Applications of CNNs in Computer Vision: Detection, Segmentation

Ju Sun
Computer Science & Engineering
Mar 25, 2025

University of Minnesota
Driven to Discover℠

Disclaimer

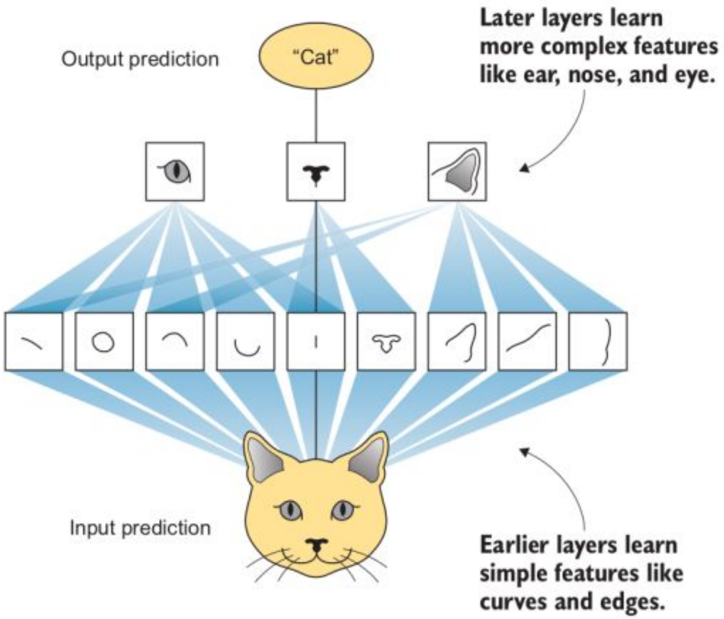
This set of slides are modified from slides made by **Ms. Andrea Walker** in 2020 Fall on the same topic for CSCI8980: Think Deep Learning. The object detection part borrows a lot of materials from the book: “Deep Learning for Vision Systems” by Mohamed Elgendy
https://www.manning.com/books/deep-learning-for-vision-systems

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])

Applications of CNNs in computer vision

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

Hybrid classification-regression problems in CV

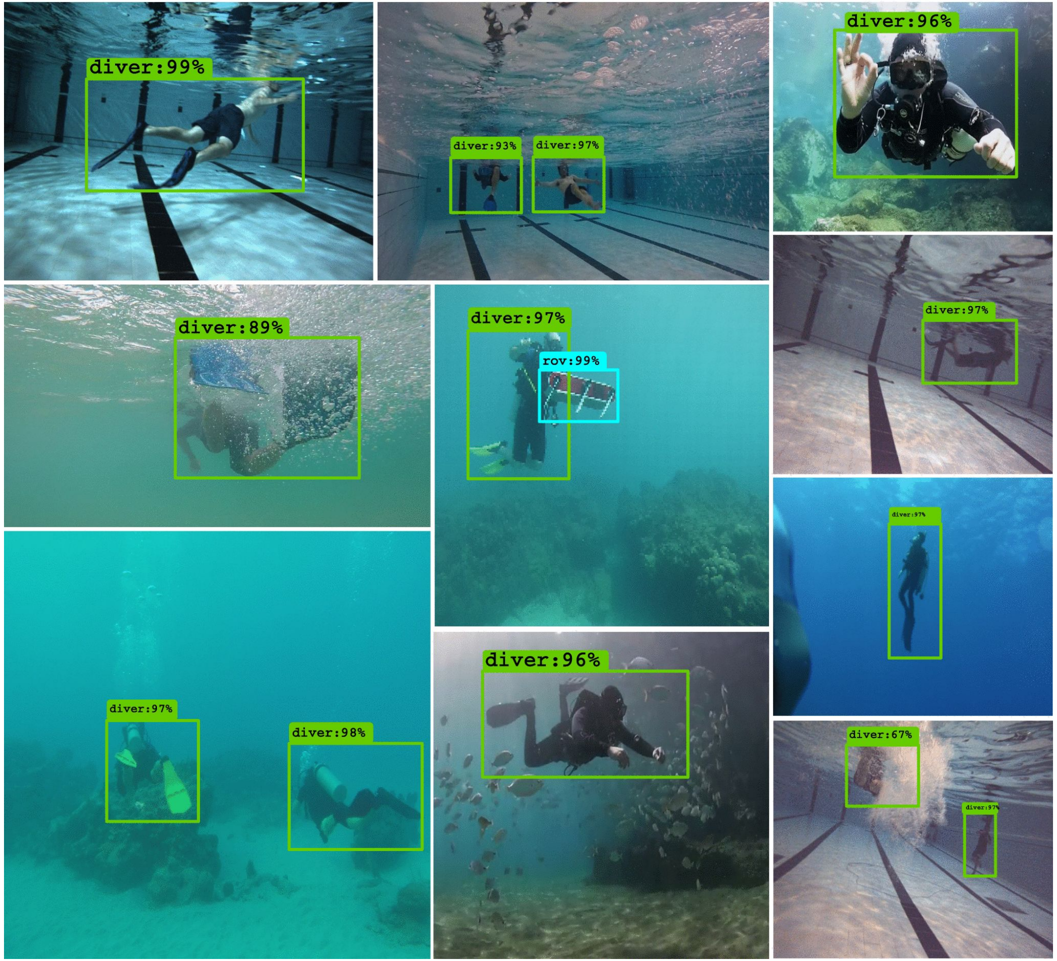
Object detection

- **Localization**:
  where the objects are (by providing bounding boxes)

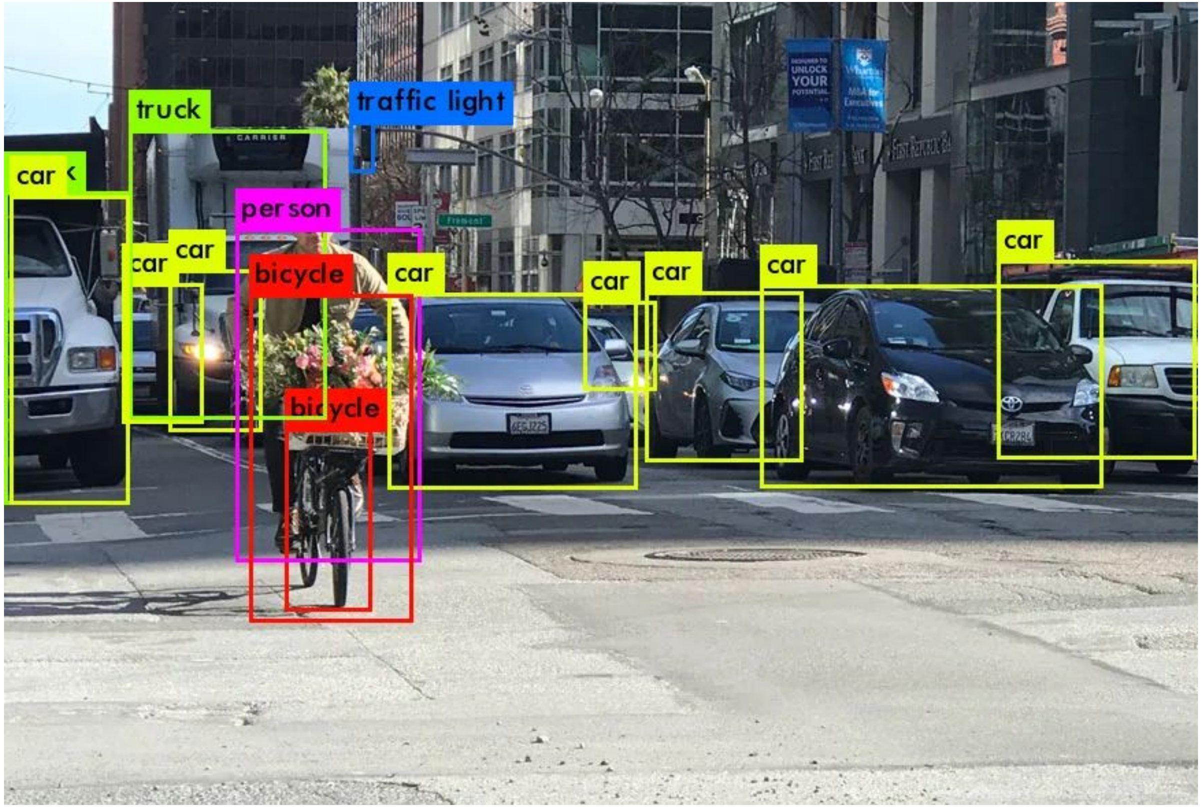
- **Classification**:
  what the objects are (by providing label for each bounding box)

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

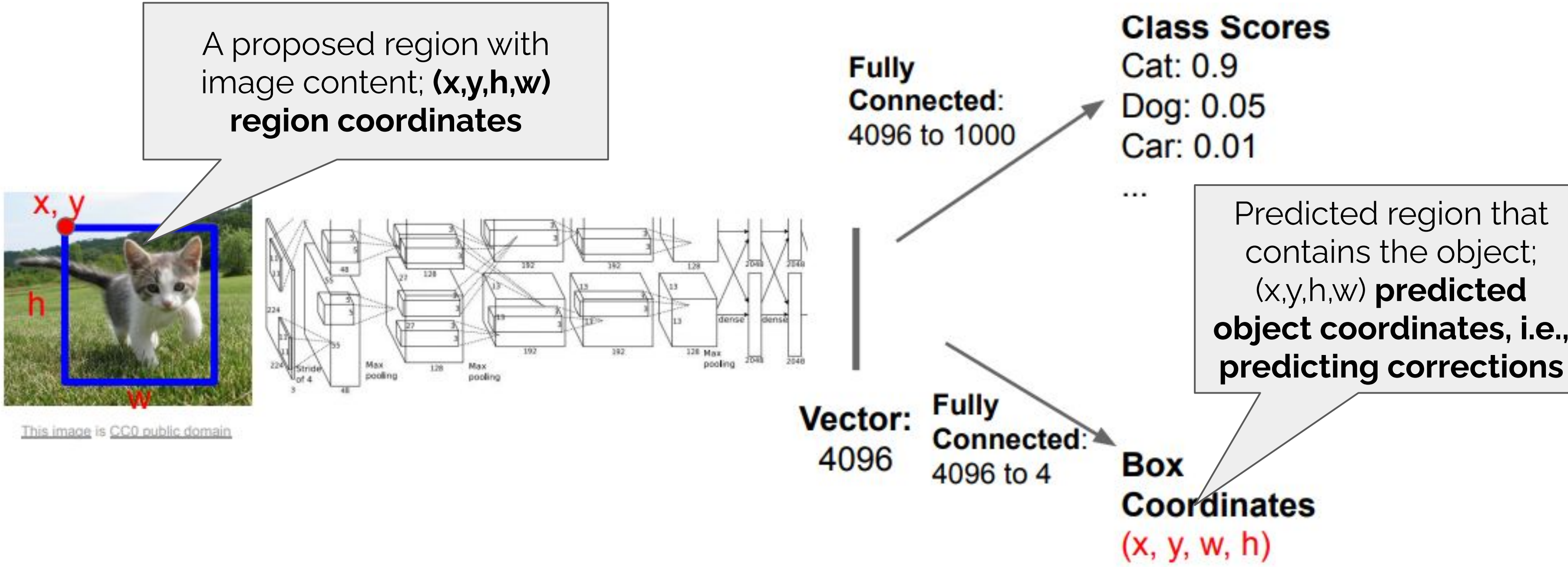
Paper from UMN IRVLab: https://irvlab.cs.umn.edu/



