

Applications of CNNs in Computer Vision: Detection, Segmentation

Ju Sun

Computer Science & Engineering

Mar 26, 2026



UNIVERSITY OF MINNESOTA
Driven to DiscoverSM

Disclaimer

This set of slides are modified from slides made by **Ms. Andrea Walker** in 2020 Fall on the same topic for CSCI8980: Think Deep Learning. The object detection part borrows a lot of materials from the book: “Deep Learning for Vision Systems” by Mohamed Elgendy

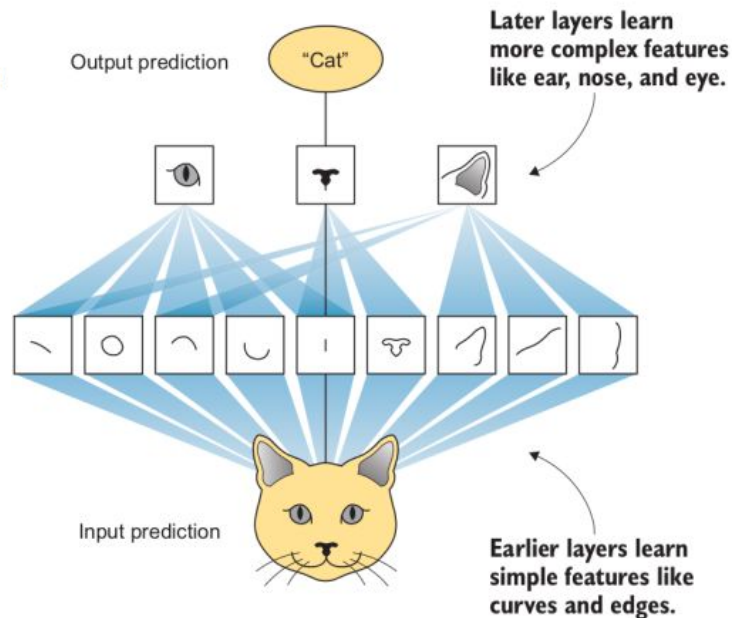
<https://www.manning.com/books/deep-learning-for-vision-systems>

Most models we talked so far for classification

Classification

The following classification models are available, with or without pre-trained weights:

- AlexNet
- ConvNeXt
- DenseNet
- EfficientNet
- EfficientNetV2
- GoogLeNet
- Inception V3
- MaxVit
- MNASNet
- MobileNet V2
- MobileNet V3
- RegNet
- ResNet
- ResNeXt
- ShuffleNet V2
- SqueezeNet
- SwinTransformer
- VGG
- VisionTransformer
- Wide ResNet



(Credit: [Elgendy, 2020])

Other main tasks in computer vision

- **Object detection**
- **Segmentation**

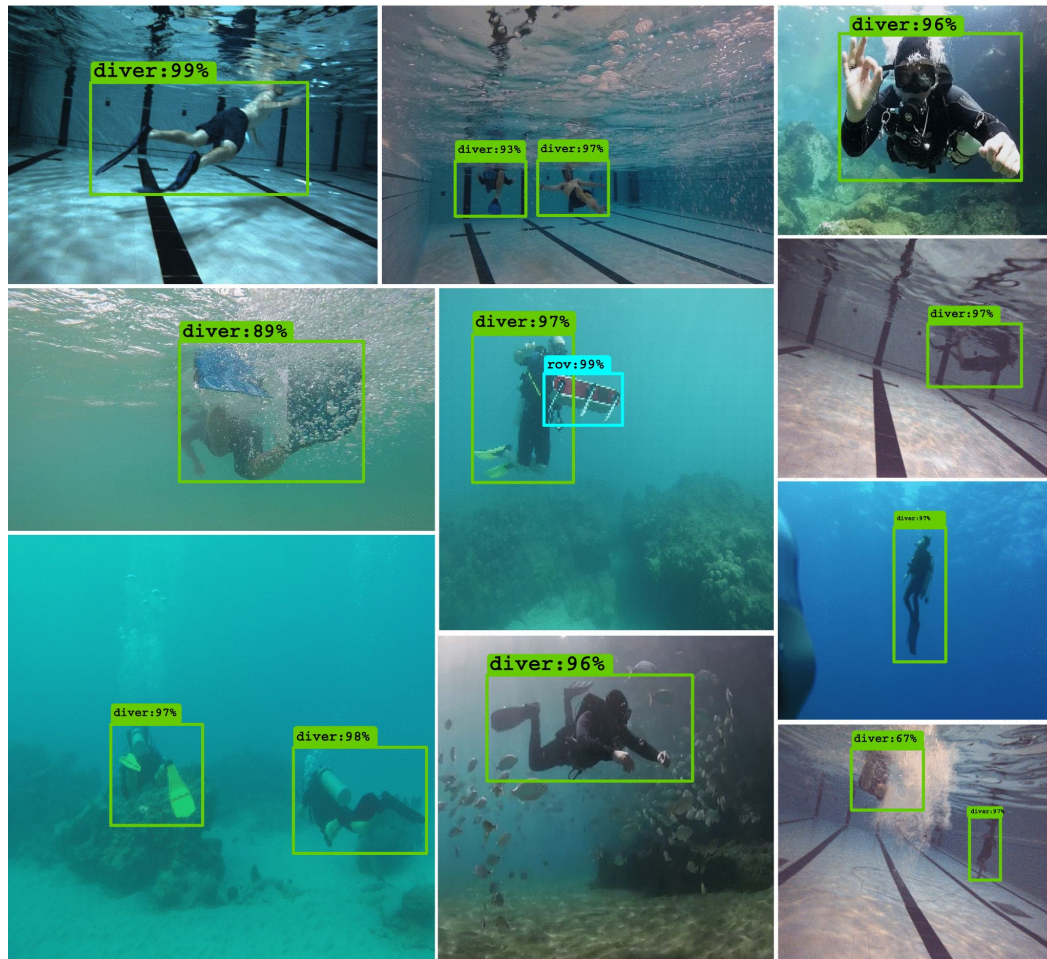
Hybrid classification-regression problems in CV

Object detection

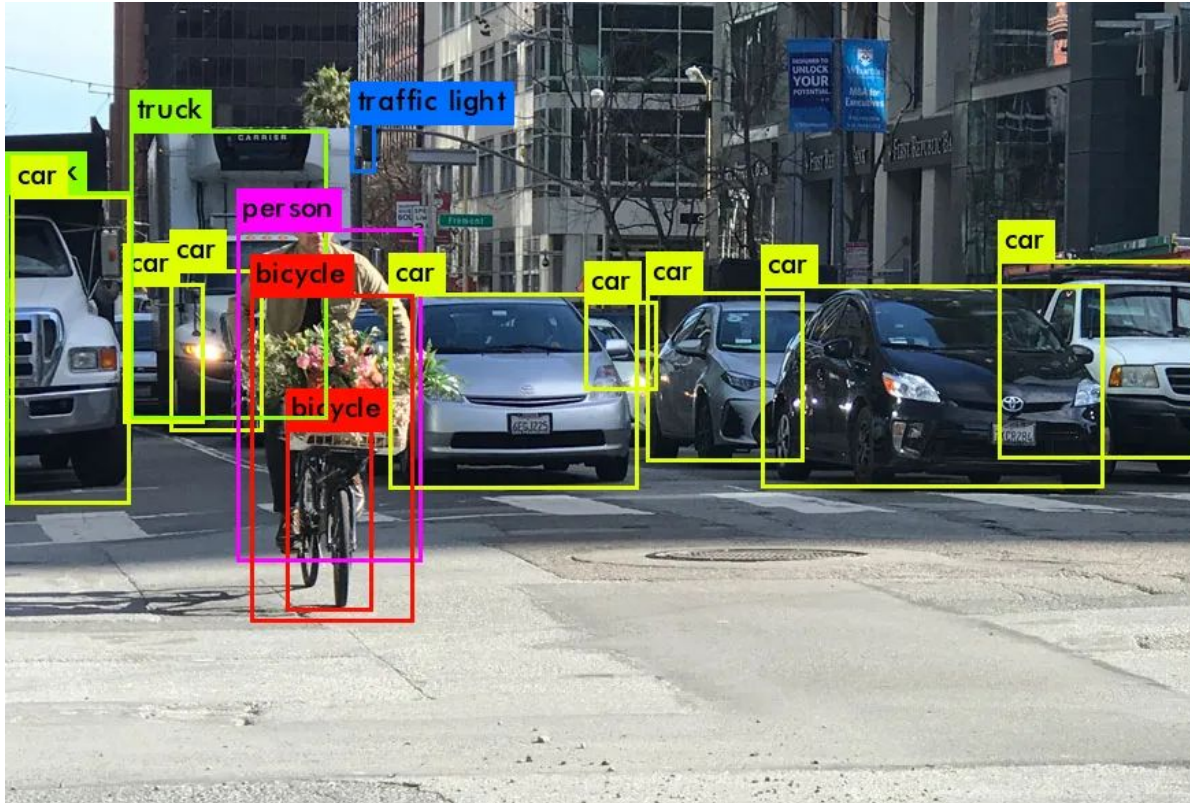
- **Localization:**
where the objects are (by providing bounding boxes)
- **Classification:**
what the objects are (by providing label for each bounding box)

(Islam et al., "Toward a Generic Diver-Following Algorithm: Balancing Robustness and Efficiency in Deep Visual Detection," 2019)

