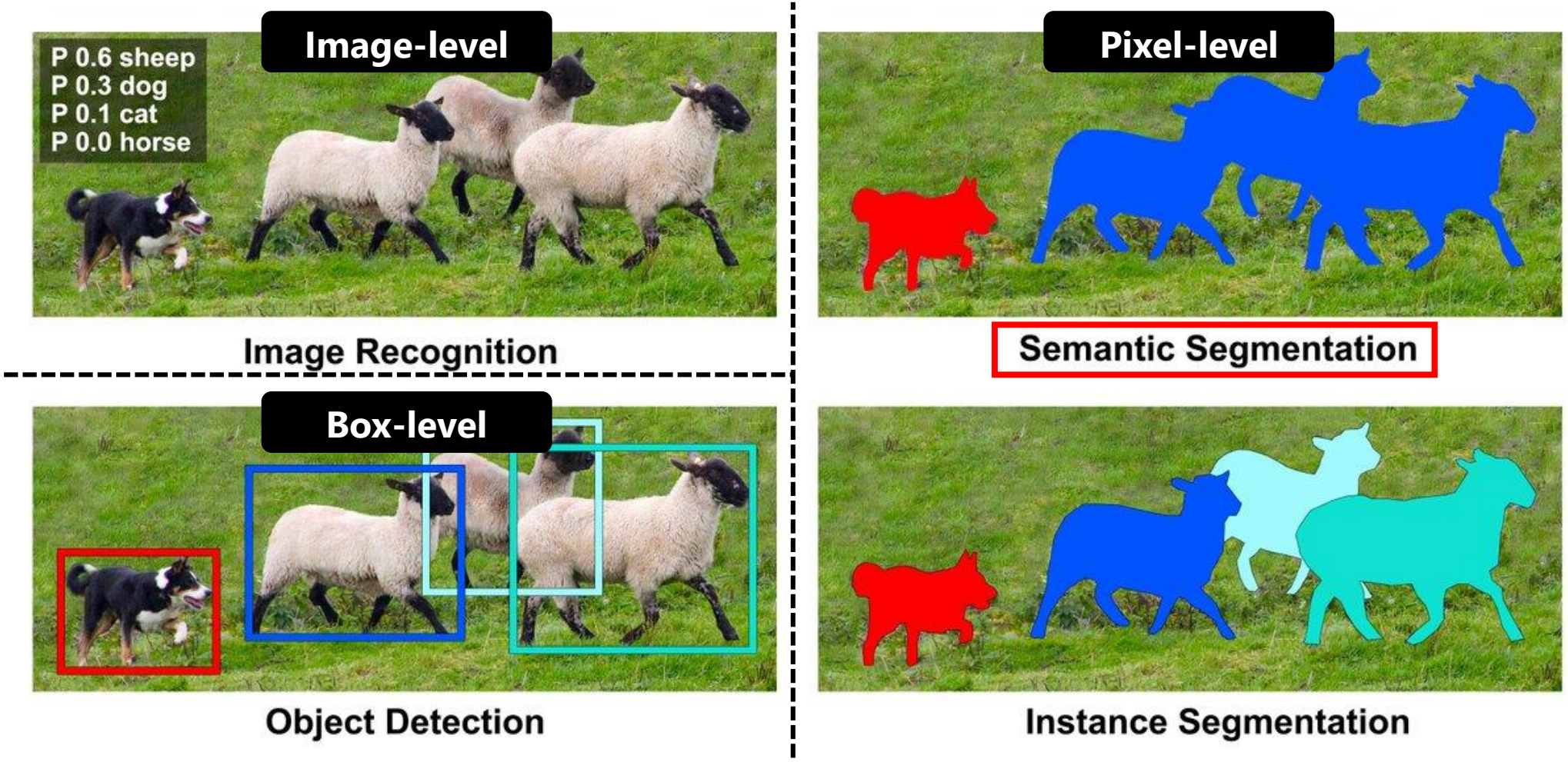
→ A lot more creative and high-performance architectures have been released so far, and they resemble **human** cognitive processes!

[1] Speed/accuracy trade-offs for modern convolutional object detectors. Huang et al.

KENTECH
Korea Institute of Energy Technology

Semantic Segmentation

Computer Vision Tasks



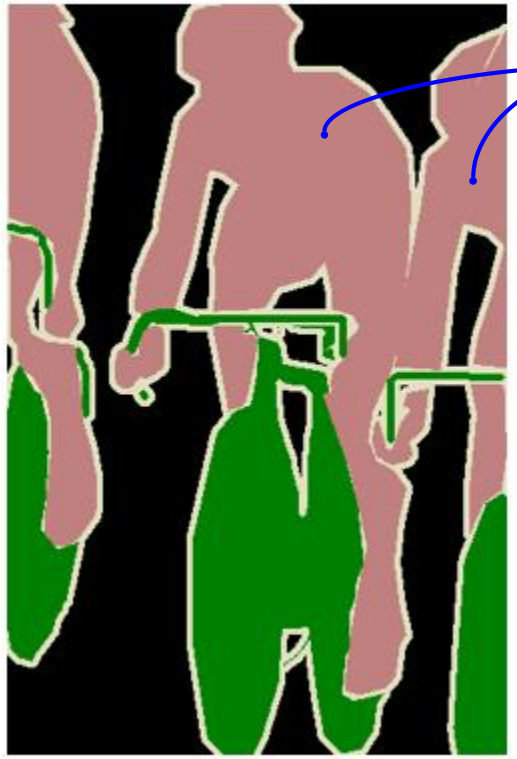
[1] Figure credits: John Wilson

Semantic Segmentation

: Task of assigning **a label** for **every pixel** of an image with a corresponding **class**