For autonomous driving



Object detection: predictor

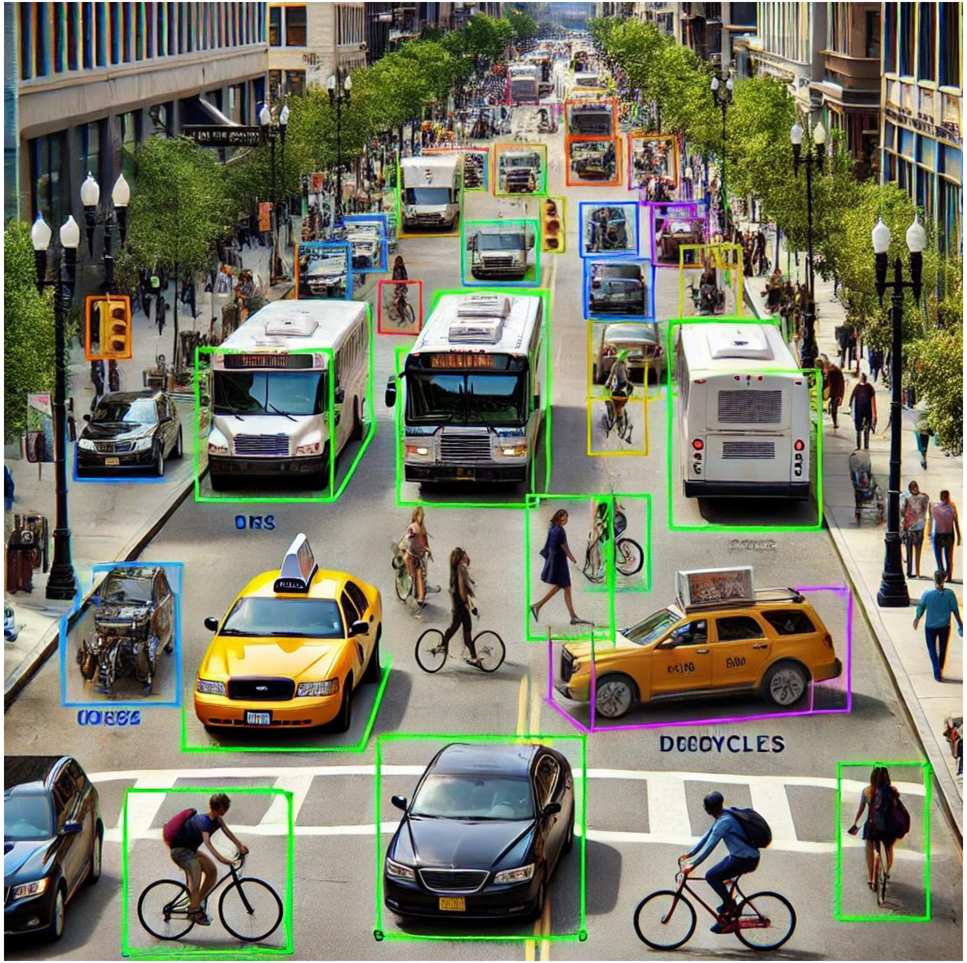


(Li et al., *Detection and Segmentation* 2020)

Training datasets

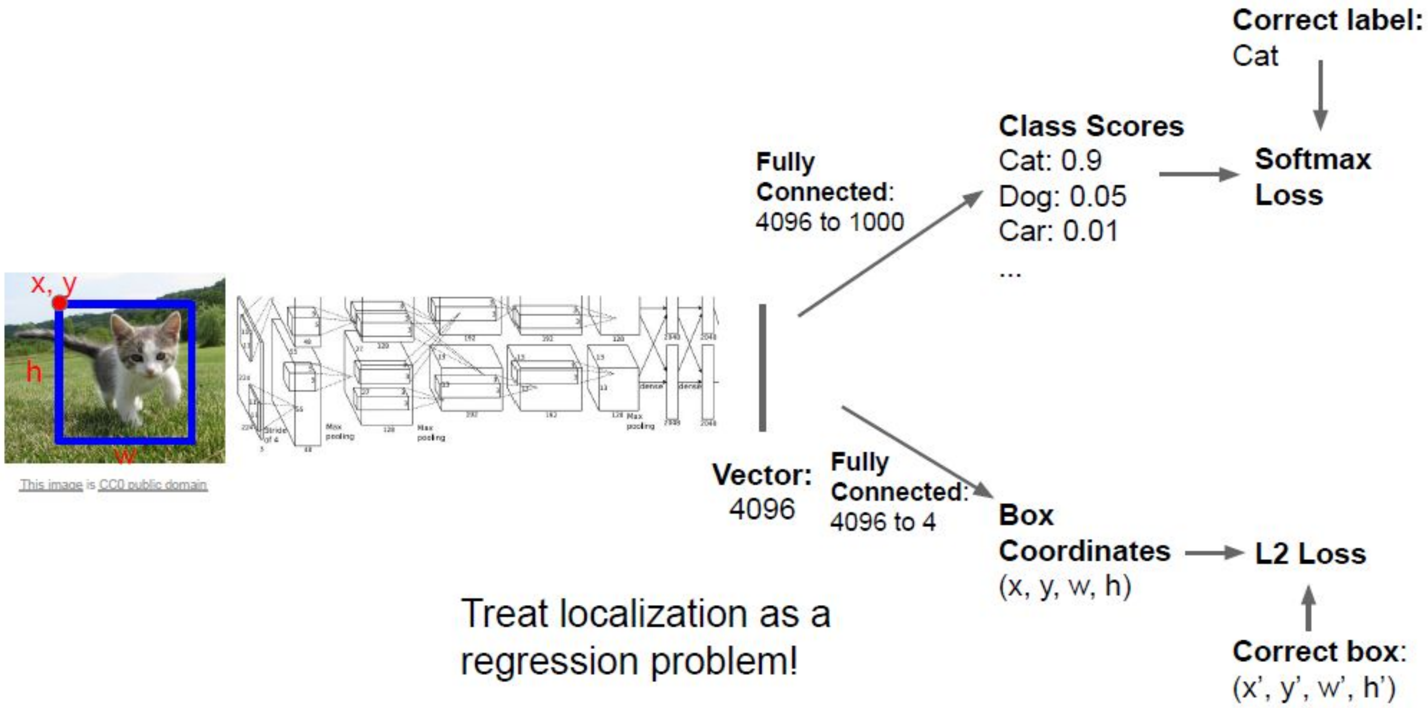
Bounding boxes are stored as

- **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



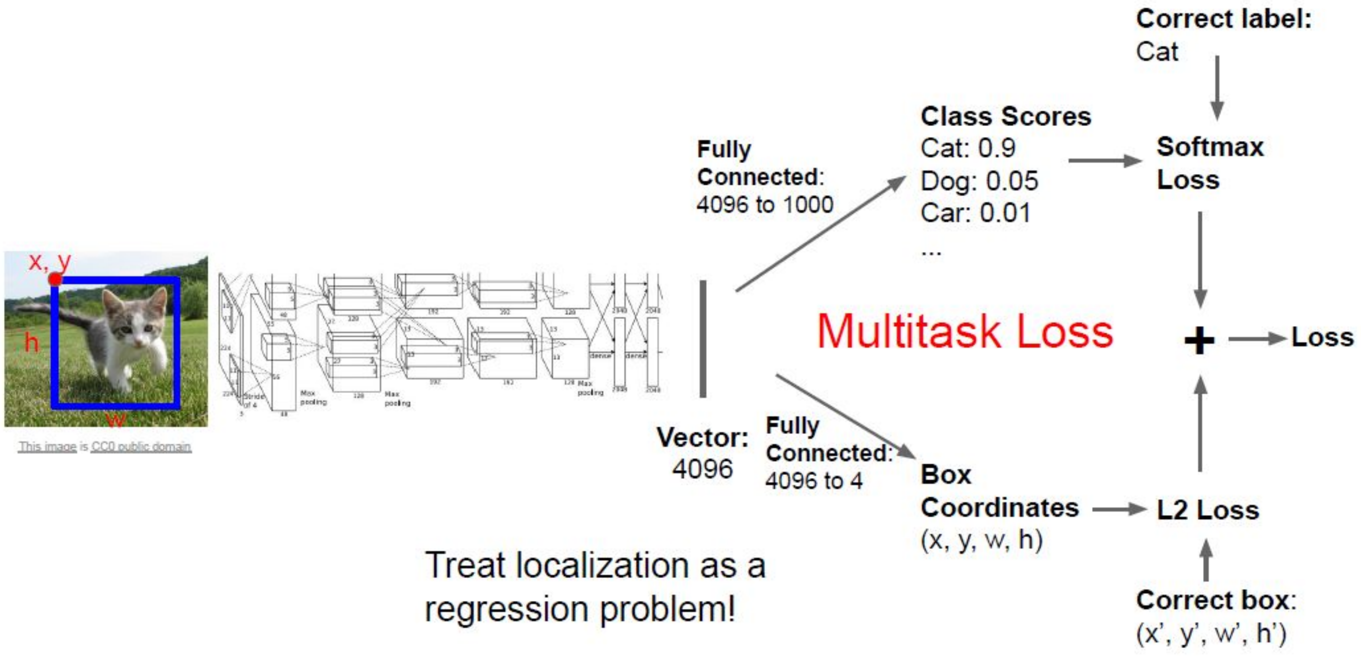
… contain many images with bounding boxes