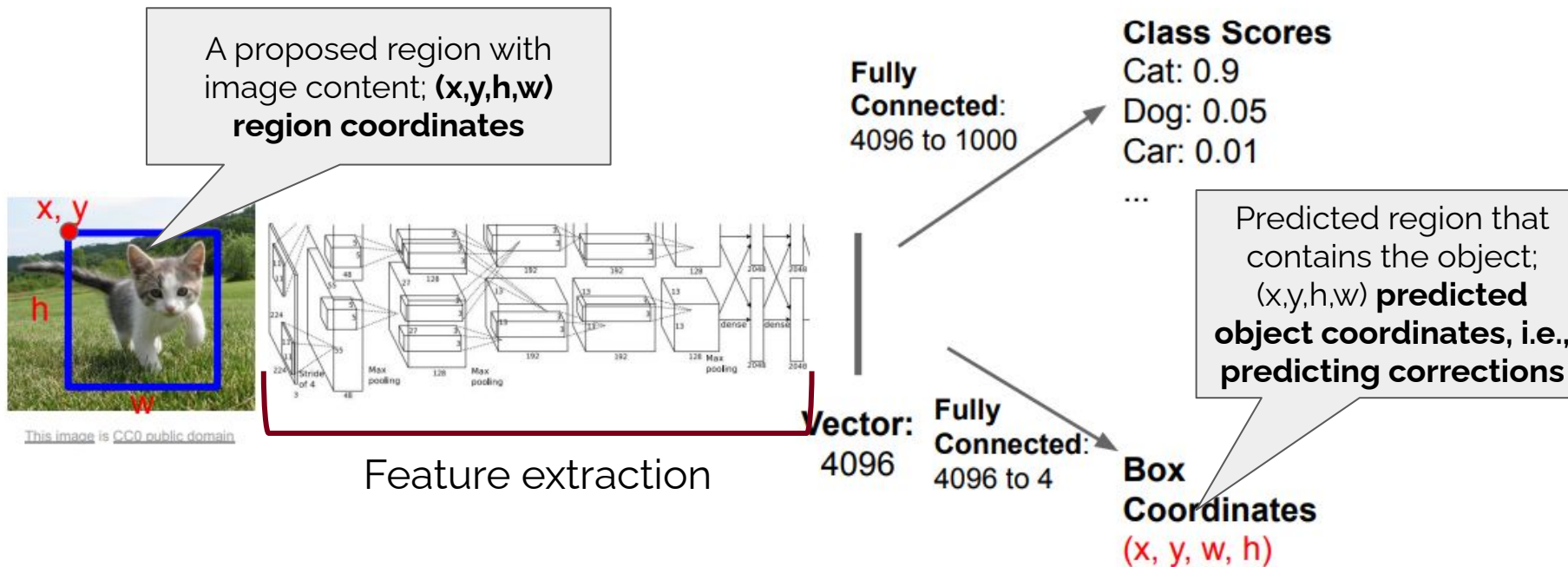
Paper from UMN IRVLab: <https://irvlab.cs.umn.edu/>



For autonomous driving



Object detection: inference



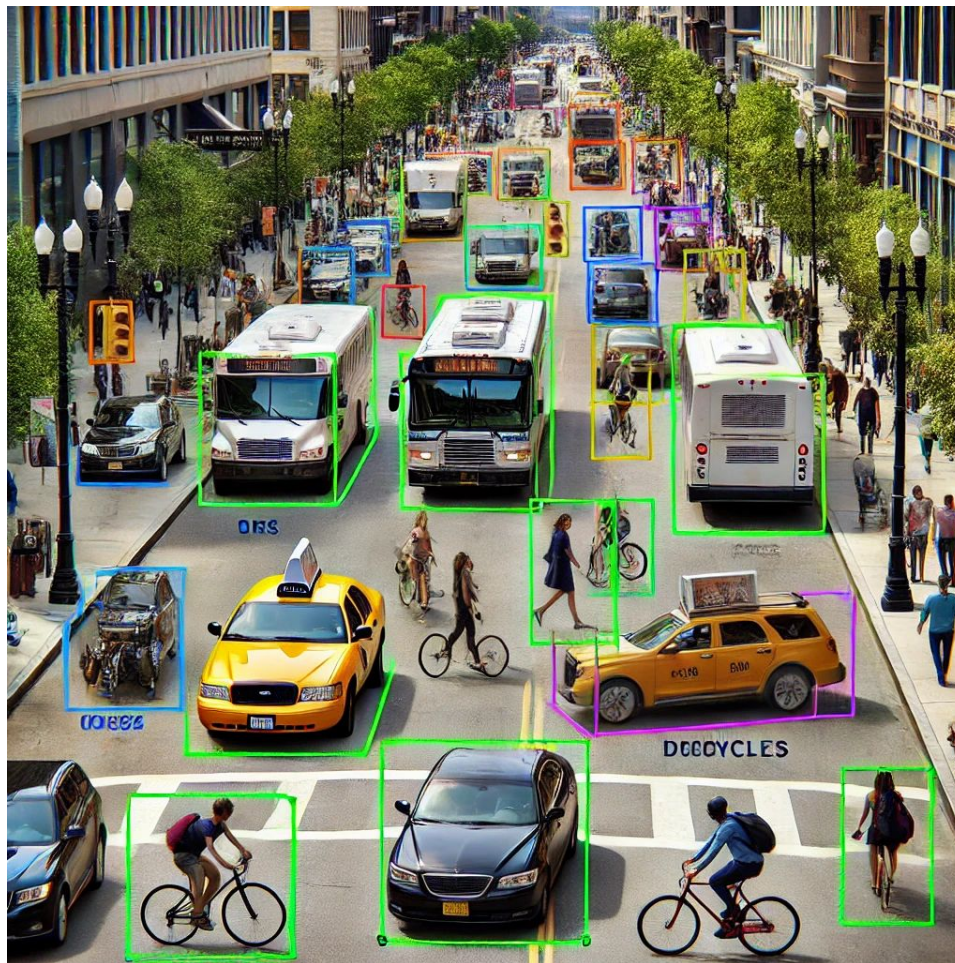
Training datasets

Bounding boxes are stored as

COCO format: Uses JSON files with details like `bbox` (x, y, width, height).

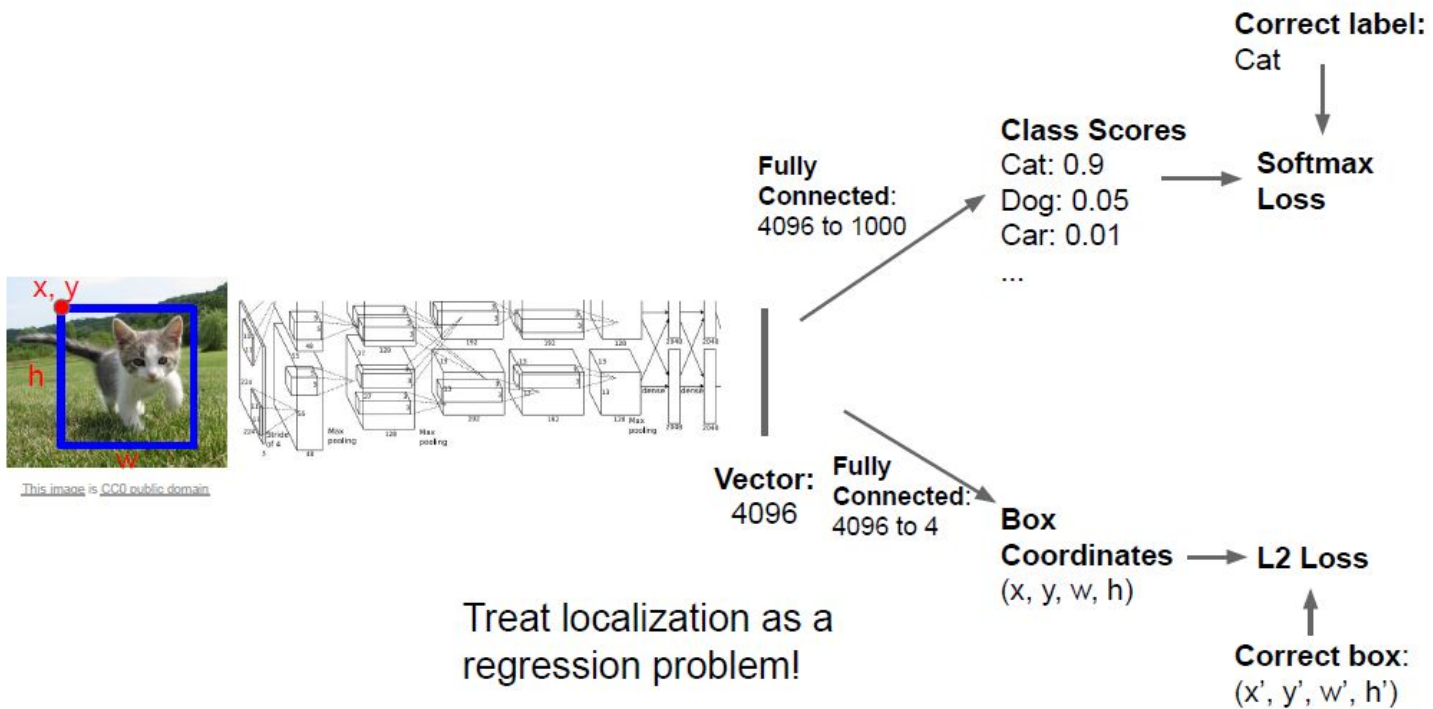
Pascal VOC format: Uses XML files with coordinates.