Dense prediction

predict



Different objects,
but same prediction
→ limitation

Person
Bicycle
Background

[1] Figures from Pascal VOC Challenge 2012

Semantic Segmentation: Input & Output

Groud Truth (GT) → supervised learning

segmented

1: Person
2: Purse
3: Plants/Grass
4: Sidewalk
5: Building/Structures

3	3	3	3	3	3	3	3	3	3	3	3	5	5	5	5	5	5
3	3	3	3	3	3	3	3	3	3	3	3	5	5	5	5	5	5
3	3	3	3	3	3	1	1	3	3	3	3	5	5	5	5	5	5
3	3	3	3	3	1	1	1	1	3	3	3	5	5	5	5	5	5
3	3	3	3	3	3	1	1	3	3	3	5	5	5	5	5	5	5
5	5	3	3	3	3	1	1	3	3	5	5	5	5	5	5	5	5
4	4	3	4	1	1	1	1	1	1	4	4	4	5	5	5	5	5
4	4	3	4	1	1	1	1	1	1	4	4	4	4	4	5	5	5
4	4	4	1	1	1	1	1	1	1	1	4	4	4	4	4	4	4
3	3	3	1	1	1	1	1	1	1	1	4	4	4	4	4	4	4
3	3	3	1	2	2	1	1	1	1	1	4	4	4	4	4	4	4
3	3	3	1	2	2	1	1	1	1	1	4	4	4	4	4	4	4

Input → Input image resolution: $\boldsymbol{W} \times \boldsymbol{H} \times \mathbf{3}$ Semantic Labels

Note that this is a low-resolution prediction map for visual clarity.
In reality, the segmentation label resolution should match the original input's resolution ($\boldsymbol{W} \times \boldsymbol{H} \times \mathbf{3}$).

[1] Figure credits: Jeremy Jordan

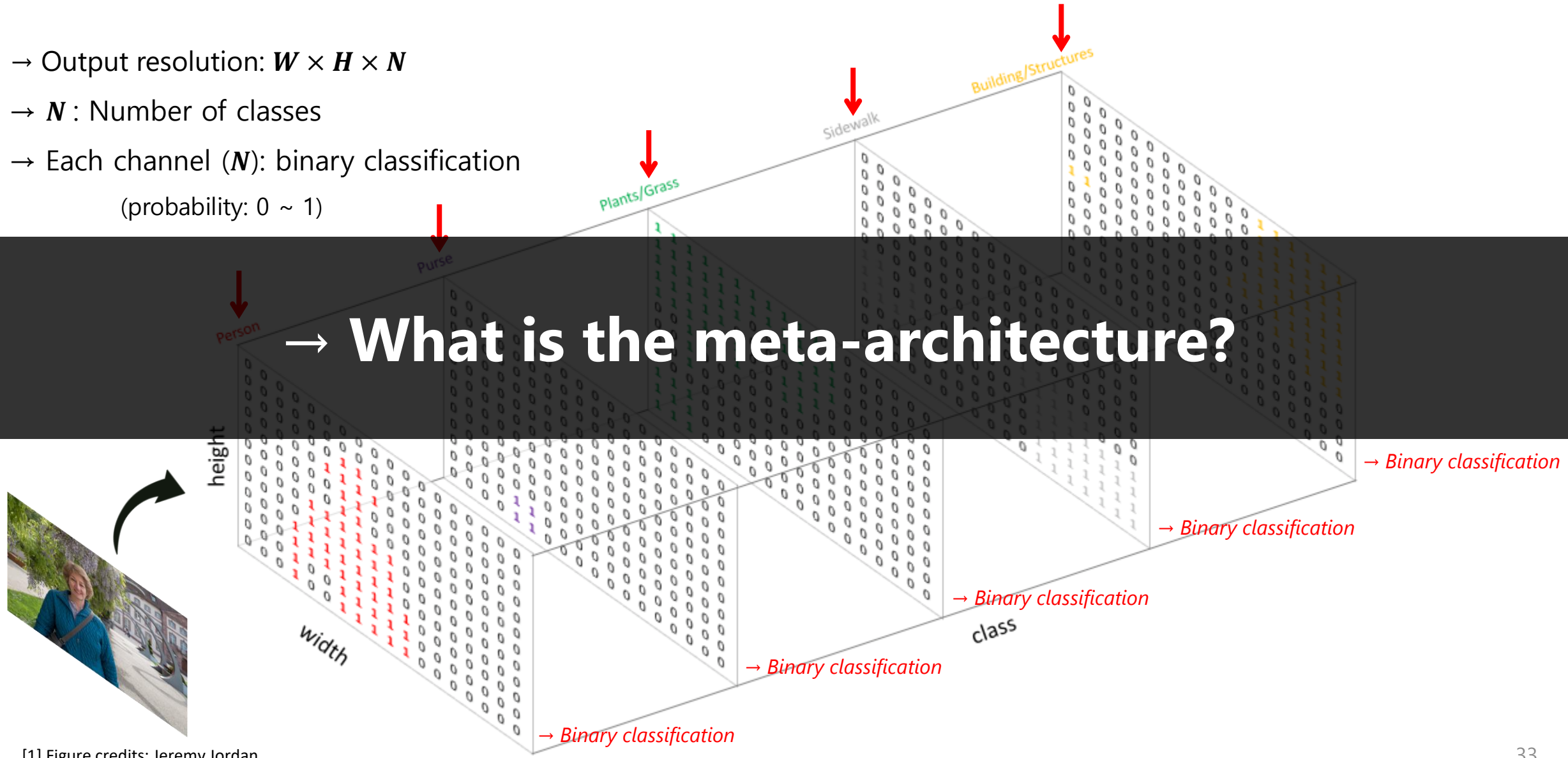
Semantic Segmentation: Input & Output

→ Output resolution: $\boldsymbol{W} \times \boldsymbol{H} \times \boldsymbol{N}$

→ $\boldsymbol{N}$: Number of classes

→ Each channel ($\boldsymbol{N}$): binary classification

(probability: 0 ~ 1)



→ What is the meta-architecture?