Object detection: training



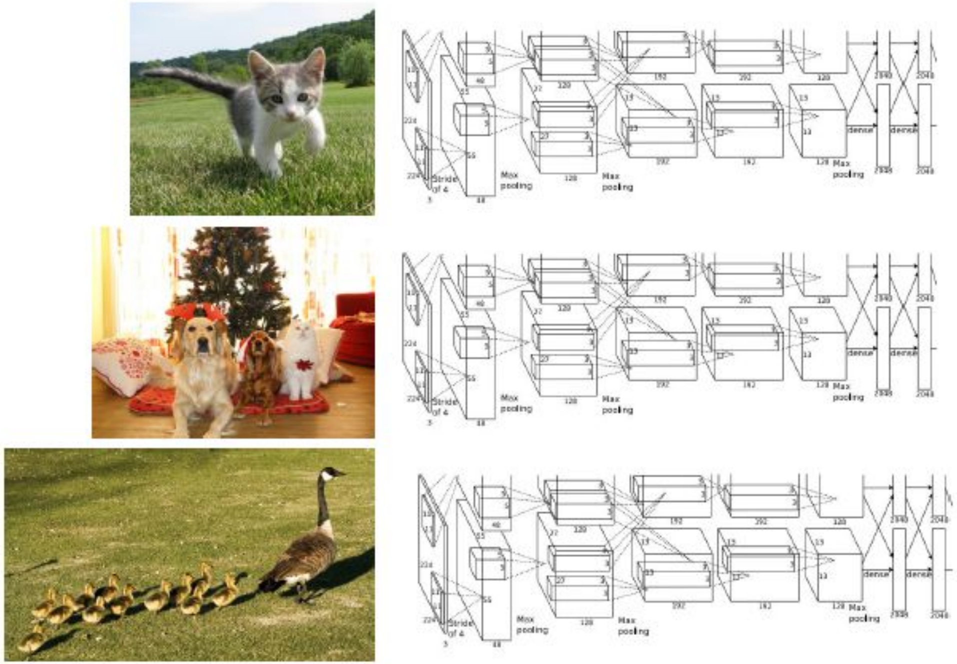
(Li et al., *Detection and Segmentation* 2020)

Object detection: training



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

Multiple objects: multiple outputs



CAT: (x, y, w, h)

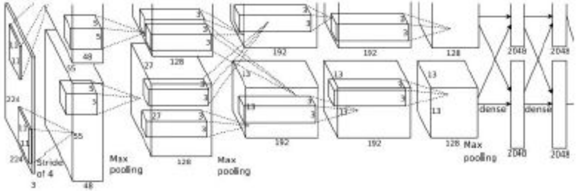
DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)

DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
....

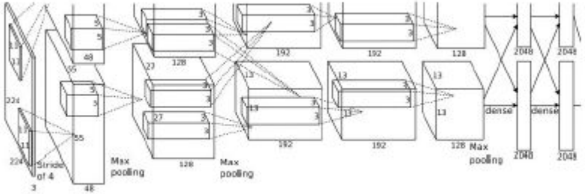
(Li et al., *Detection and Segmentation* 2020)

Multiple objects: initial solution

Scanning window method



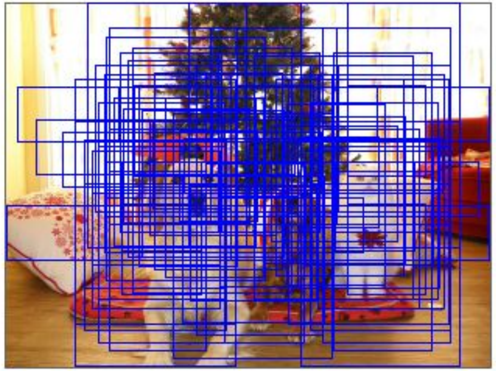
Dog? NO
Cat? NO
Background? YES



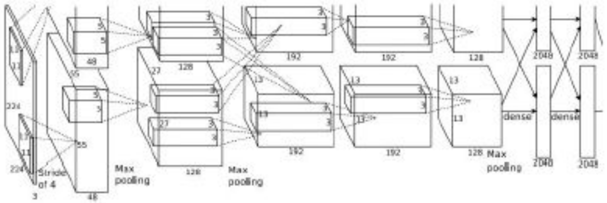
Dog? YES
Cat? NO
Background? NO

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

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!

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

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

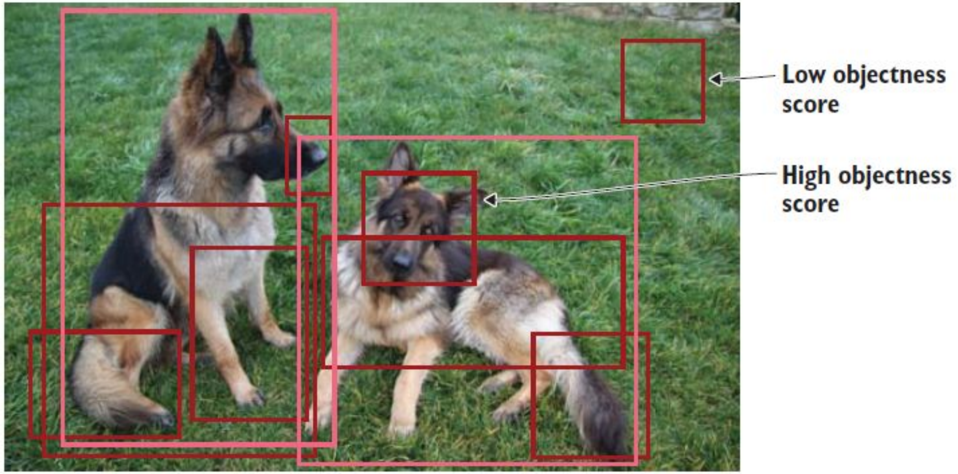
4. **Evaluation metrics:** evaluate performance of model

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



(Elgendy, 2020)

Selective search

Greedy search algorithm for region (blob) proposal

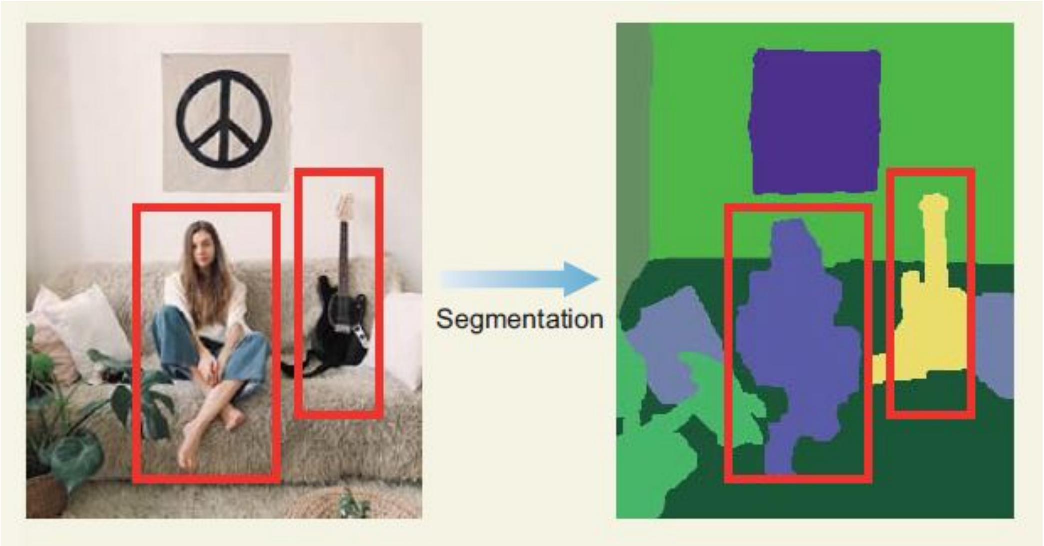
Bottom-up clustering (segmentation)

- Start with many small patches

Repeat:

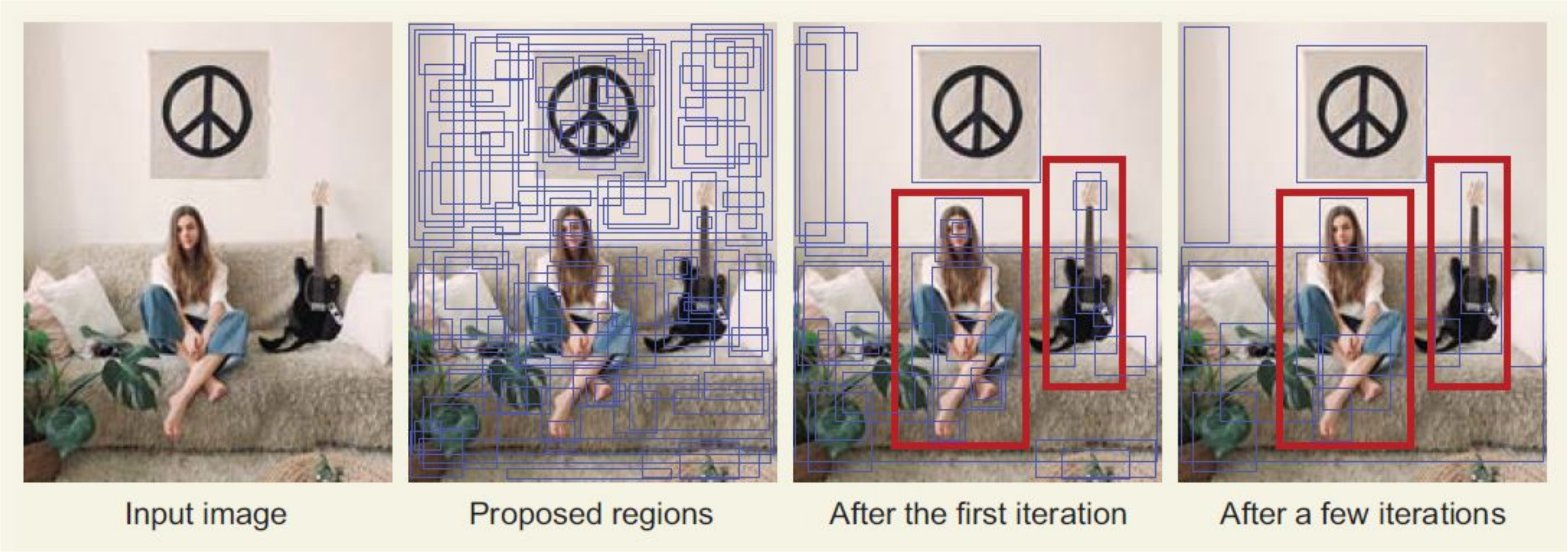
- Most similar patches are merged

Until target #patches reached



(Elgendy, 2020)

Selective search



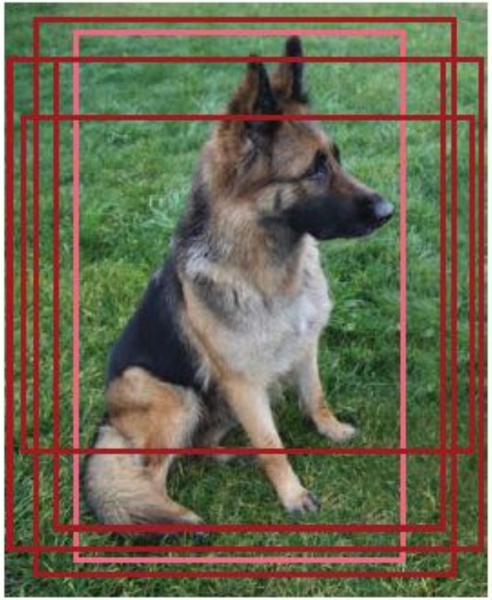
(Elgendy, 2020)

2. Feature extraction & prediction in each RoI

- Extract features using a pretrained CNN

  (Remember **transfer learning**?)