YOLO format: Uses a text file with normalized (x_center, y_center, width, height) values



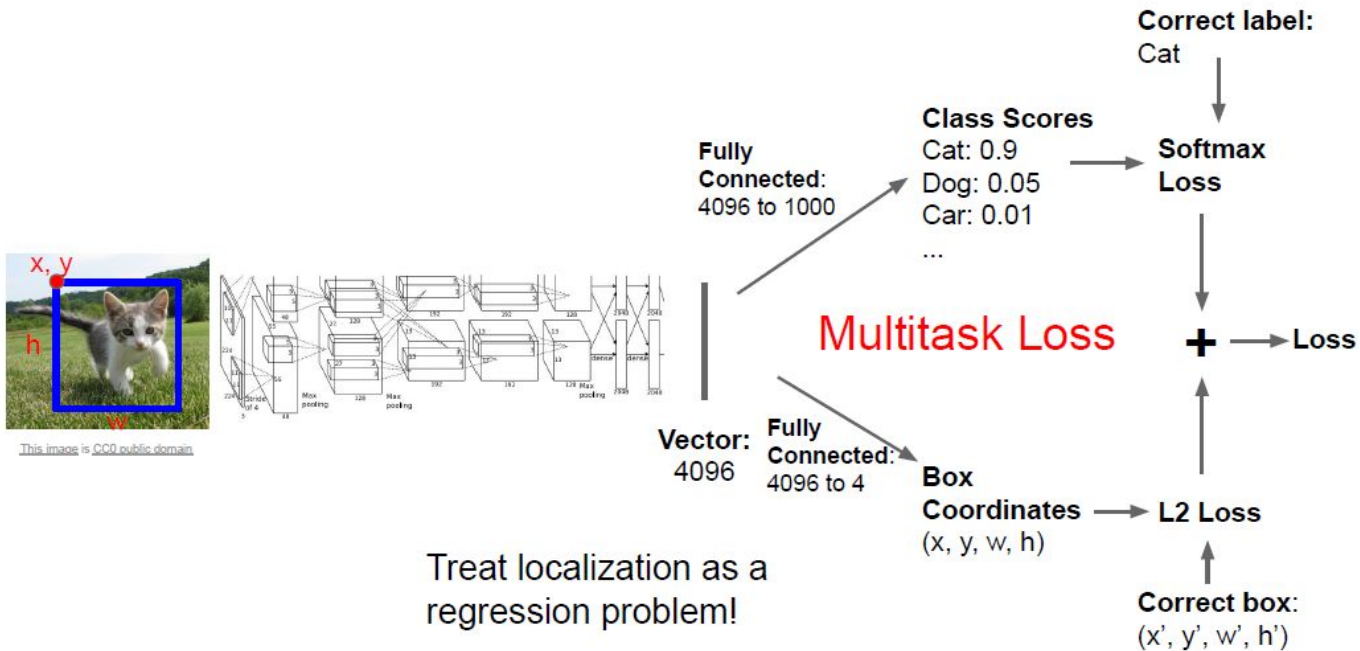
... contain many images with bounding boxes

Object detection: training

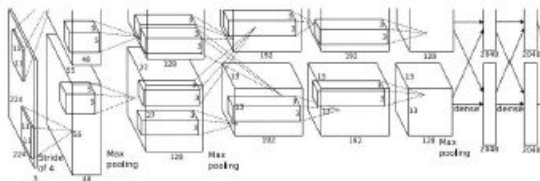


(Li et al., *Detection and Segmentation* 2020)

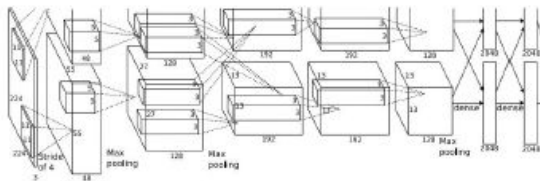
Object detection: training



Multiple objects: multiple outputs



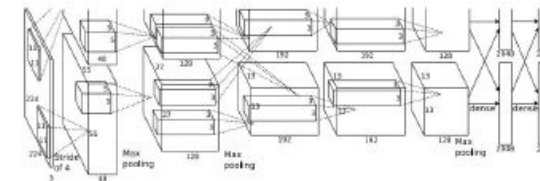
CAT: (x, y, w, h)



DOG: (x, y, w, h)

DOG: (x, y, w, h)

CAT: (x, y, w, h)



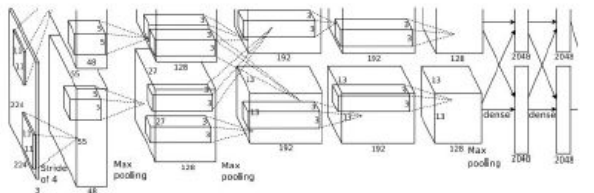
DUCK: (x, y, w, h)

DUCK: (x, y, w, h)

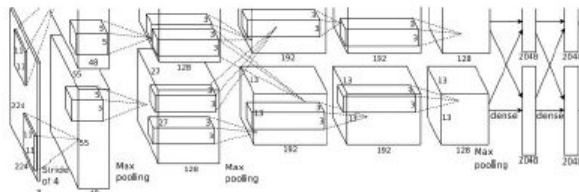
....

Multiple objects: initial solution

Scanning window method



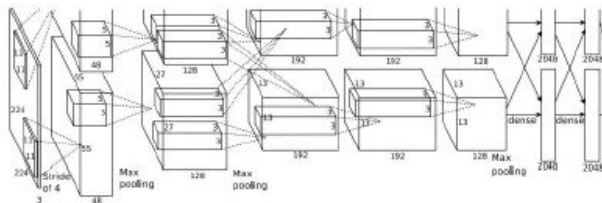
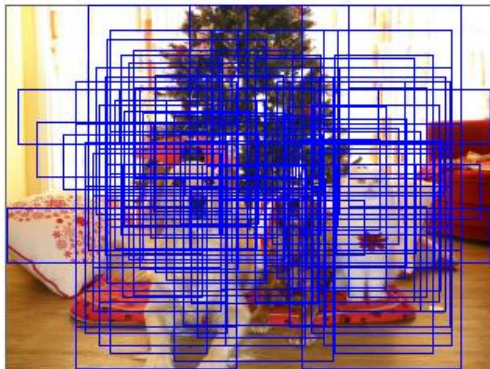
Dog? NO
Cat? NO
Background? YES



Dog? YES
Cat? NO
Background? NO

Multiple objects: heavy computational cost

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO
Cat? YES
Background? NO

Problem: Need to apply CNN to huge number of locations, scales, and aspect ratios, very computationally expensive!

Solution: 4-step object-detection framework

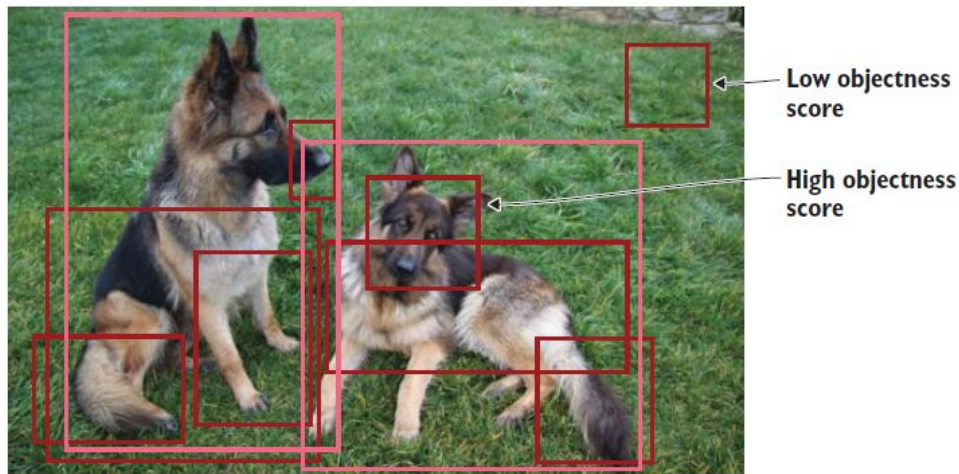
1. **Region proposal:** identify regions of interest (RoI) for potential locations of objects
2. **Feature extraction:** extract visual features within each RoI for classification
3. **Non-maximum suppression:** avoid repeated detections
4. **Evaluation metrics:** evaluate performance of model

1. Region proposal

Propose Regions of Interest (Rois)

- General procedures
 - Generate thousands of bounding boxes (BBs)
 - Classify BBs as foreground or background based on 'objectness score'
 - Pass only foreground through rest of network

- Popular: **selective search**
 - Fast algorithm, ~200 region proposals in a few seconds on CPU

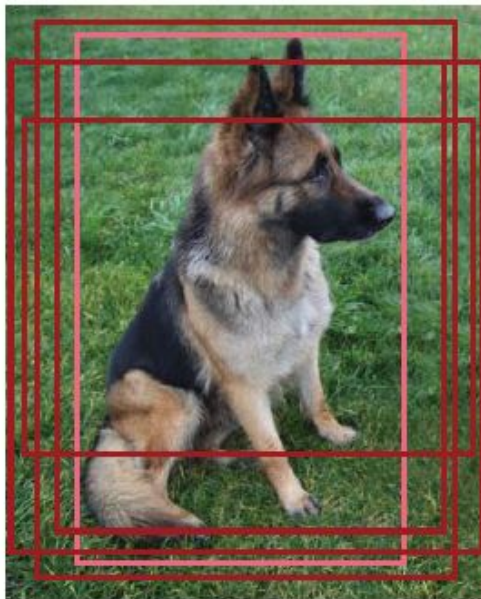


Selective search



2. Feature extraction & prediction in each RoI

- Extract features using a pretrained CNN
(Remember **transfer learning**?)
- Make 2 predictions using additional layers:
 - Bounding box prediction (x, y, width, height)
 - Class prediction (softmax function predicting the class probability for each object class)



3. Remove duplicate object detections

Non-maximum suppression (NMS), aka **Spatial Clustering**, to eliminate duplicate detections

1. Discard BBs with predictions below a **confidence threshold**.
2. Select the BB with the highest probability
3. Calculate IoU scores of all other BB's with the selected
4. Discard BB's with small IoU scores (e.g., ≤ 0.5) and average those left

Intersection over Union (IoU)

Person

$$\text{Score} = \frac{\text{Area of overlap}}{\text{Area of union}}$$


BB: bounding box



Predictions before NMS



After applying non-maximum suppression

4. Evaluation metrics for detection performance

1. **Frames per second (FPS)** - detection speed
2. **Mean Average Precision (mAP)** - detection accuracy

mAP: Class average of AP, which is the area under the **precision-recall curve**

Intersection over Union (IoU)

$$\text{IoU} = \frac{B_{\text{ground truth}} \cap B_{\text{predicted}}}{B_{\text{ground truth}} \cup B_{\text{predicted}}}$$