[1] Figure credits: Jeremy Jordan

Image Classification

This image is CC0 public domain

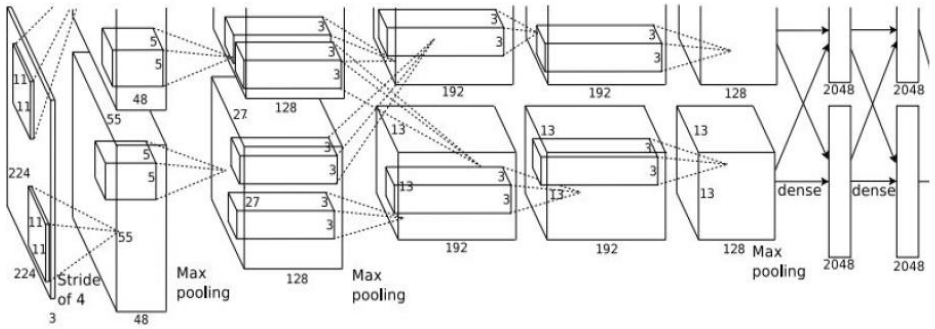


Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

Vector:
4096

Fully-Connected:
4096 to 1000

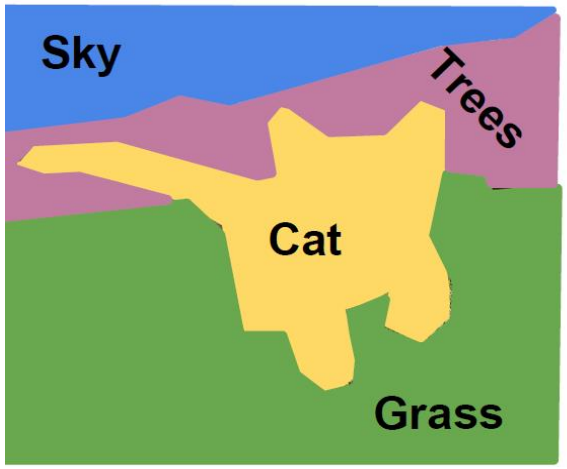
Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

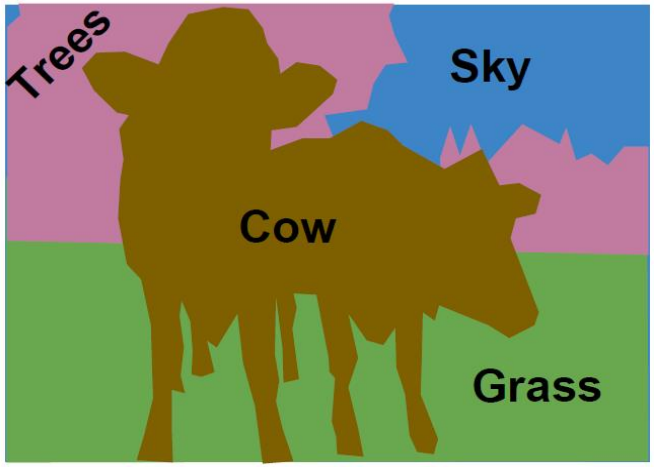
[1] Stanford university cs231n lecture slide