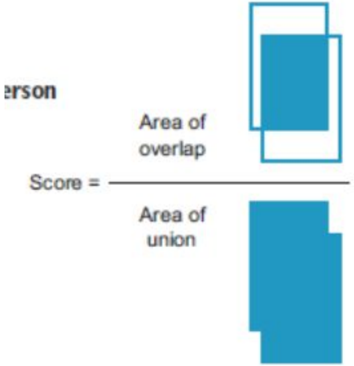
- Make 2 predictions using additional layers:
  - Bounding box prediction (x, y, width, height)
  - Class prediction (softmax function predicting the class probability for each object class)



(Elgendy, 2020)

3. Remove duplicate object detections

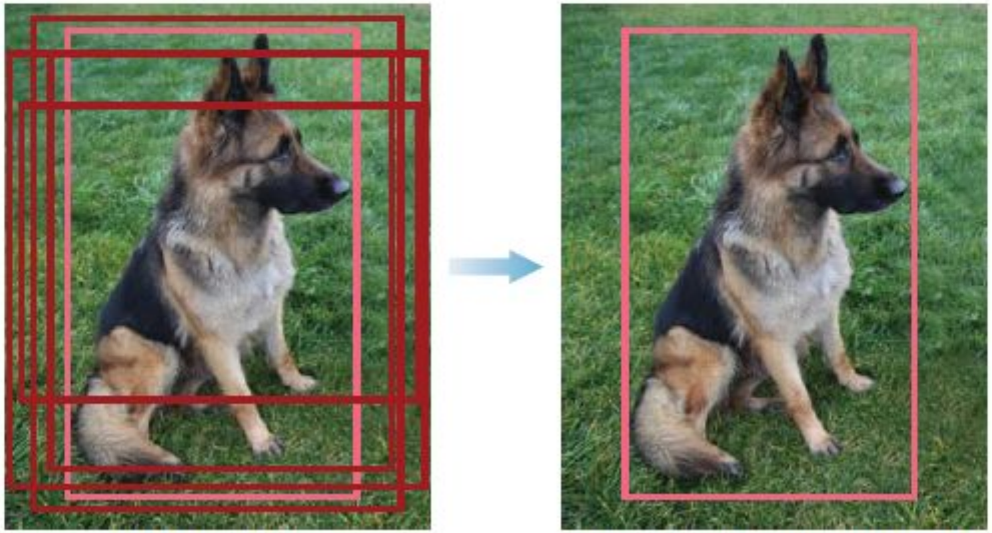
Intersection over Union (IoU)



BB: bounding b

Non-maximum suppression (NMS):
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



Predictions before NMS   After applying non-maximum suppression

(Elgendy, 2020)

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**

(Elgendy, 2020)

Intersection over Union (IoU)

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

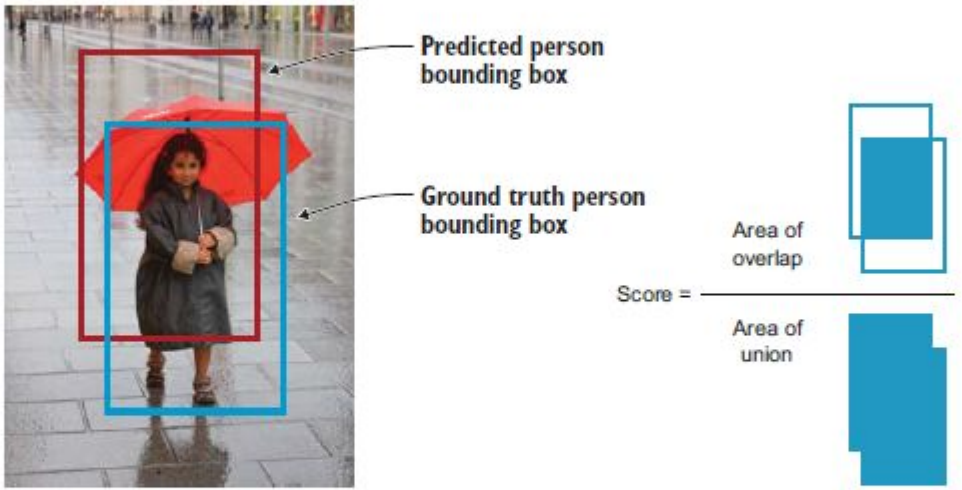


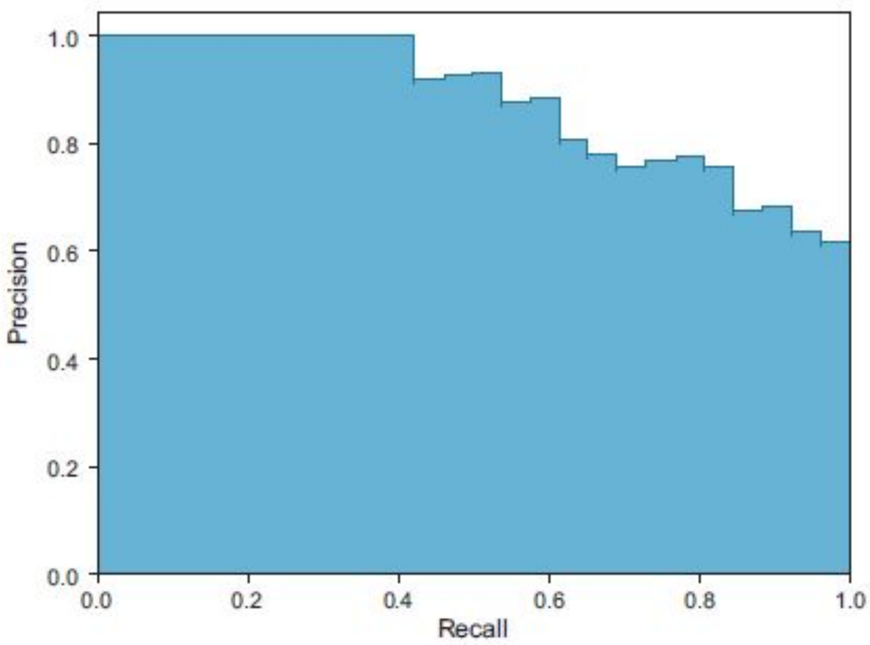
Figure 7.5 The IoU score is the overlap between the ground truth bounding box and the predicted bounding box.

(Elgendy, 2020)

Precision-recall curve and the area under

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

When we vary the IoU threshold …



Can't we use accuracy?

Precision-Recall (PR):

Completeness:

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

Sharpness:

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

(Elgendy, 2020)

State of the Art Object Detection CNNs

- R-CNNs
- SSD
- YOLO

Object Detection

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

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

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

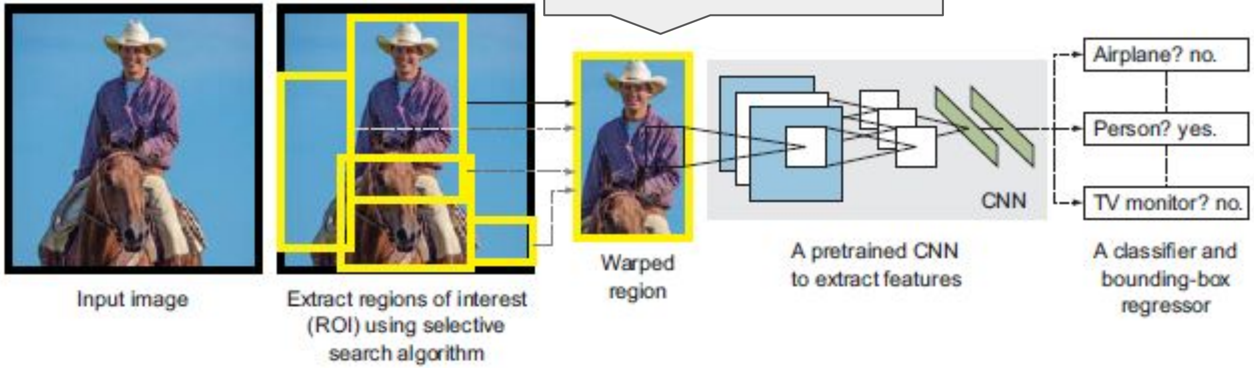
R-CNNs : Region-based CNNs

R-CNN family of networks :