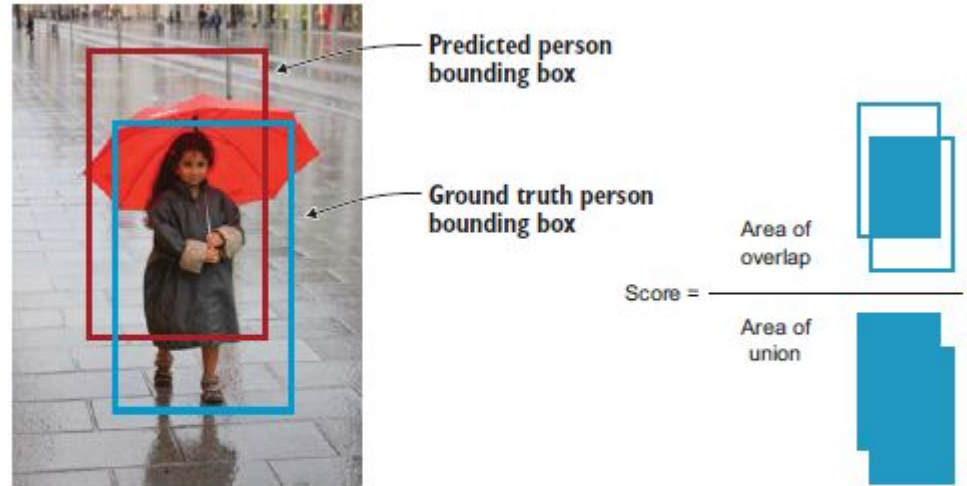
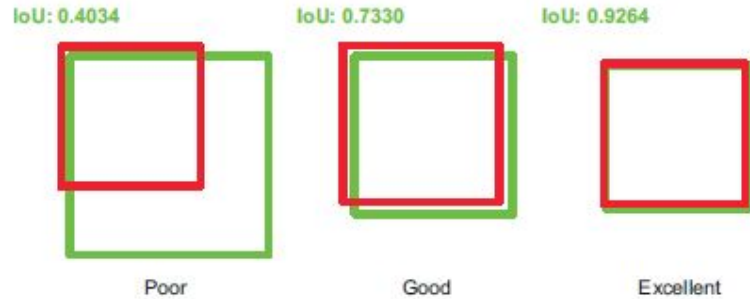


Figure 7.5 The IoU score is the overlap between the ground truth bounding box and the predicted bounding box.

Precision-recall curve and the area under

		Predicted condition	
		Positive (PP)	Negative (PN)
Total population = P + N			
Actual condition	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

Can't we use accuracy?

Precision-Recall (PR):

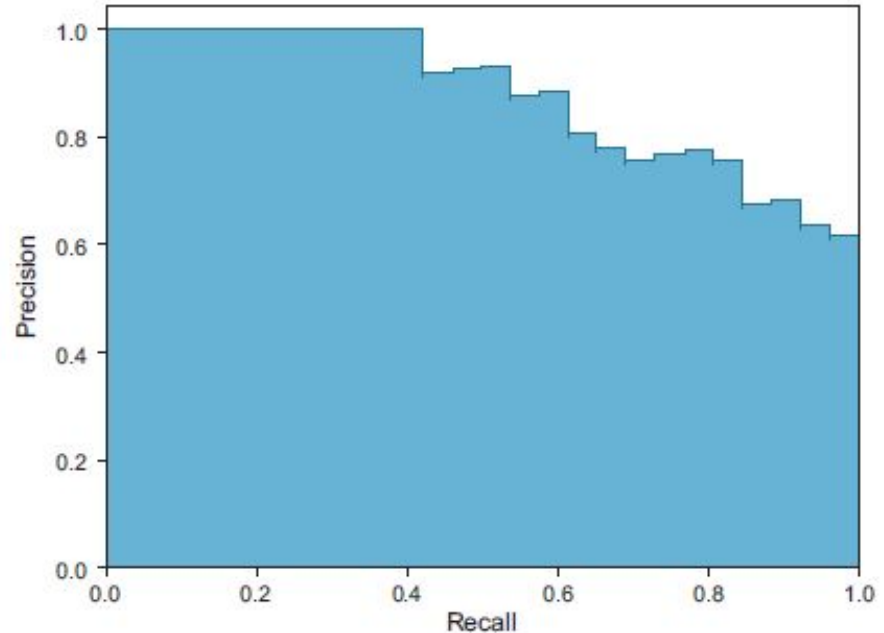
Completeness:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Sharpness:

$$\text{Precision} = \frac{TP}{TP + FP}$$

When we vary the IoU threshold ...



State of the Art Object Detection CNNs

- R-CNNs

Object Detection

- SSD

The following object detection models are available, with or without pre-trained weights:

- YOLO

- Faster R-CNN
- FCOS
- RetinaNet
- SSD
- SSDlite

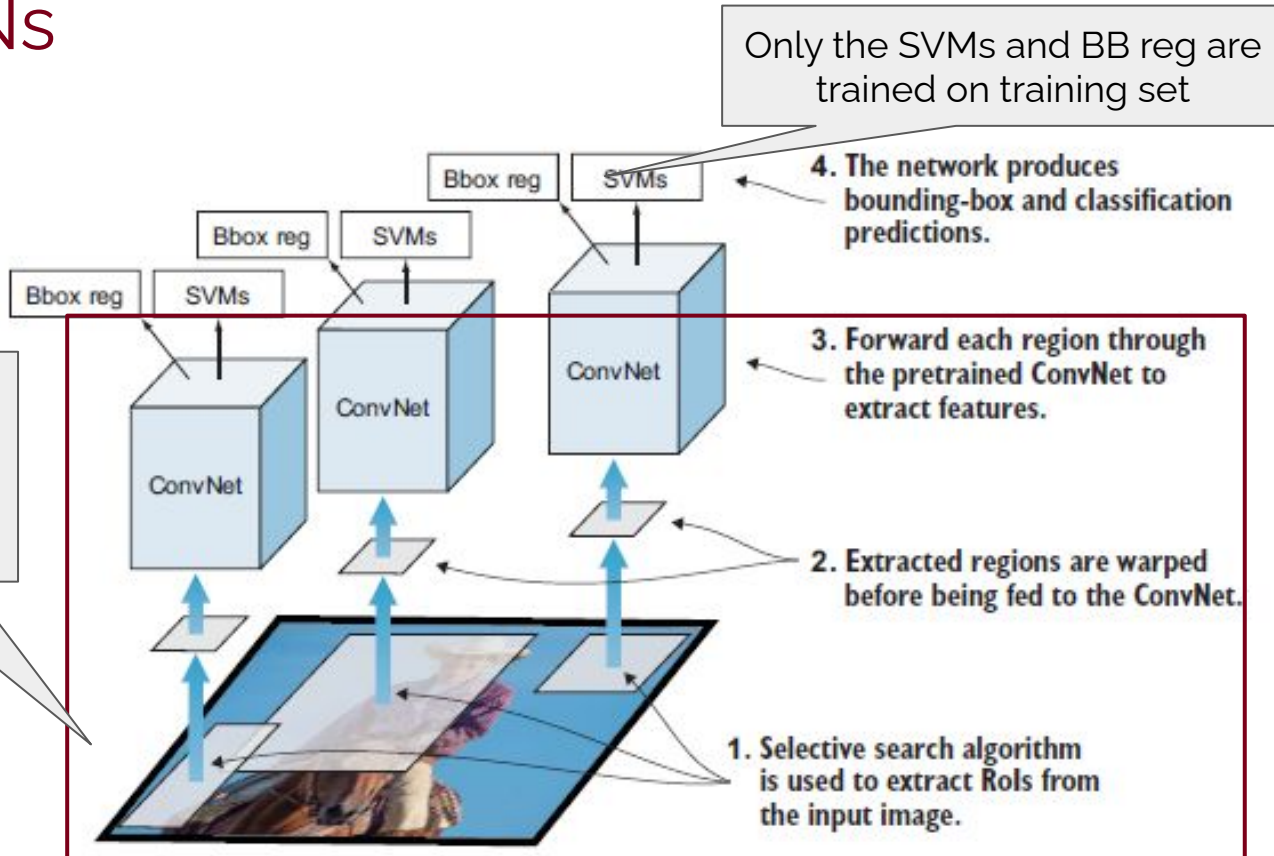
<https://pytorch.org/vision/stable/models.html#object-detection-instance-segmentation-and-person-keypoint-detection>

R-CNNs : Region-based CNNs

R-CNN family of networks :

- **R-CNN**
- Fast-RCNN
- Faster-RCNN

R-CNNs

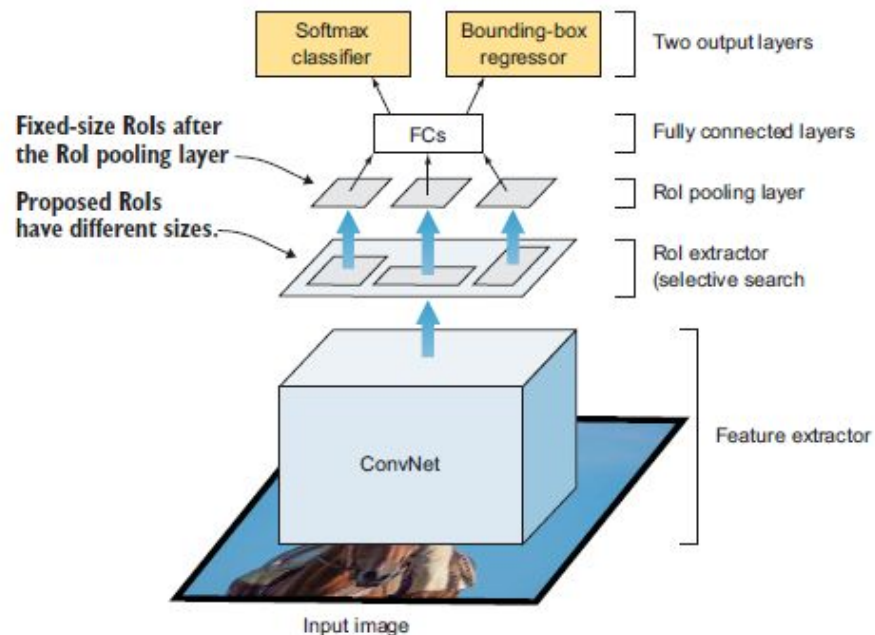


Fast R-CNN

Improves on R-CNN in both detection **speed** and **accuracy**.