Semantic Image Segmentation

Label each pixel in the image with a category label

- **Pixel-level** classification task
- Requires **spatial** information

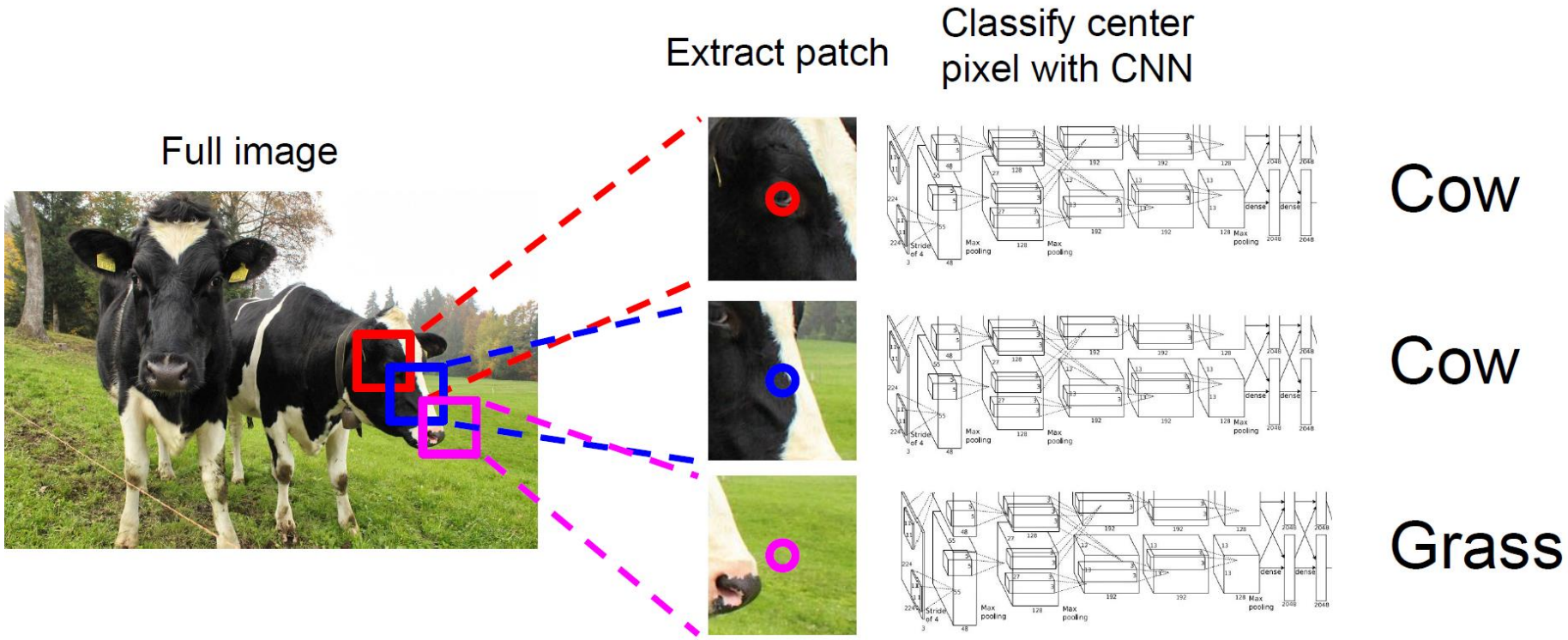




[1] Stanford university cs231n lecture slide

Sliding Window Approach

Perform classification by **sliding window**



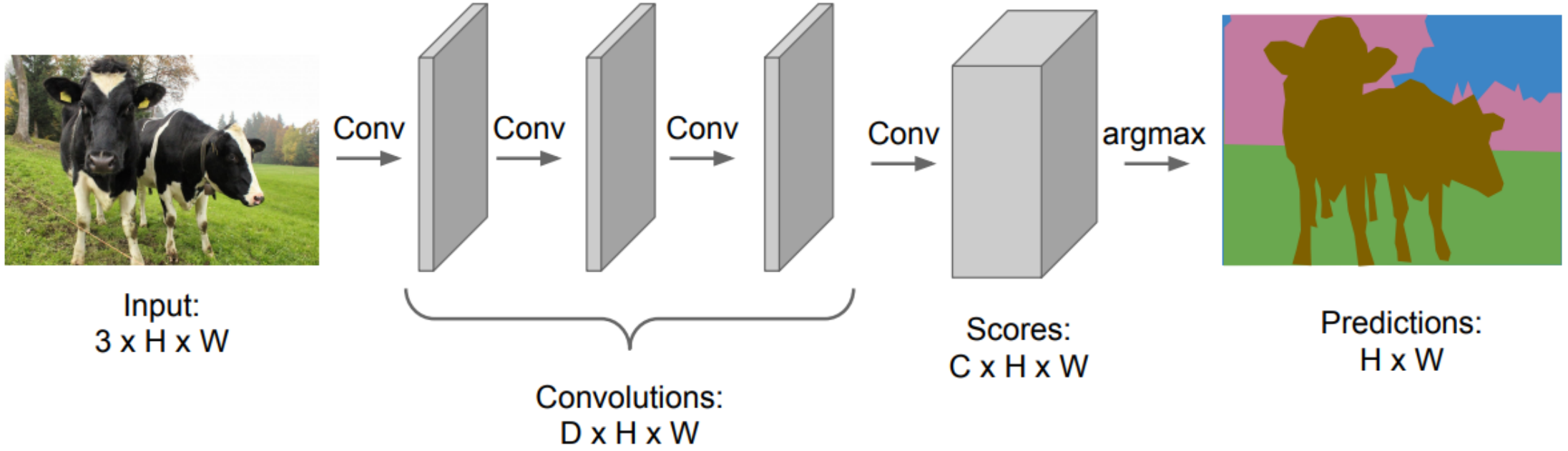
→ Computationally **inefficient**

= Not reusing features for shared image region

[1] Stanford university cs231n lecture slide

Fully Convolutional Approach

Design a network as **a bunch of convolutional layers** (preserving spatial information) to make predictions for pixels **all at once**!

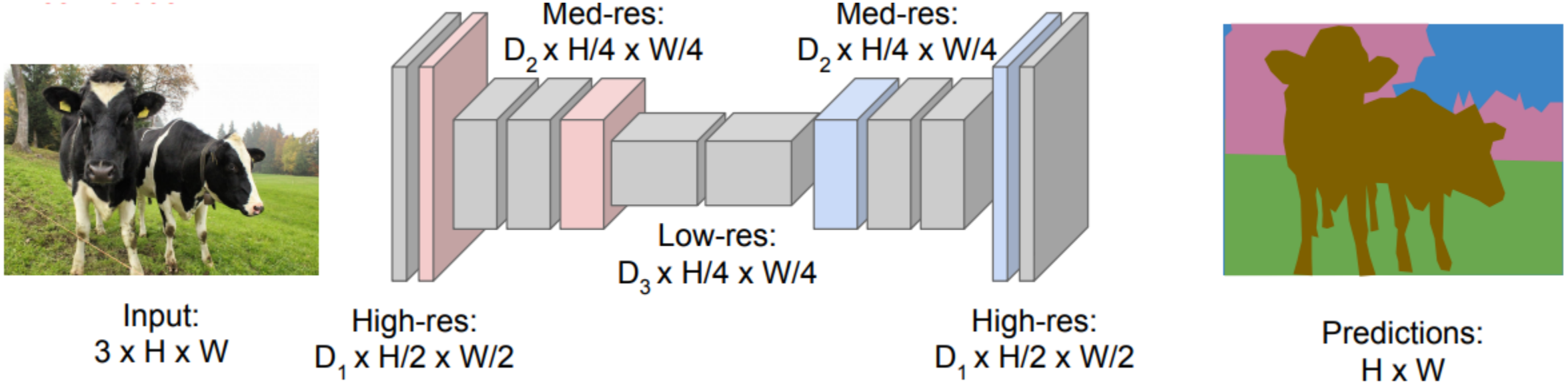


→ Convolutions at original image resolution will be **very expensive!**

[1] Stanford university cs231n lecture slide

Fully Convolutional Approach

Design a network as a bunch of convolutional layers,
with **downsampling** and **upsampling** inside the network!



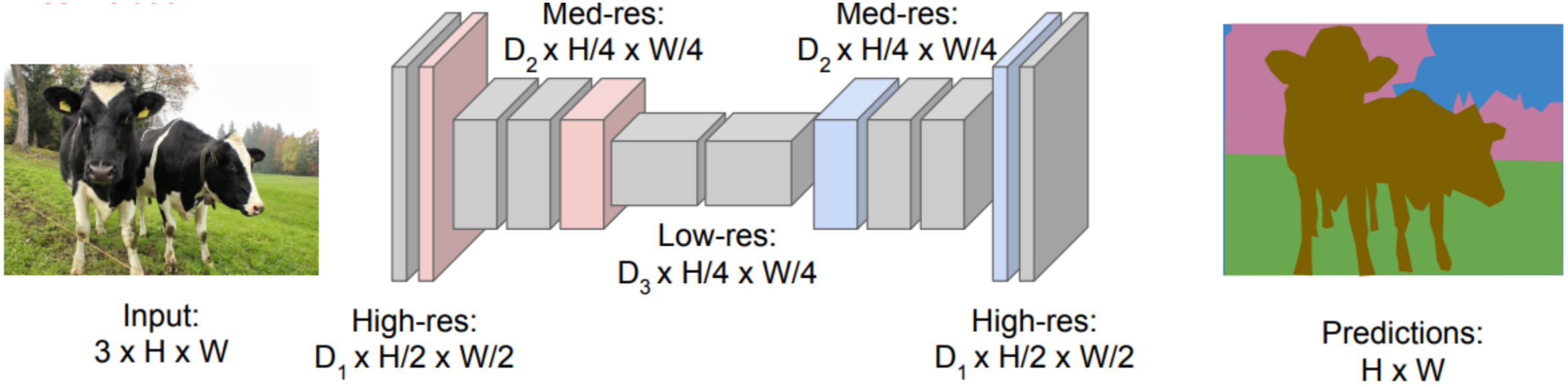
Downsampling: pooling, strided convolution (encoder)

Upsampling: unpooling, strided transpose convolution (decoder)

[1] Stanford university cs231n lecture slide

Fully Convolutional Approach

Design a network as a bunch of convolutional layers,
with **downsampling** and **upsampling** inside the network!