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

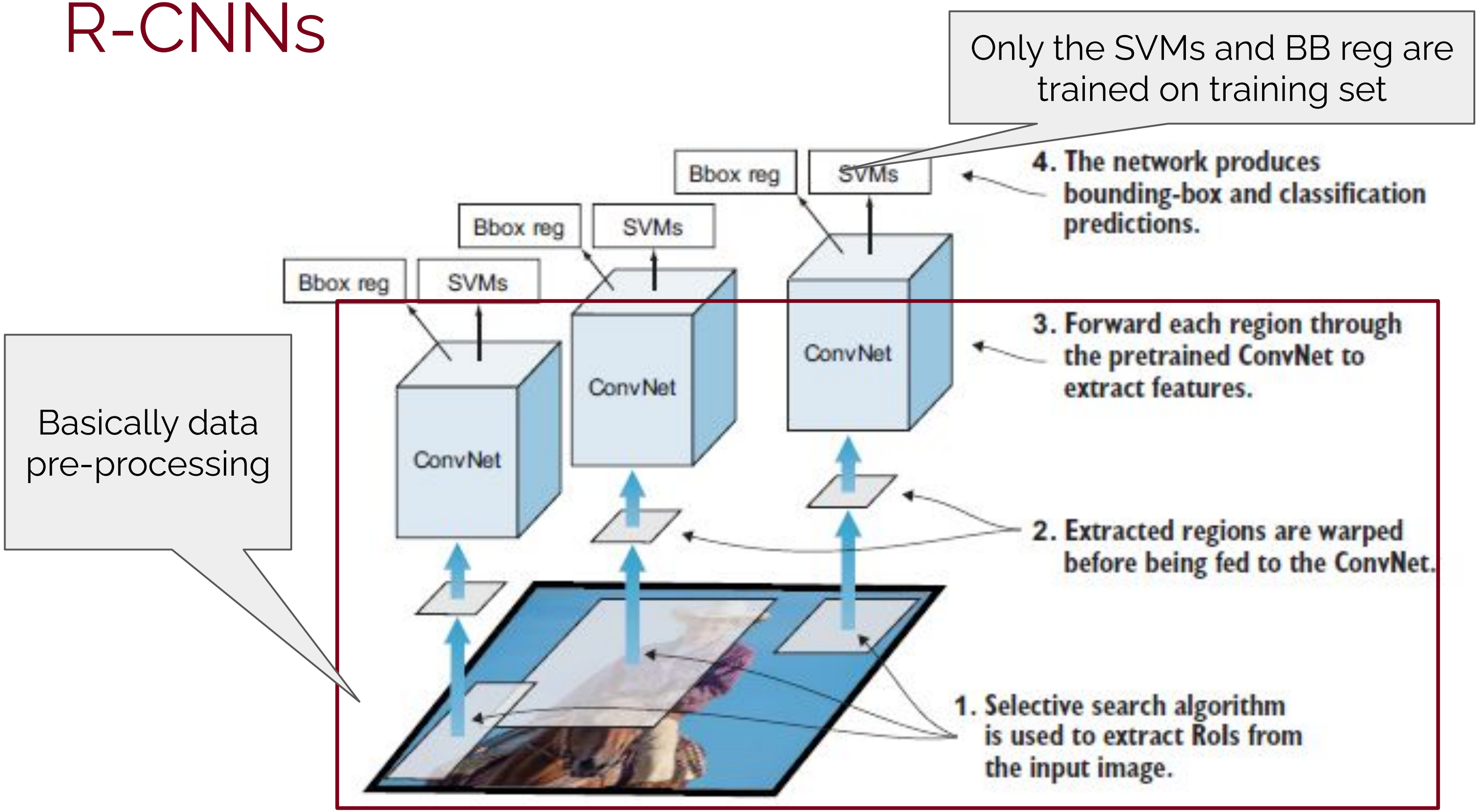
R-CNN architecture

Make ROI's uniform in size



(Elgendy, 2020)

R-CNNs



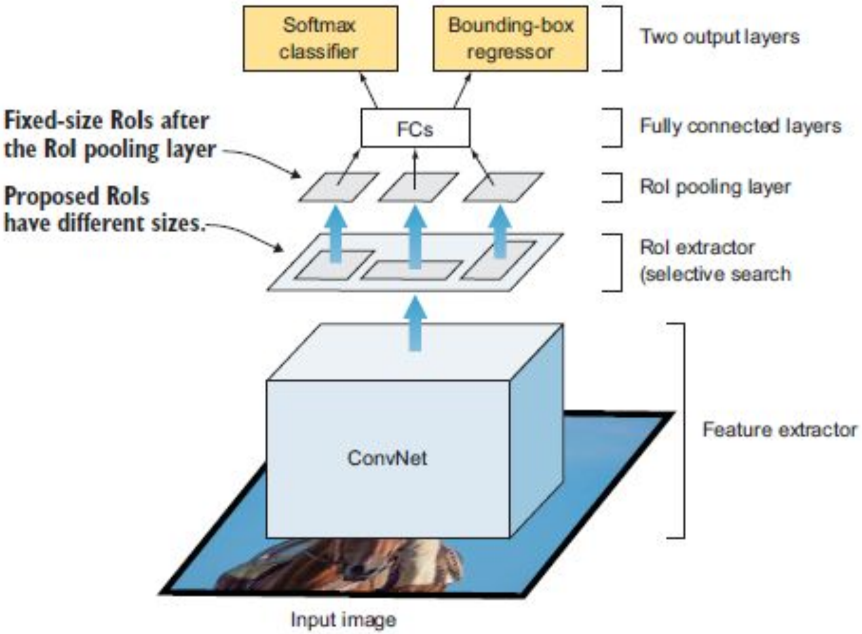
(Elgendy, 2020)

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 RoIs
  - Only run one CNN instead of ~2000 CNNs on overlapping RoIs
- (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



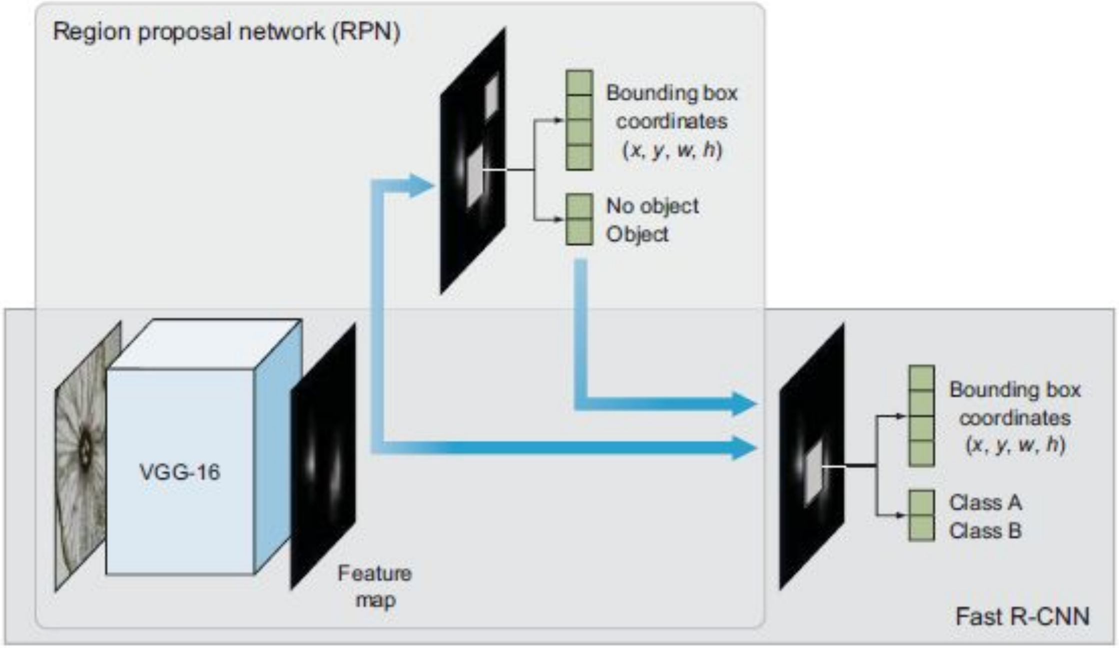
(Elgendy, 2020)

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

(Elgendy, 2020)

Multi-stage vs single-stage detectors

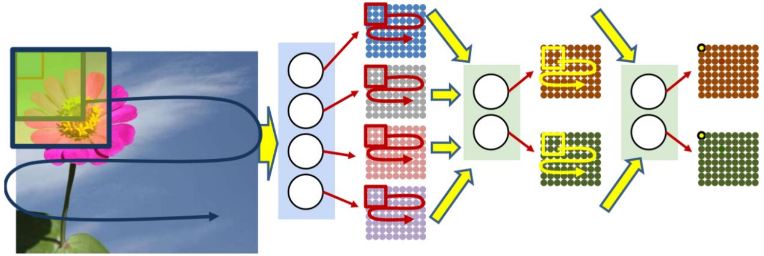
Multistage detectors:

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

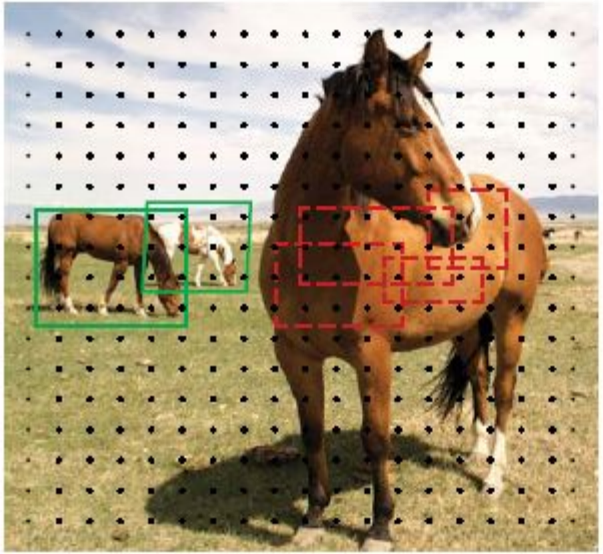
Single-stage detectors:

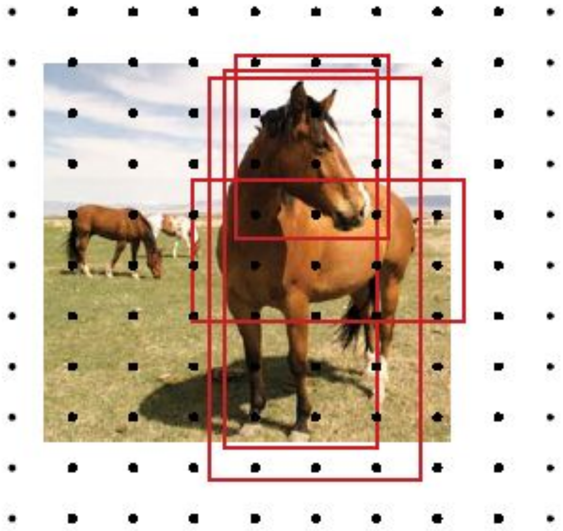
- No RoI proposal stage: direct predictions on densely sampled RoIs
- Fast but less accurate