Architecture changes:

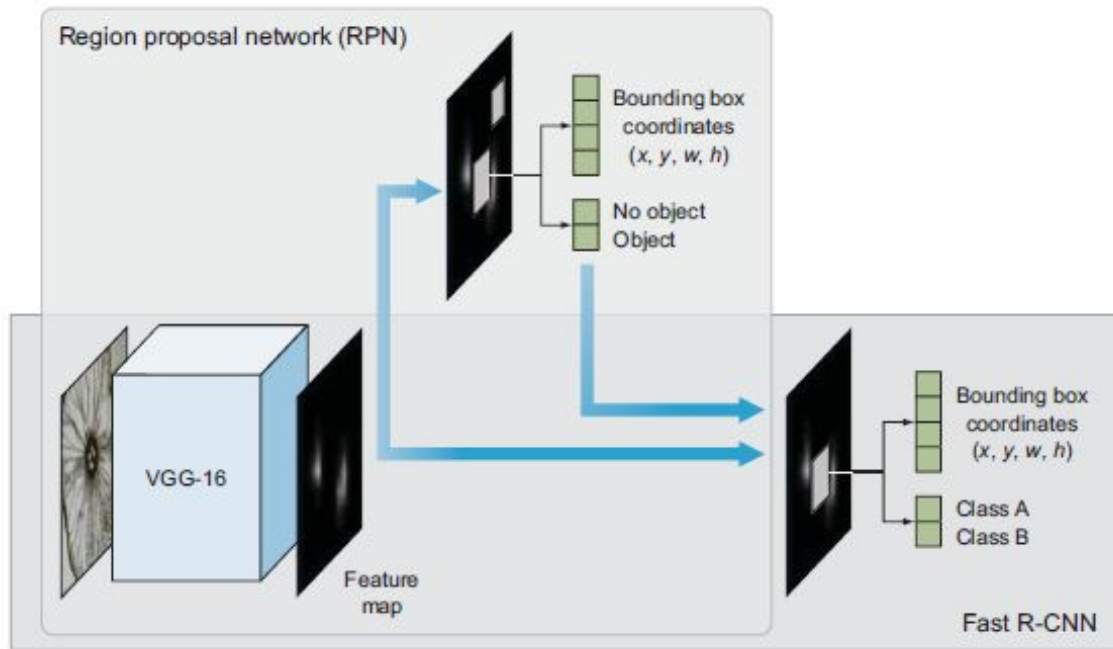
- CNN feature extractor first applied to entire image, then extract features for the proposed Rols
 - Only run one CNN instead of ~2000 CNNs on overlapping Rols
- (C)DNN performs **both** the **classification** and **feature extraction**
 - Feature extractor trainable also (initialized from a pretrained model)
 - SVM machine replaced with a softmax layer



Faster R-CNN

Architecture

- Same overall structure as Fast R-CNN except for **region proposal** algorithm
- Selective search replaced with **region proposal network**, which outputs
 - Objectness score
 - Bounding box location



So now the whole pipeline is trained end-to-end

Multi-stage vs single-stage detectors

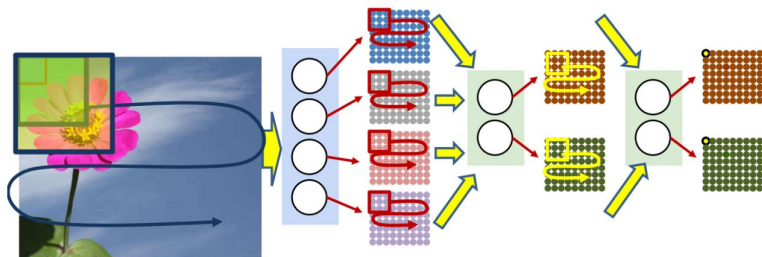
Multistage detectors:

- Two separate components: (sparse) Rols proposal & final prediction on Rols
- Slow but more accurate

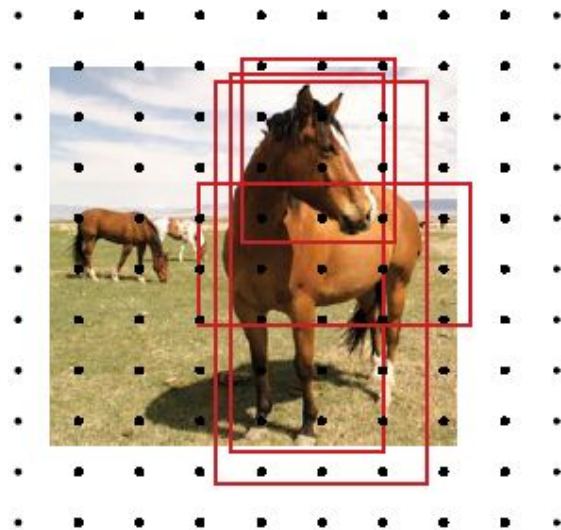
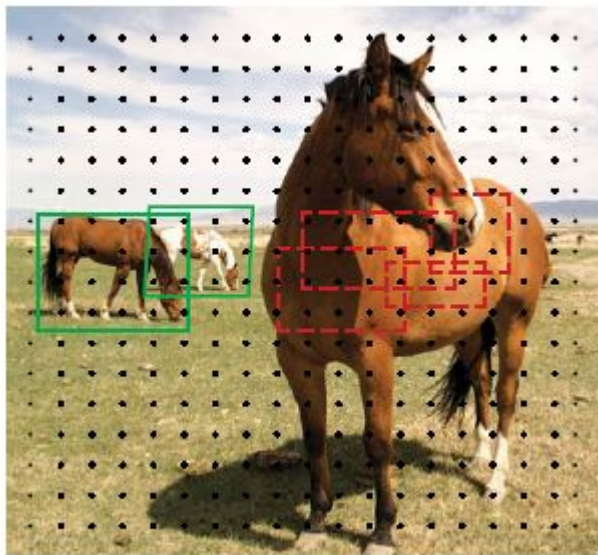
Single-stage detectors:

- No Rol proposal stage: direct predictions on densely sampled Rols
- Fast but less accurate

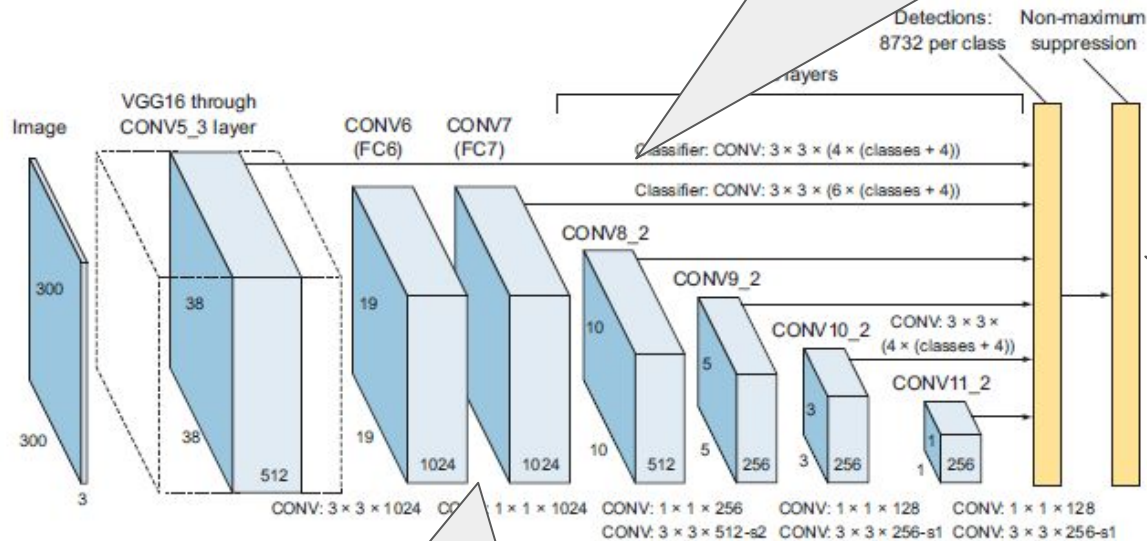
Effective multi-scale RoI grids & Rols



size of receptive field



SSD (Single-shot detector)