→ This is the **basic structure (Encoder-Decoder)** for segmentation task.

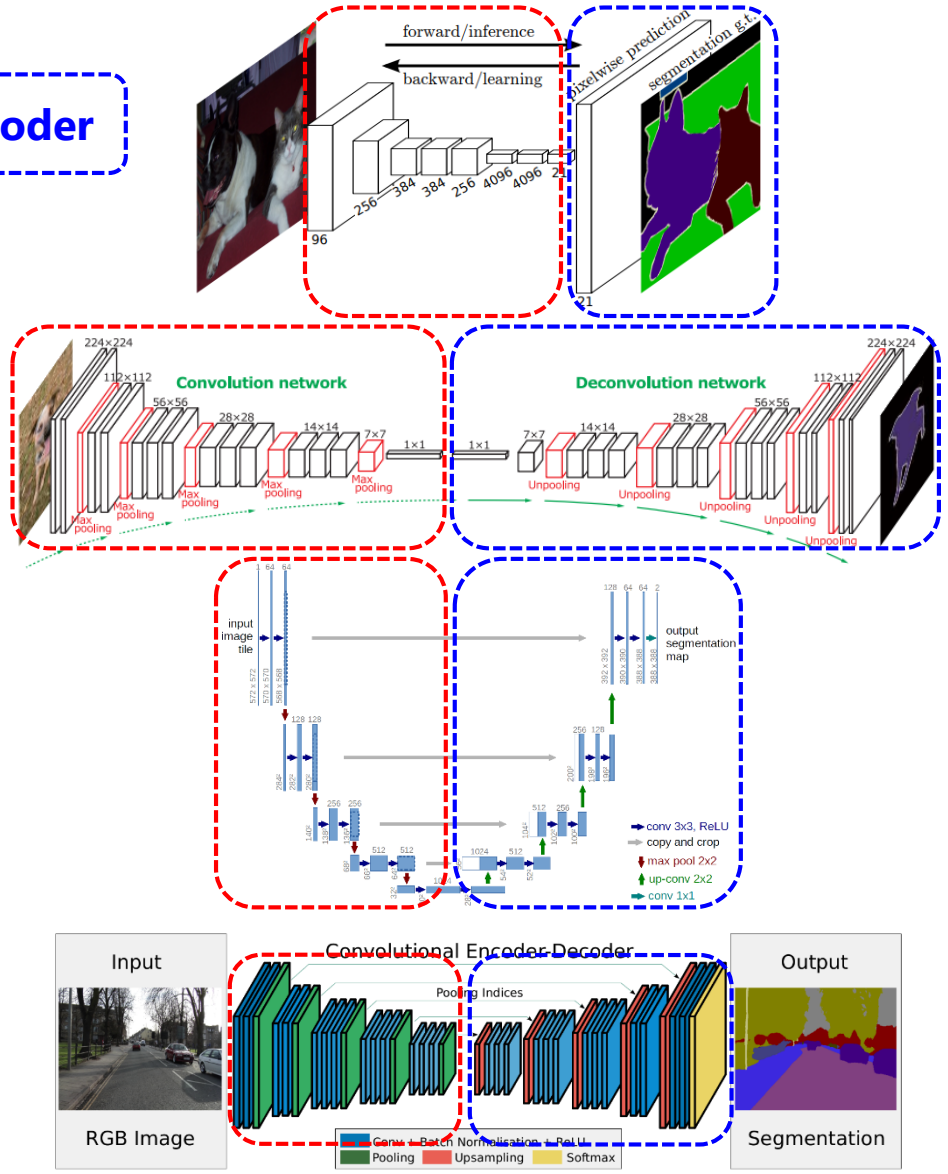
[1] Stanford university cs231n lecture slide

Encoder-Decoder Architectures

References

Encoder Decoder

- J. Long, et al., "**Fully Convolutional Networks.**" *CVPR*, 2015.
  → Meta-architecture for semantic segmentation
- H. Noh, et al., "**Learning Deconvolution Network for Semantic Segmentation.**" *ICCV*, 2015.
- O. Ronneberger, et al., "**U-Net: Convolutional Networks for Biomedical Image Segmentation.**" *MICCAI*, 2015.
- V. Badrinarayanan, et al., "**SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation.**" *T-PAMI*, 2016.



*Back to Basics: FC & Convolution

Fully-connected (FC) layer

→ A.k.a. multi-layer perceptron (MLP)