Effective multi-scale RoI grids & RoIs



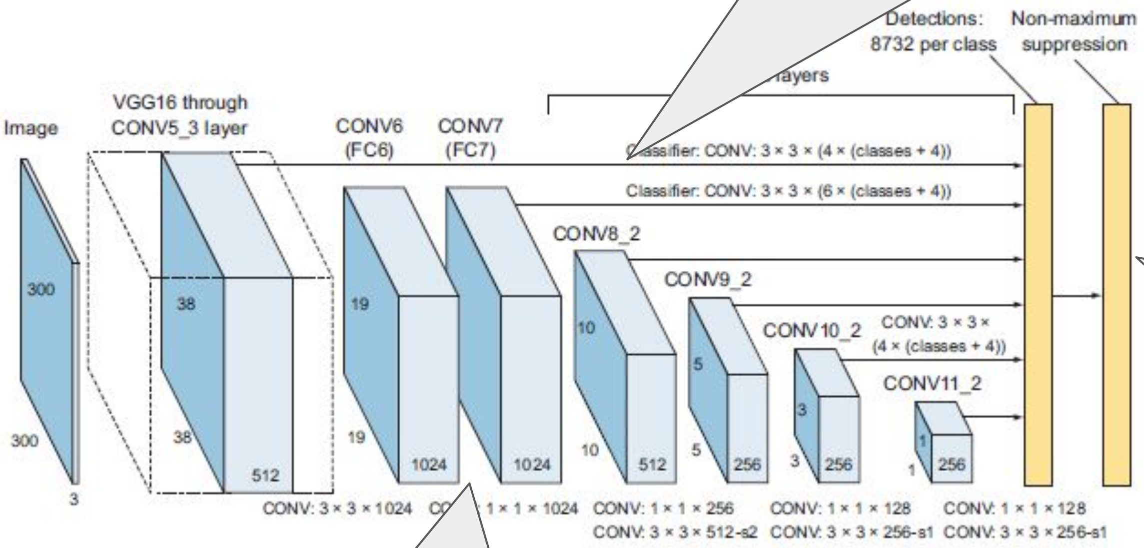
size of receptive field





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

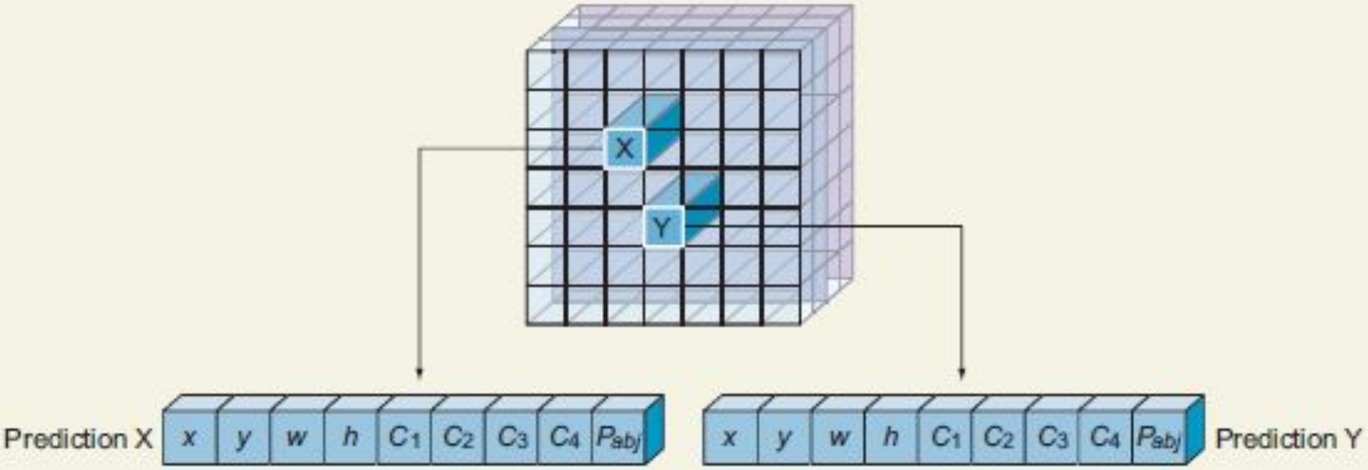


NMS to reduce overlaps

Conv layers for hierarchical feature extraction

(Elgendy, 2020)

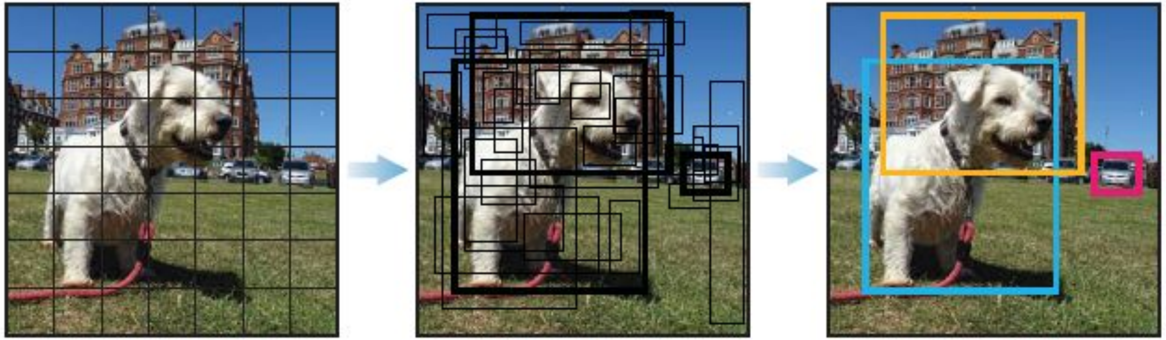
SSD: prediction for each feature location



An example visualization of the output prediction when we have four classes in our problem. The convolutional layer predicts the bounding box coordinates, objectness score, and four class probabilities: C_1, C_2, C_3, and C_4.

(Elgendy, 2020)

YOLO: Real-time Object Detection



Splits the image into grids | Predicts bounding boxes and classifications | Final predictions after non-maximum suppression

- 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

(Elgendy, 2020)

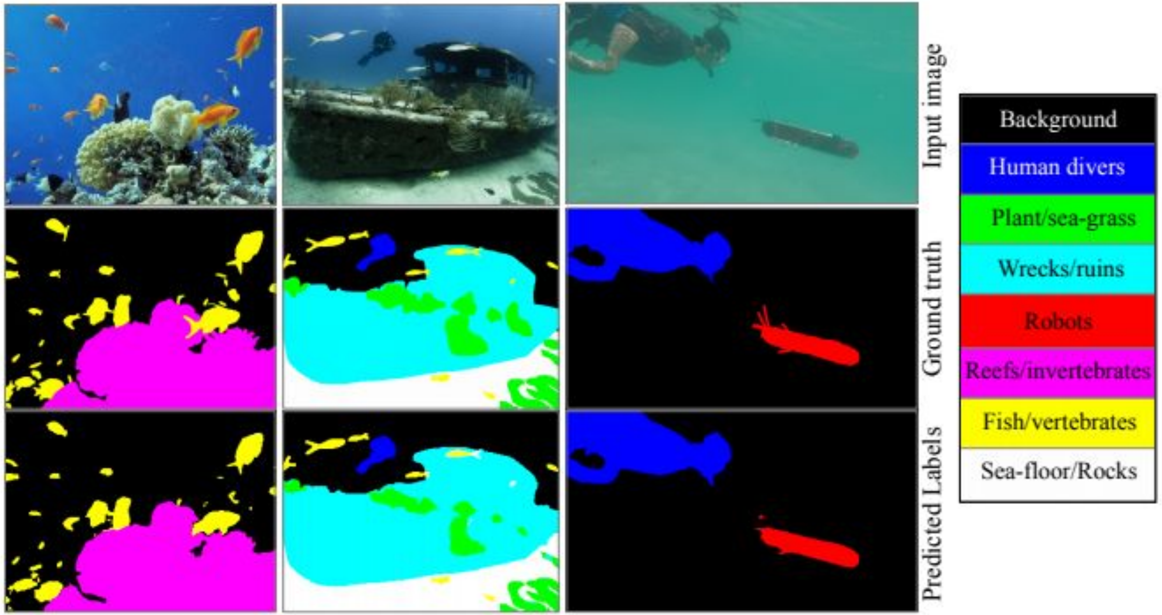
Applications of CNNs in Computer Vision

- Object Detection
- **Segmentation**

Segmentation

Segmentation: grouping the pixels by their "meanings"

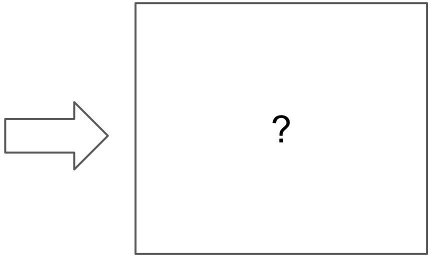
Semantic segmentation: segmentation + assigning a label to each pixel of the image



(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?



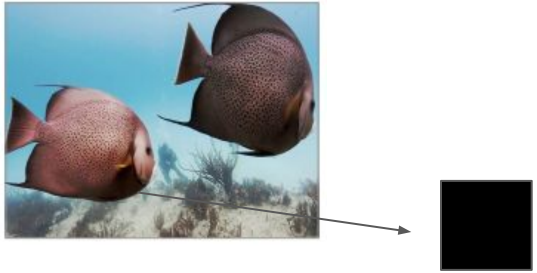
FISH, DIVER, BACKGROUND, AQUATIC PLANTS, SEAFLOOR

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

During test, classify each pixel of the new image.

(Islam et al., "Semantic Segmentation of Underwater Imagery: Dataset and Benchmark," 2020)
Paper from UMN IRVLab http://irvlab.dl.umn.edu/

Semantic Segmentation: Sliding Window

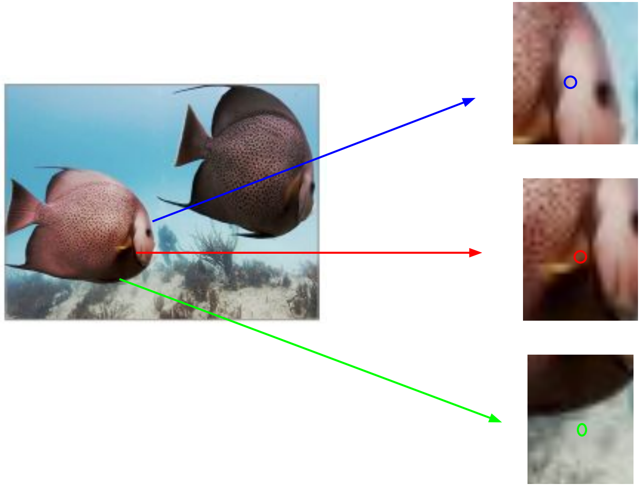


Impossible to classify without context!

How do we include context?