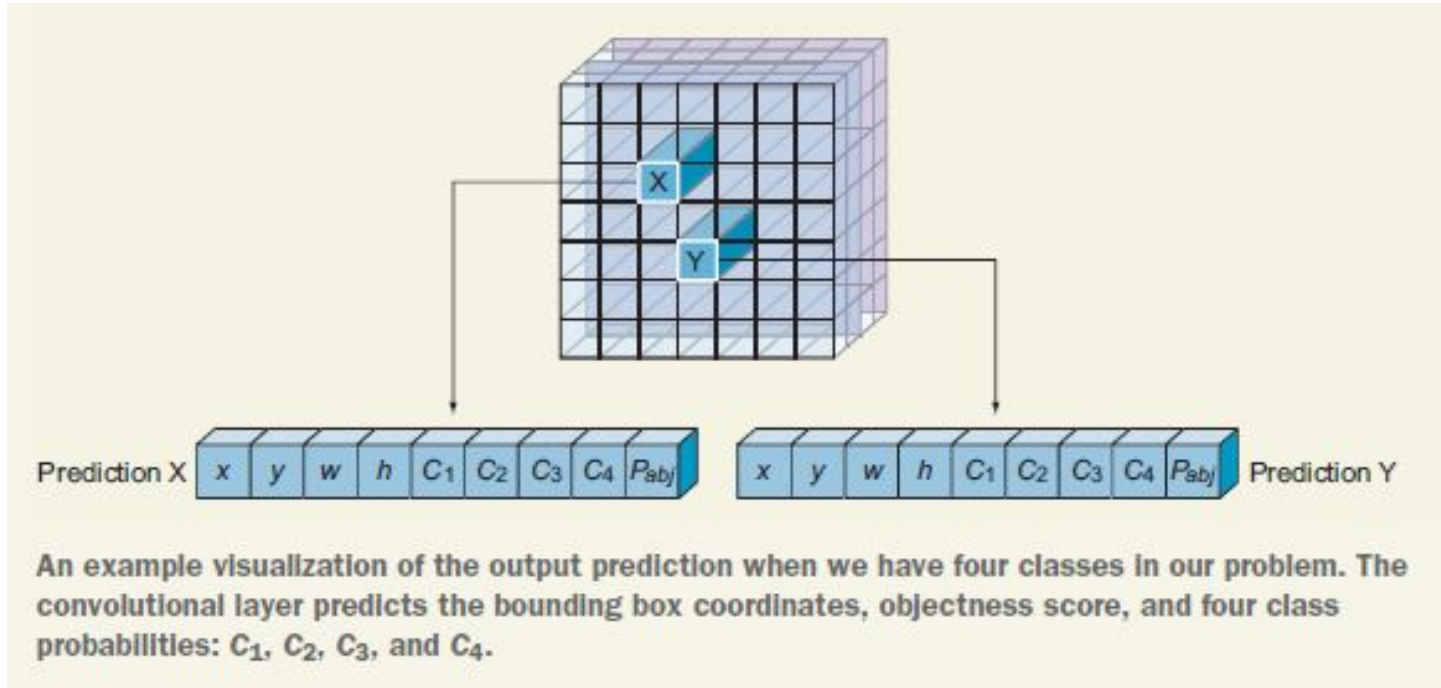


Conv to directly make prediction on dense grid of the feature maps, corresponding to dense sampled RoIs (receptive fields) of various scales over the original image

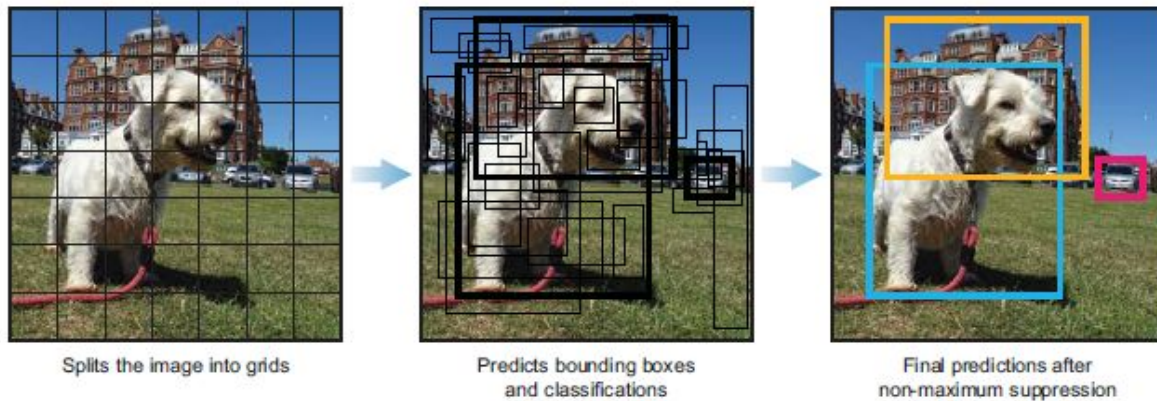
Conv layers for hierarchical feature extraction

NMS to reduce overlaps

SSD: prediction for each feature location



YOLO: Real-time Object Detection



- No region proposal network
- Performs predictions based on a grid of cells (**sacrifice accuracy for speed**)
- Each cell directly predicts the BB and object class
- NMS yields final prediction

Applications of CNNs in Computer Vision

- Object Detection
- **Segmentation**

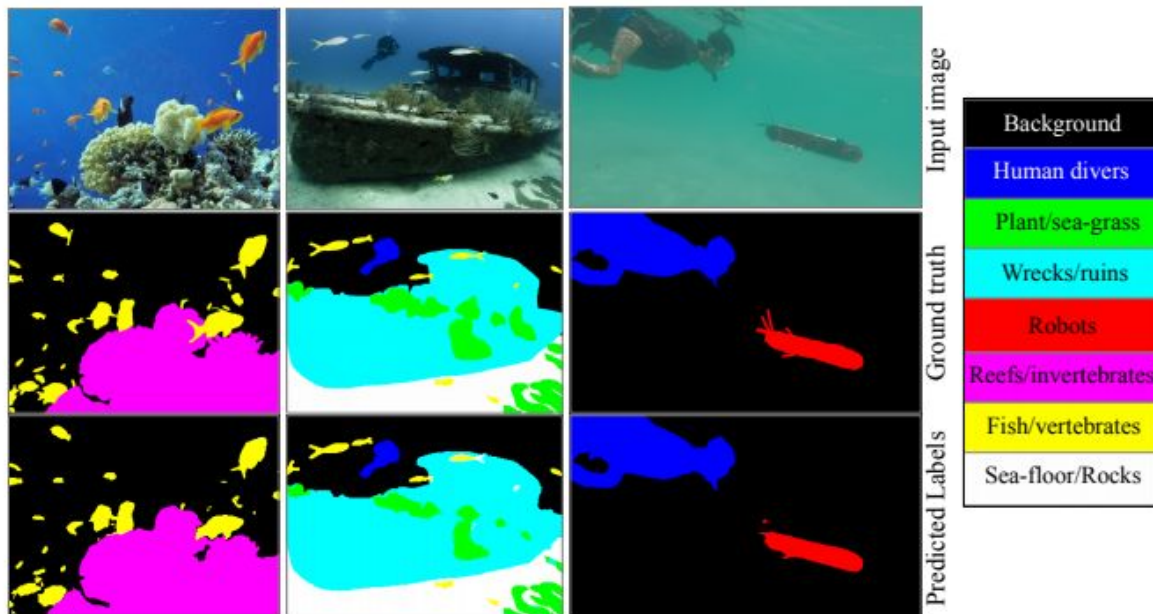
Segmentation

Segmentation: grouping the pixels by their "meanings"

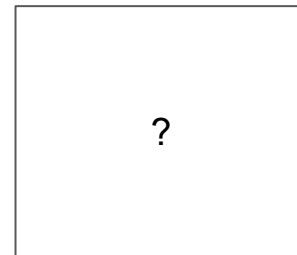
Semantic segmentation: segmentation + assigning a label to each pixel of the image

(Islam et al., "Semantic Segmentation of Underwater Imagery: Dataset and Benchmark," 2020)

Paper from UMN IRVLab: <http://irvlab.dl.umn.edu/>



What is semantic segmentation?

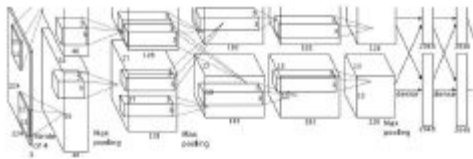


FISH, DIVER, BACKGROUND, AQUATIC
PLANTS, SEAFLOOR

Training data paired: Each pixel
labeled with a semantic category.

During test, classify each pixel of the new
image.

End-to-end learning for semantic segmentation



Intuition: encode the entire image with a CNN, then do semantic segmentation at the end.

Challenge: Classification architectures **reduce feature sizes** as they go deeper into the network; Semantic segmentation requires output size == input.

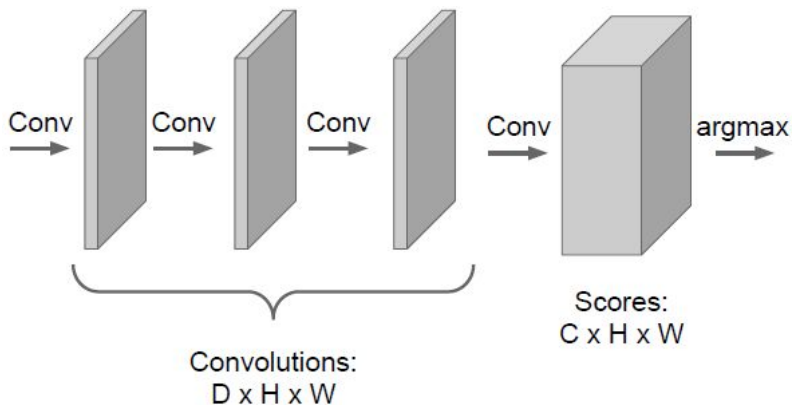
End-to-end learning for semantic segmentation

Challenge: keep the output size the same as that of input

Solution: eliminate any downsampling (e.g., from pooling, strides, etc)



Input:
 $3 \times H \times W$



Predictions:
 $H \times W$

Issue: expensive

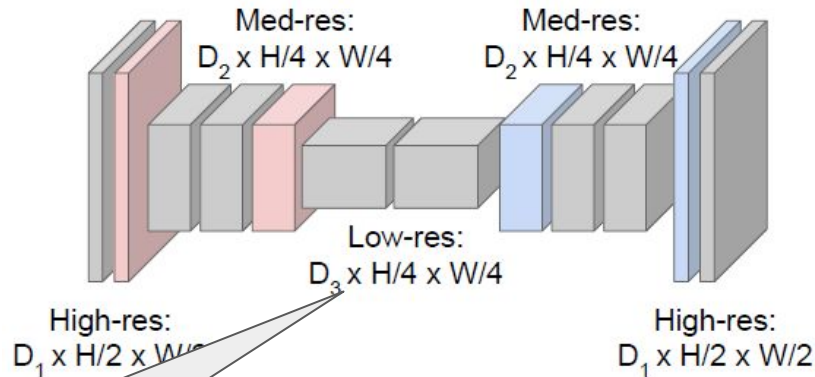
End-to-end learning for semantic segmentation

Issue: using convolution still expensive

Solution: add both downsampling and upsampling inside network!