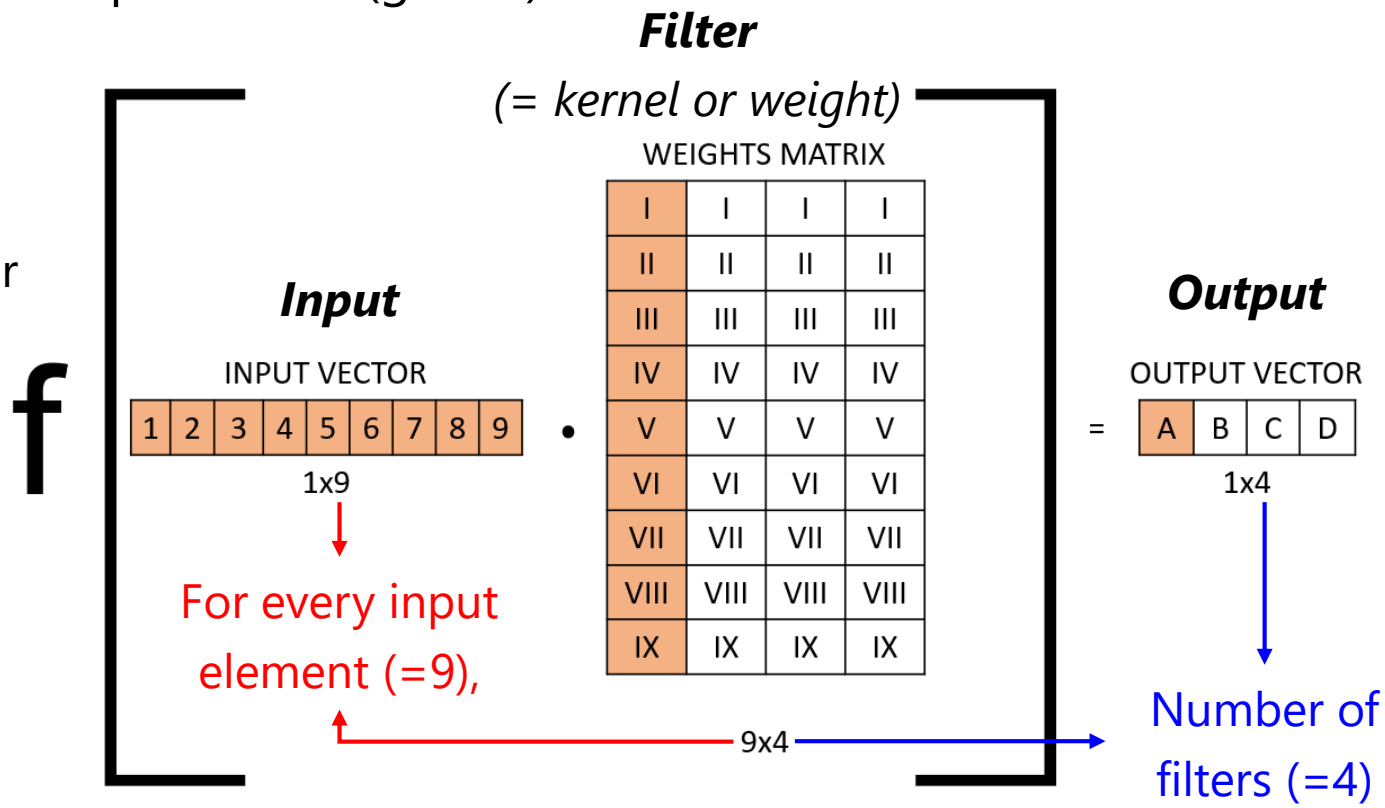
→ **Vector** operation

→ Every input element affects output vector (global)

→ Input size is **fixed**

Input feature 3 × 3 × 1

→ stretch to 1 × 9 vector



[1] Stanford university cs231n lecture slide

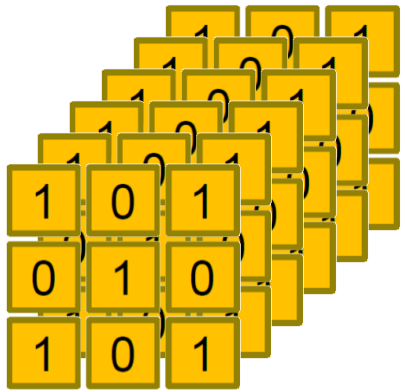
*Back to Basics: FC & Convolution

Convolutional layer

→ **Matrix** (2D or 3D) operation

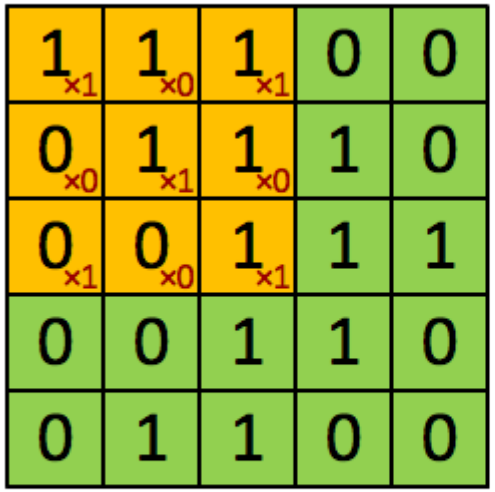
→ Aggregate **local** & **spatial** features (flexible input resolution)