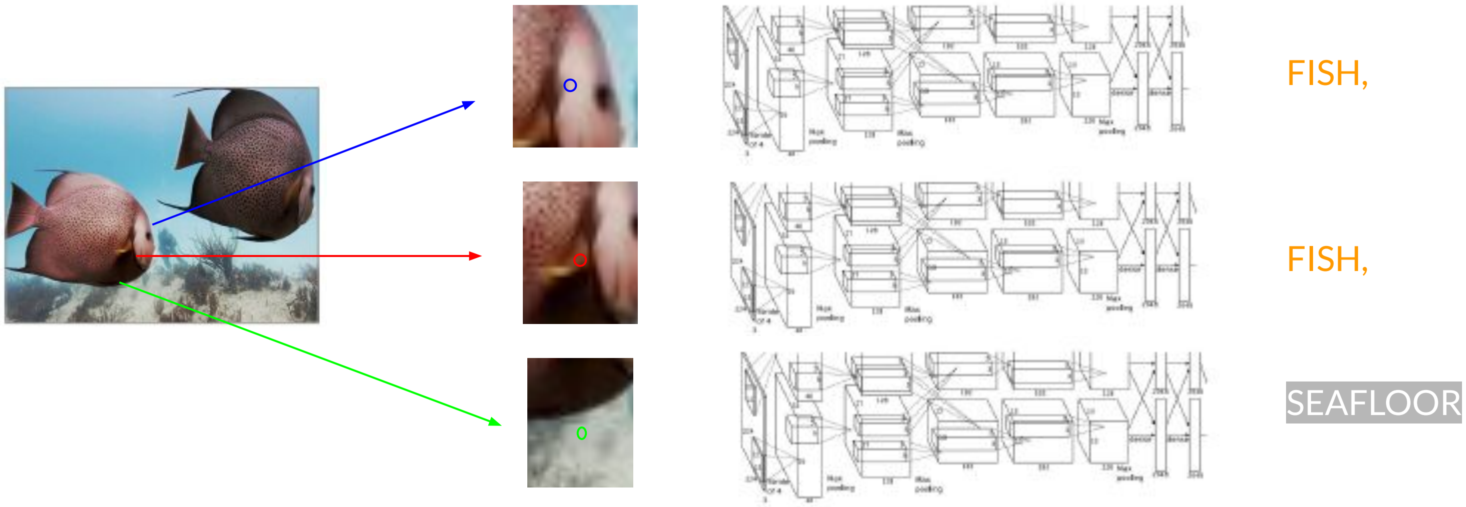
(Islam et al., "Semantic Segmentation of Underwater Imagery: Dataset and Benchmark," 2020)
Paper from UMN IRVLab http://irvlab.dl.umn.edu/

Semantic Segmentation: Sliding Window



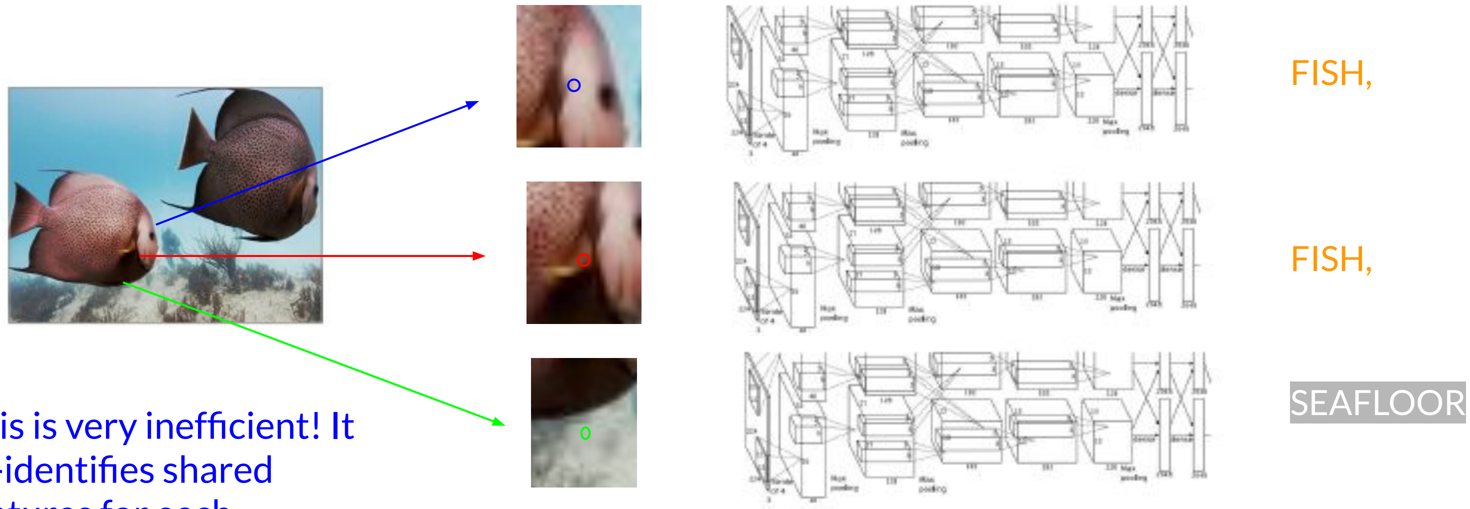
(Islam et al., "Semantic Segmentation of Underwater Imagery: Dataset and Benchmark," 2020)
Paper from UMN IRVLab http://irvlab.dl.umn.edu/
(Li et al., Detection and Segmentation 2020)

Semantic Segmentation: Sliding Window



(Islam et al., "Semantic Segmentation of Underwater Imagery: Dataset and Benchmark," 2020)
Paper from UMN IRVLab http://irvlab.dl.umn.edu/
(Li et al., Detection and Segmentation 2020)

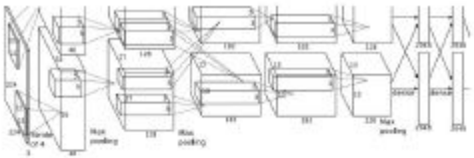
Semantic Segmentation: Sliding Window



This is very inefficient! It re-identifies shared features for each overlapping patch.

(Islam et al., "Semantic Segmentation of Underwater Imagery: Dataset and Benchmark," 2020)
Paper from UMN IRVLab http://irvlab.dl.umn.edu/
(Li et al., Detection and Segmentation 2020)

End-to-end learning for semantic segmentation



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

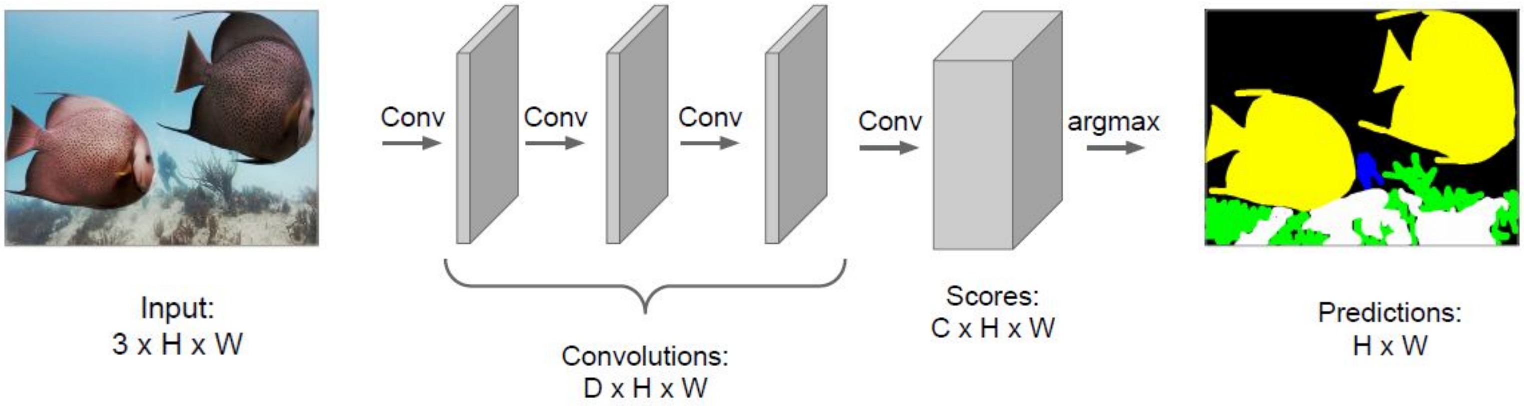
Challenge: Classification architectures **reduce feature sizes** as they go deeper into the network; Semantic segmentation requires output size == input.

(Islam et al., "Semantic Segmentation of Underwater Imagery: Dataset and Benchmark," 2020)
Paper from UMN IRVLab http://irvlab.dl.umn.edu/
(Li et al., Detection and Segmentation 2020)

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)



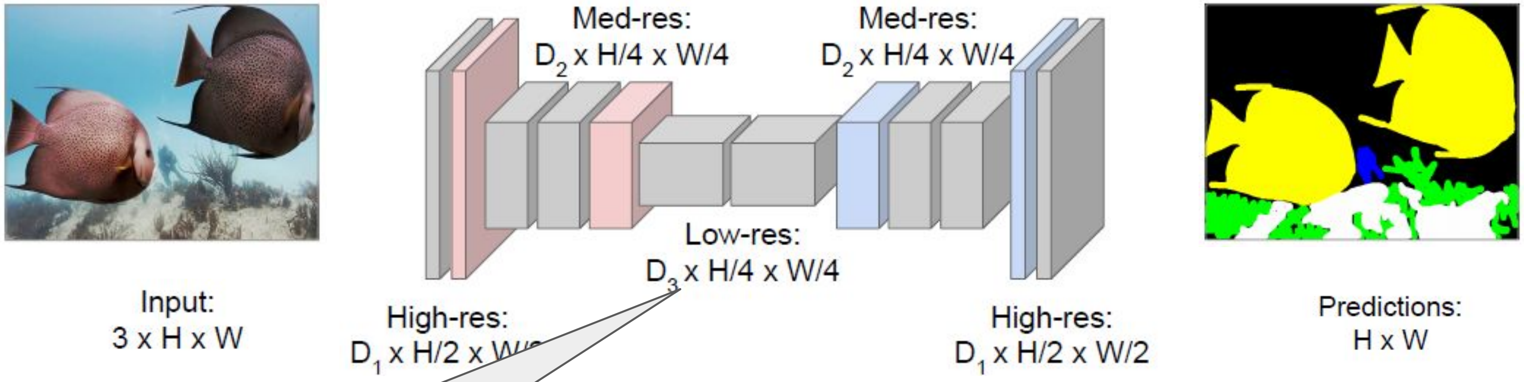
Issue: expensive

(Islam et al., "Semantic Segmentation of Underwater Imagery: Dataset and Benchmark," 2020)
Paper from UMN IRVLab http://irvlab.dl.umn.edu/
(Li et al., Detection and Segmentation 2020)

End-to-end learning for semantic segmentation

Issue: using convolution still expensive

Solution: add both downsampling and upsampling inside network!