Input:
 $3 \times H \times W$



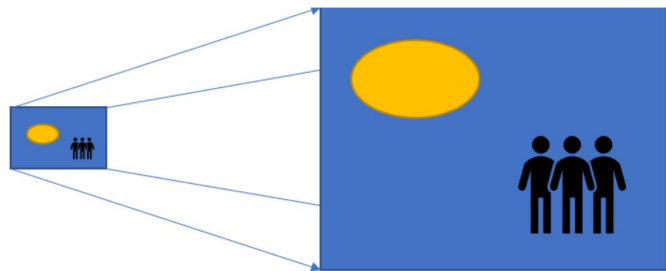
Predictions:
 $H \times W$

Encoder-decoder network,
or bottleneck network

How to do upsampling with convolution?

convolution with strides: downsampling

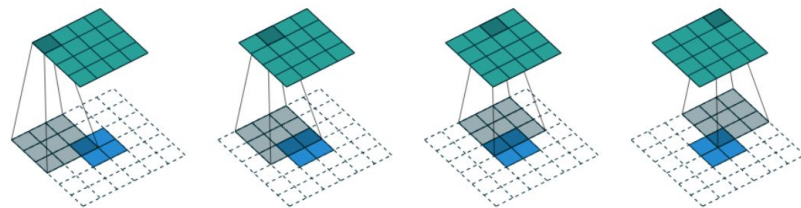
transposed convolution: upsampling



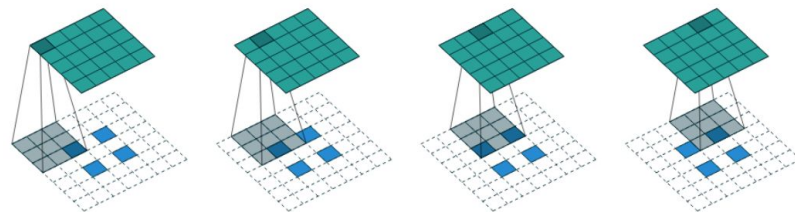
(Credit: <https://naokishibuya.medium.com/>)

often used for segmentation, generation, or other regression—outputs are structured objects such as images, videos, time series, speech, etc

- traditional methods: e.g., nearest neighbor/bilinear/bicubic **interpolation**
- here: interpolation with a **learnable filter**

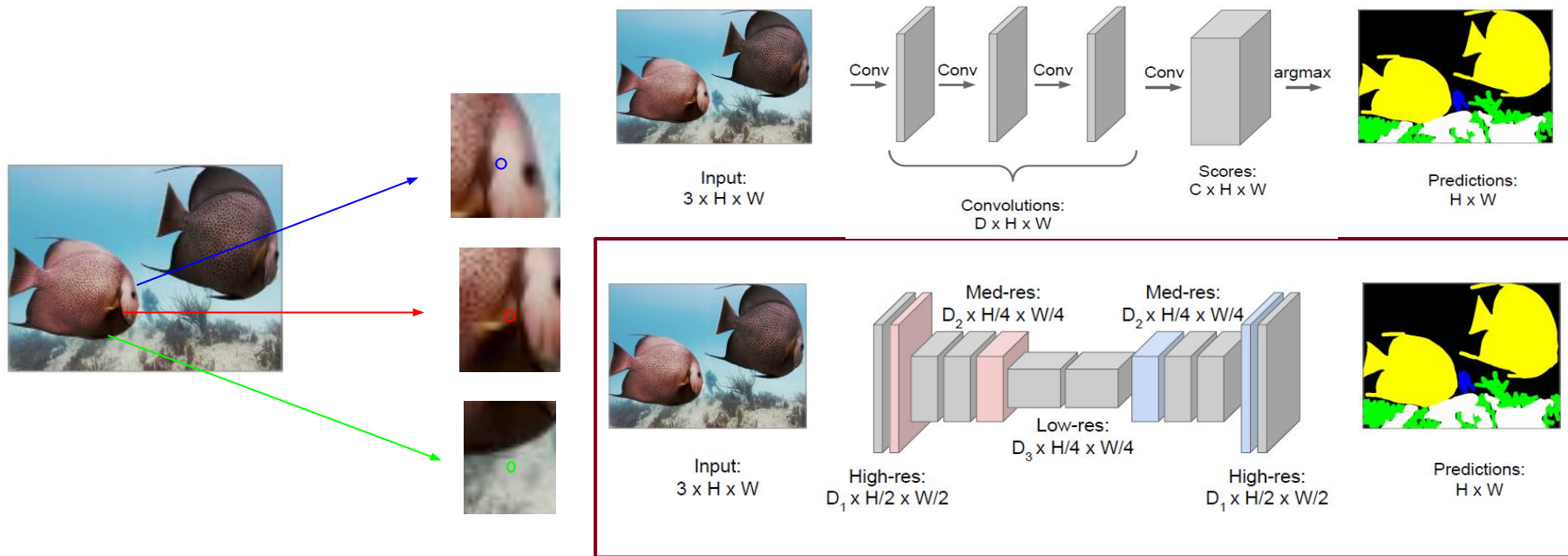


forward stride = 1



forward stride = 2

Semantic Segmentation: Summary

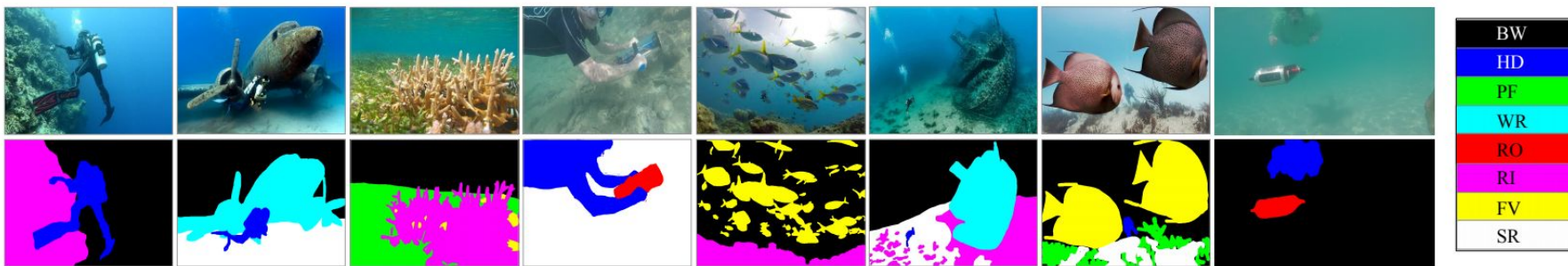


Semantic Segmentation: Summary

Goal: label each pixel in the image with a category label.

Don't differentiate between different instances of the same class of object; only care about the pixel-level.

Object category	RGB color	Code
Background (waterbody)	000	BW
Human divers	001	HD
Aquatic plants and sea-grass	010	PF
Wrecks or ruins	011	WR
Robots (AUVs/ROVs/instruments)	100	RO
Reefs and invertebrates	101	RI
Fish and vertebrates	110	FV
Sea-floor and rocks	111	SR

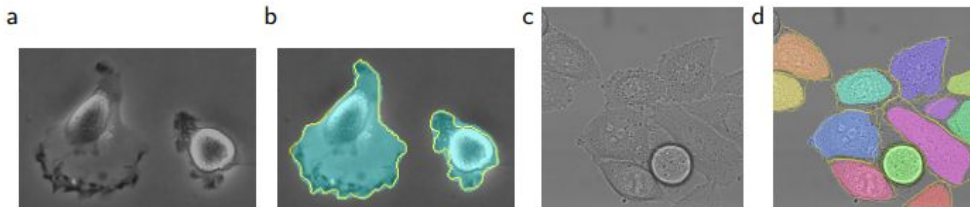
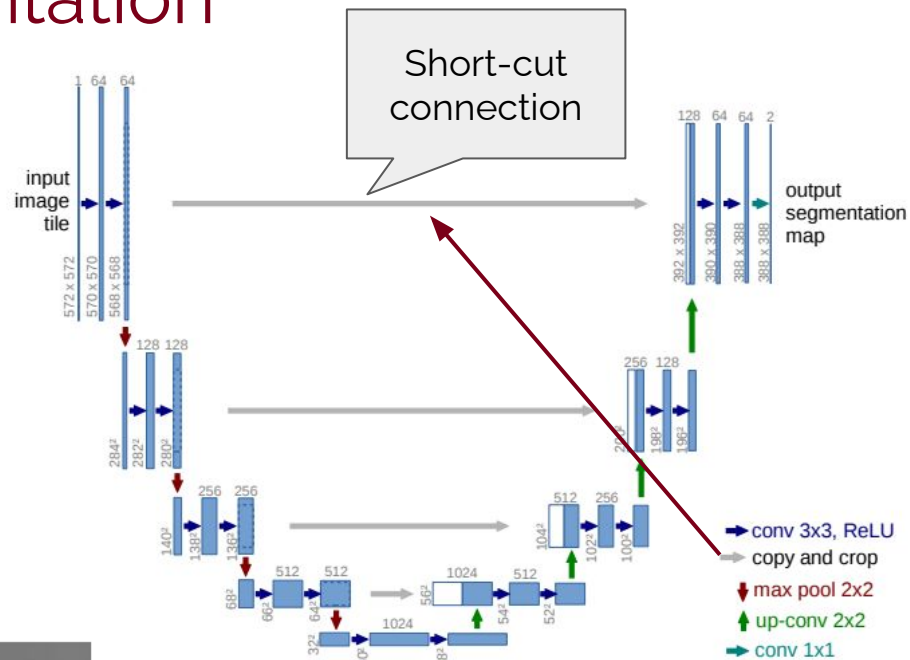