→ 1 × 1 convolution = FC layer

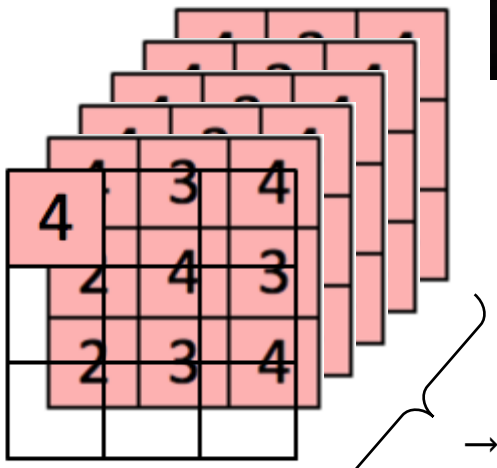


Filter size 3 × 3

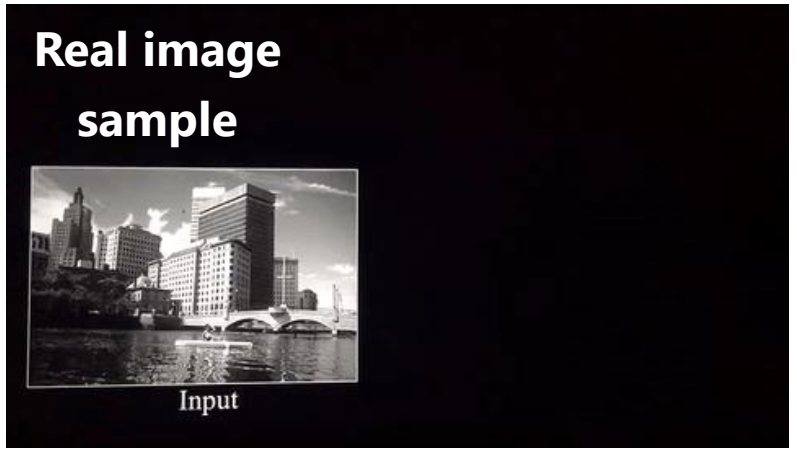
Input image 5 × 5



Image



Convolved Feature



→ Sliding window operation

→ **If the number of filters = 6**

→ **Number of output features = 6**

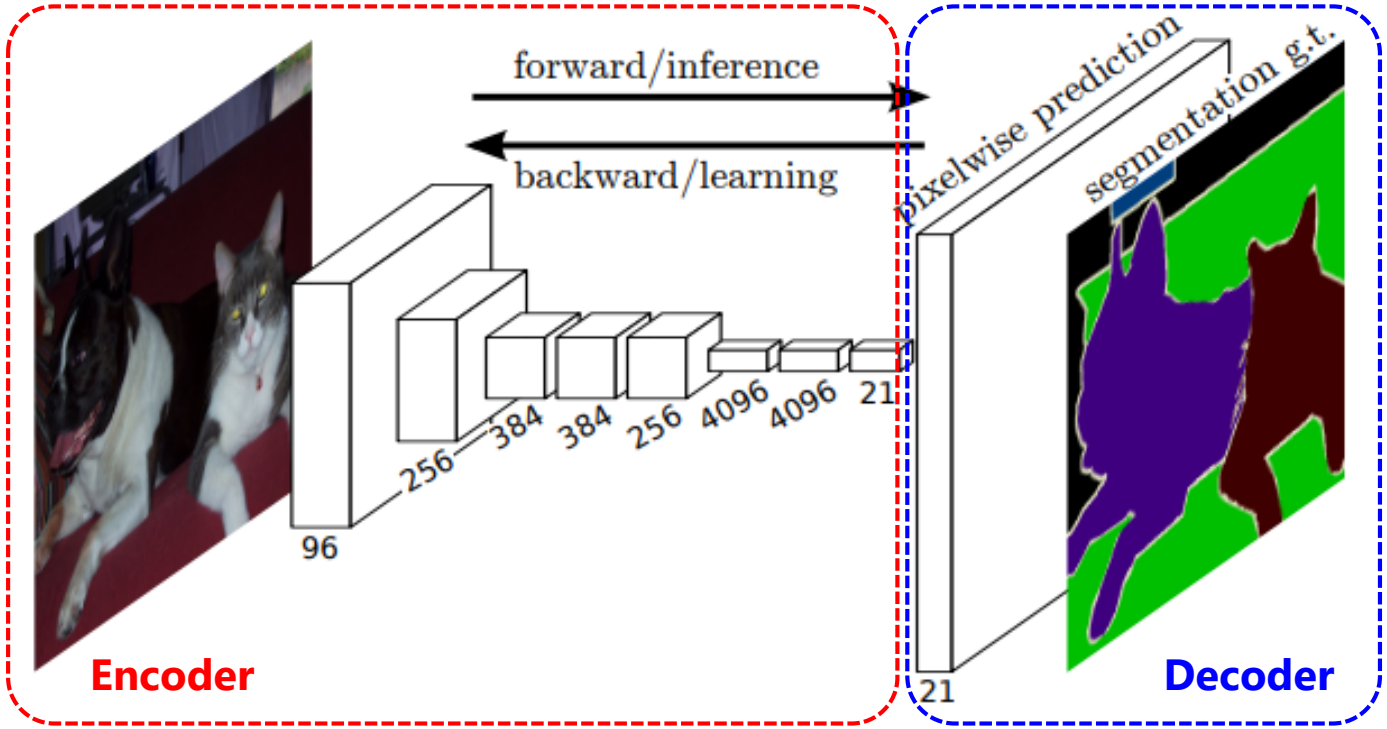
[1] Stanford university cs231n lecture slide

FCN: The First CNN-Based Segmentation Network

Key idea

→ **Fully convolutional layers** to extract **spatial** features

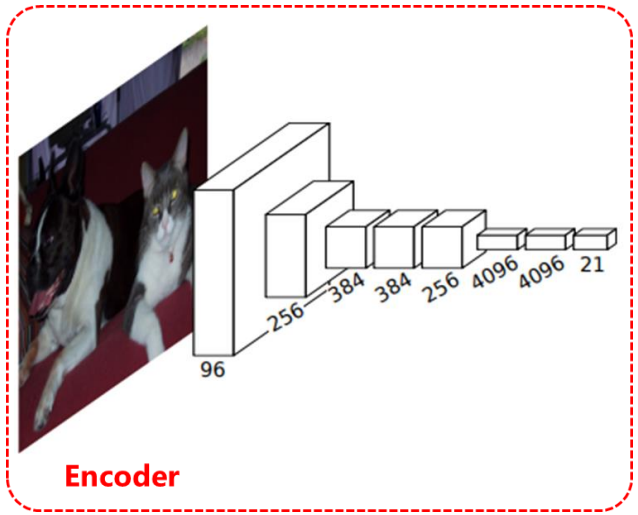
Specifically for VGG-16 model, the last three FC layers are replaced by conv layers.



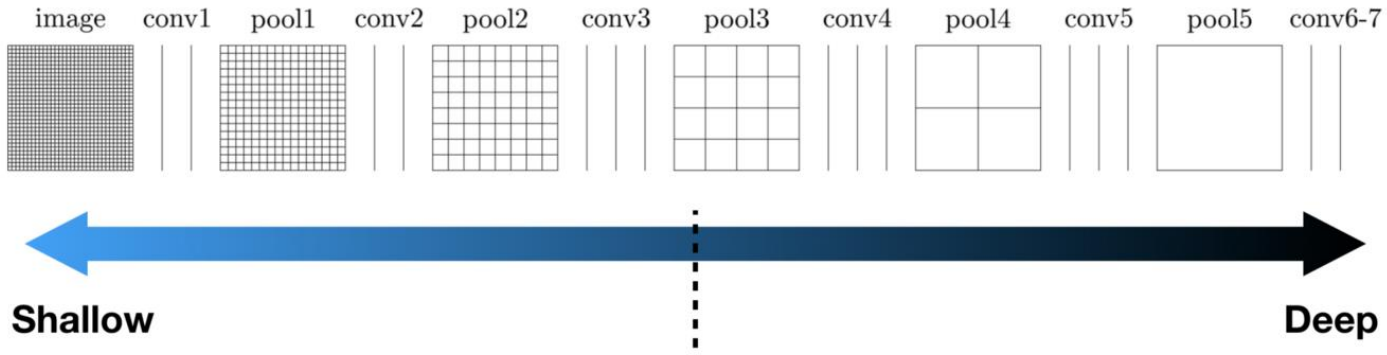
[1] J. Long, et al., "Fully Convolutional Networks." CVPR, 2015.