State of the Art Segmentation CNNs



UNET: Semantic Segmentation

- Very popular in medical image segmentation, and gradually propagated to other domains also
- Main innovation: adding "shortcut" connections to compensate for information loss, since not all features can be re-created by the decoder



Mask R-CNN for instance segmentation

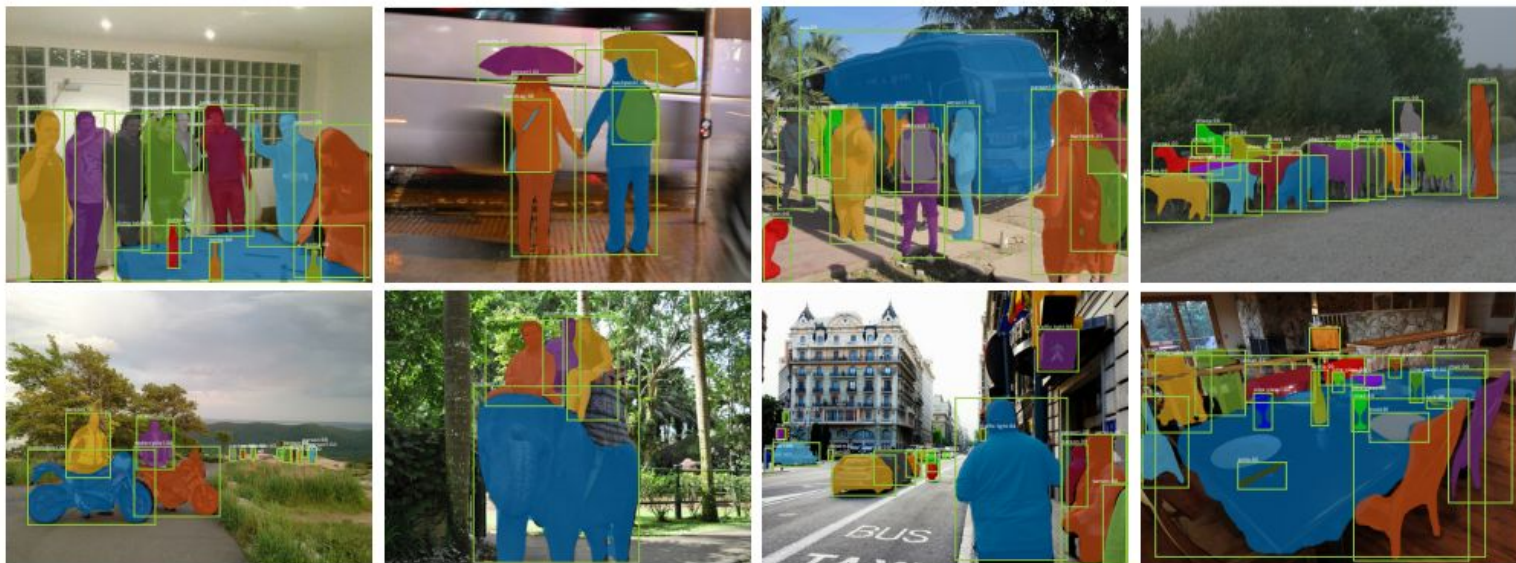


Figure 2. **Mask R-CNN** results on the COCO test set. These results are based on ResNet-101 [19], achieving a *mask AP* of 35.7 and running at 5 fps. Masks are shown in color, and bounding box, category, and confidences are also shown.

DeepLab by Google

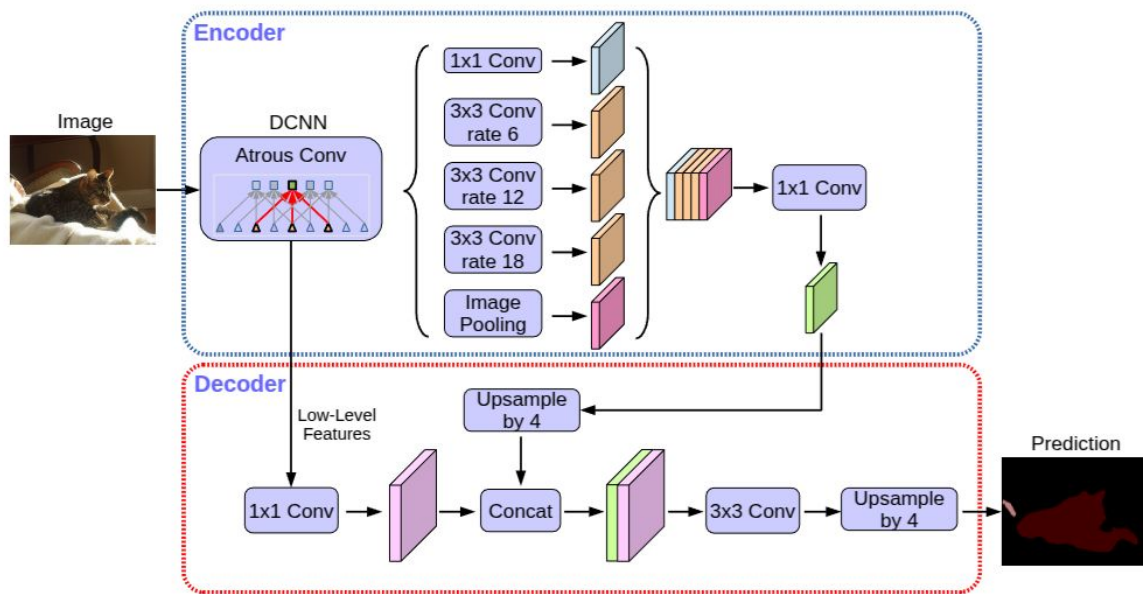
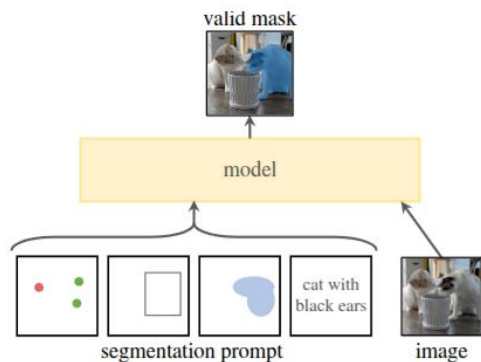


Fig. 2. Our proposed DeepLabv3+ extends DeepLabv3 by employing an encoder-decoder structure. The encoder module encodes multi-scale contextual information by applying atrous convolution at multiple scales, while the simple yet effective decoder module refines the segmentation results along object boundaries.

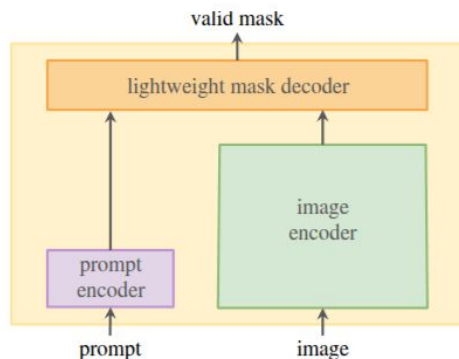
Segment anything model (SAM; by Meta)

Transformer-based

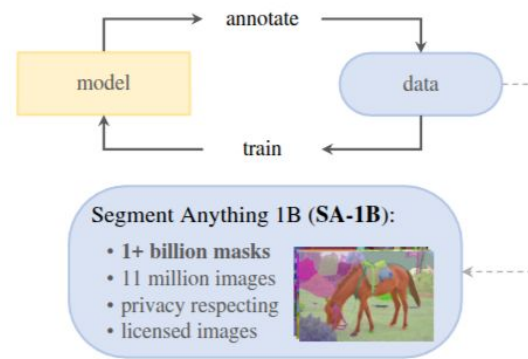
Meta AI Research, FAIR



(a) **Task:** promptable segmentation



(b) **Model:** Segment Anything Model (SAM)



(c) **Data:** data engine (top) & dataset (bottom)

Figure 1: We aim to build a foundation model for segmentation by introducing three interconnected components: a promptable segmentation *task*, a segmentation *model* (SAM) that powers data annotation and enables zero-shot transfer to a range of tasks via prompt engineering, and a *data* engine for collecting SA-1B, our dataset of over 1 billion masks.

<https://ai.meta.com/research/sam3/>

Popular Datasets for Classification, Detection, and Segmentation

- COCO (172 classes, common benchmark dataset)
 - <http://cocodataset.org/#home>
- Cityscapes (roads, lanes vehicles, objects on roads)
 - <https://www.cityscapes-dataset.com/>
- Pascal Context (real-world; over 400 classes)
 - <https://cs.stanford.edu/~roozbeh/pascal-context/>
- Lits (medical imaging, CT scans)
 - <https://competitions.codalab.org/competitions/17094>
- Inria Aerial Image Labeling

Acknowledgements

F.-F. Li, R. Krishna, and D. Xu, "Detection and Segmentation," in *CS231n: Convolutional Neural Networks for Visual Recognition*, 2020.

F.-F. Li, R. Krishna, and D. Xu, "Visualizing and Understanding," in *CS231n: Convolutional Neural Networks for Visual Recognition*, 2020.

M. J. Islam, M. Fulton and J. Sattar, "Toward a Generic Diver-Following Algorithm: Balancing Robustness and Efficiency in Deep Visual Detection," in *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 113-120, Jan. 2019, doi: 10.1109/LRA.2018.2882856.

M. J. Islam, C. Edge, Y. Xiao, P. Luo, M. Mehtaz, C. Morse, S. S. Enan, and J. Sattar, "Semantic Segmentation of Underwater Imagery: Dataset and Benchmark," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE/RSJ*, 2020.

M. Elgendy, "Object Detection with R-CNN, SSD, and YOLO," in *Deep Learning for Vision Systems*, Shelter Island, NY: O'REILLY MEDIA, 2020, pp. 283–337.

K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," 2018.

O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.