Encoding = Extracting Abstract Features

Encoding stage



=



Fine	Coarse
Location	Semantic
Local	Global
Detail	Abstract

[1] J. Long, et al., “Fully Convolutional Networks.” CVPR, 2015.

Decoding = Aggregating Details

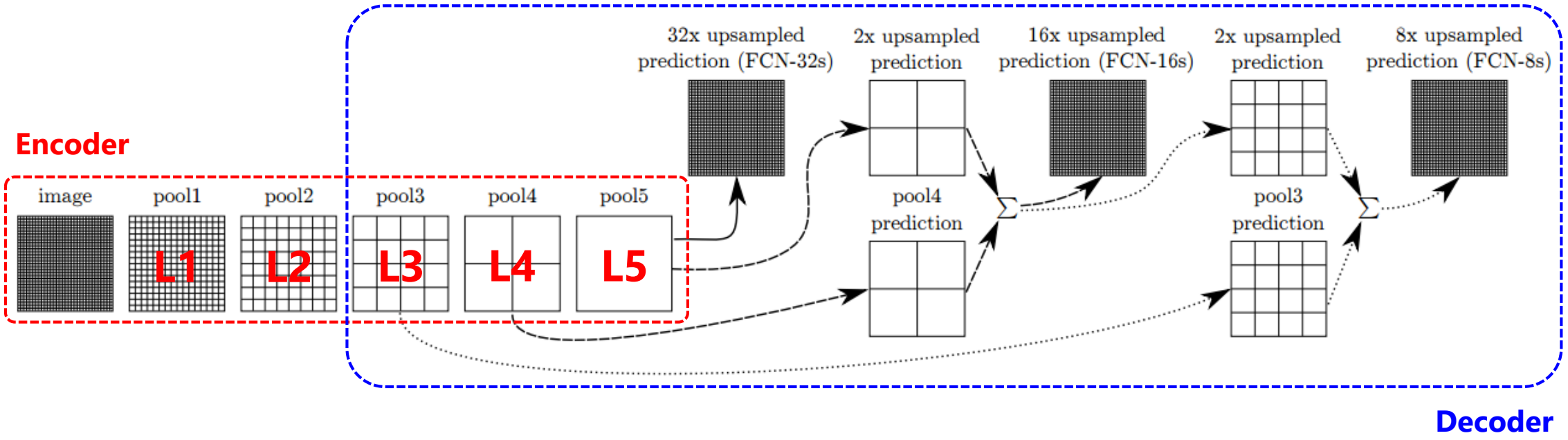
Decoding stage

→ Skip-connections to aggregate **multi-scale** spatial features.

Specifically, it takes details from **L3**, **L4**, and **L5**

→ Upsampling using a **bilinear interpolation**

Also **differentiable** for backpropagation



[1] J. Long, et al., “Fully Convolutional Networks.” CVPR, 2015.

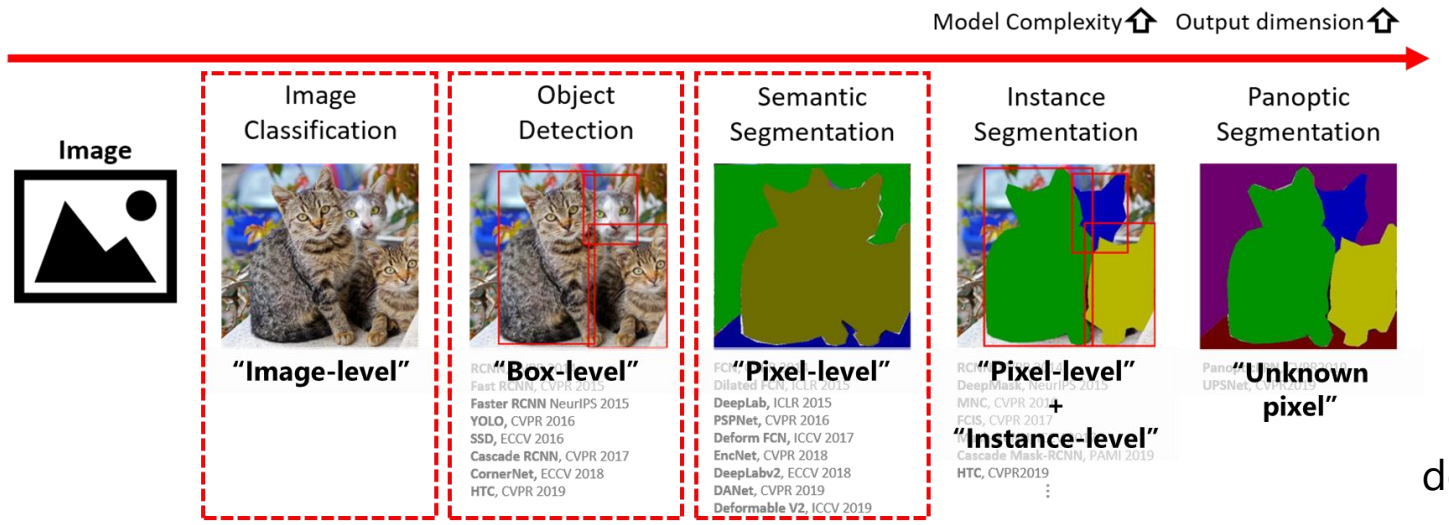
Summary

Basic computer vision tasks (semantic tasks)

→ Image classification (AlexNet, VGG, GoogleNet, ResNet): basic deep neural networks

→ Object detection (R-CNN, SSD): Trade-off between two-stage & one-stage detectors

→ Semantic segmentation (Fully Convolutional Network): Fully-connected vs. Convolutional layers



→ Combinatorial works of object detection and semantic segmentation