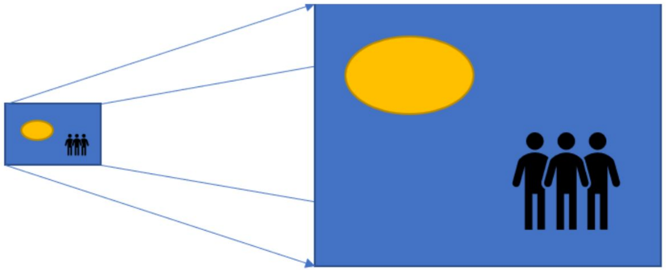


Encoder-decoder network,
or bottleneck network

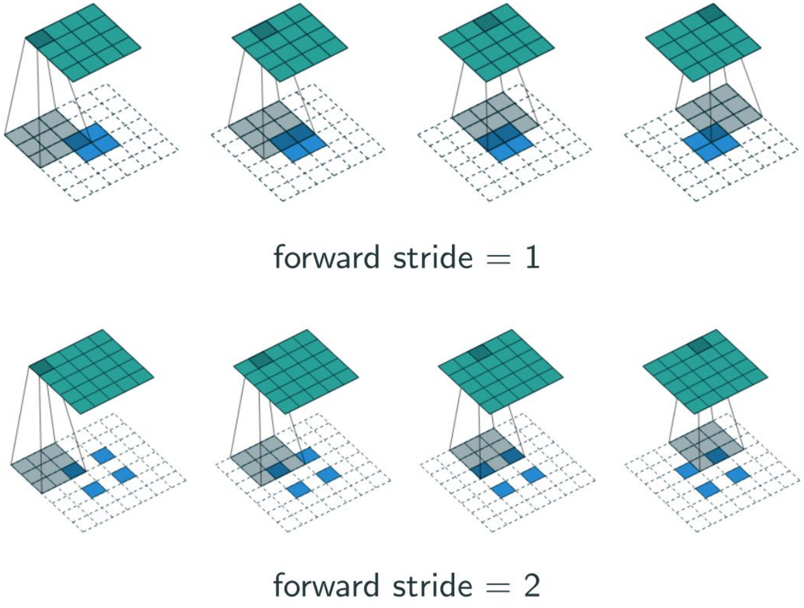
(Islam et al., "Semantic Segmentation of Underwater Imagery: Dataset and Benchmark," 2020)
Paper from UMN IRVLab http://irvlab.dl.umn.edu/
(Li et al., Detection and Segmentation 2020)

How to do upsampling with convolution?

convolution with strides: downsampling
transposed convolution: upsampling



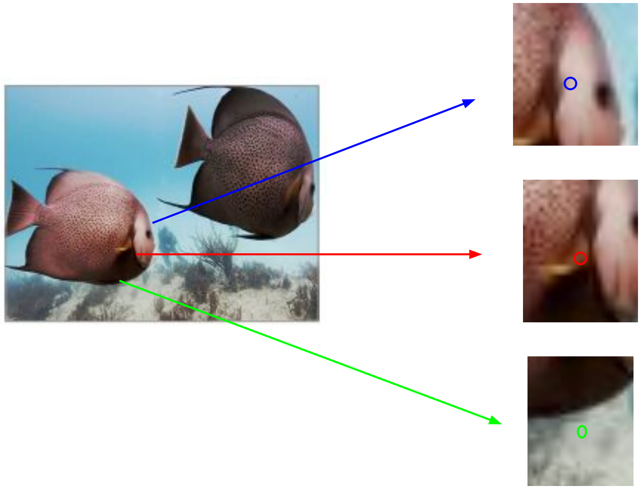
(Credit: https://naokishibuya.medium.com/)

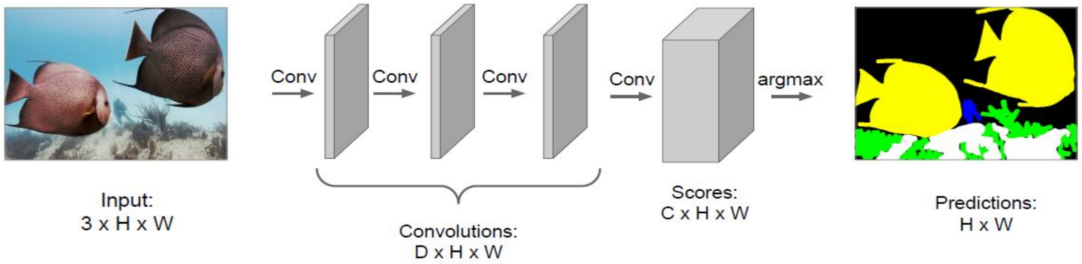


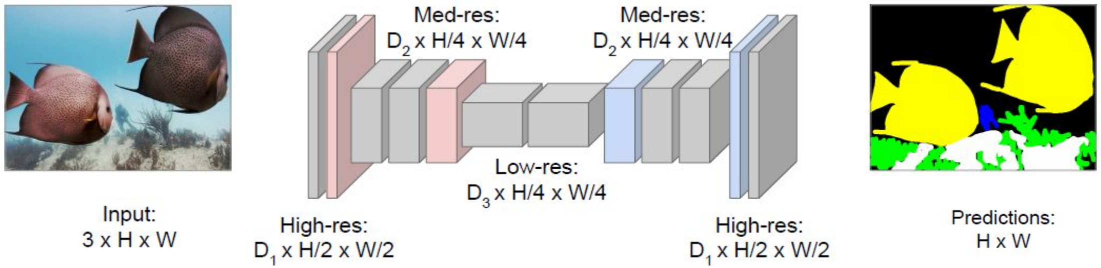
often used for segmentation, generation, or other regression—outputs are structured objects such as images, videos, time series, speech, etc

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

Semantic Segmentation: Summary







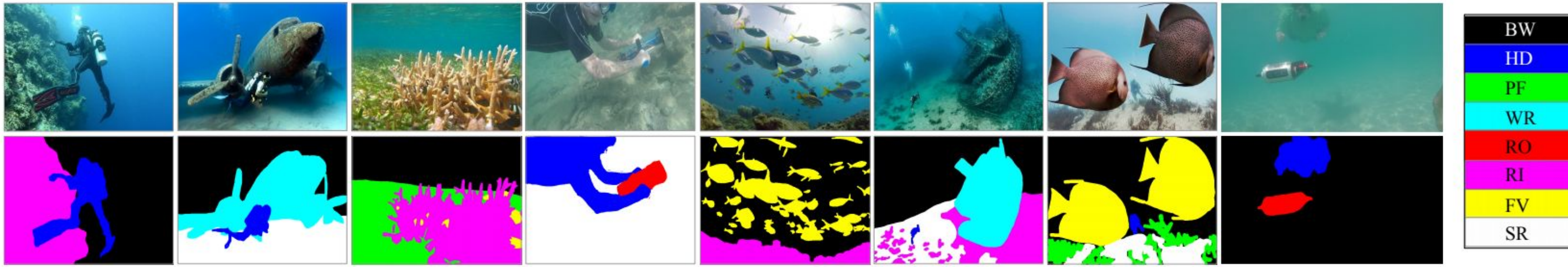
(Islam et al., "Semantic Segmentation of Underwater Imagery: Dataset and Benchmark," 2020)
Paper from UMN IRVLab http://irvlab.dl.umn.edu/
(Li et al., Detection and Segmentation 2020)

Semantic Segmentation: Summary

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

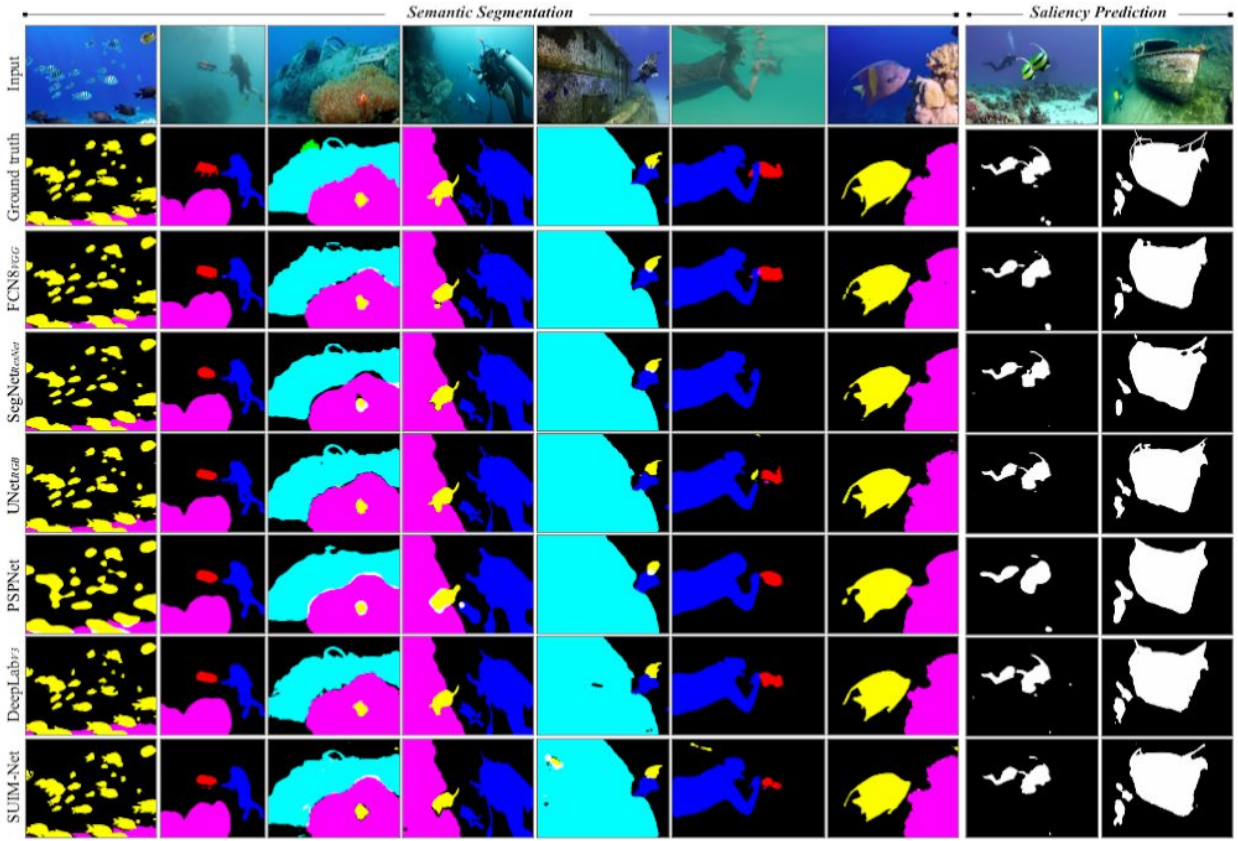
Don't differentiate between different instances of the same class of object; only care about the pixel-level.

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



(Islam et al., "Semantic Segmentation of Underwater Imagery: Dataset and Benchmark," 2020)
Paper from UMN IRVLab http://irvlab.dl.umn.edu/

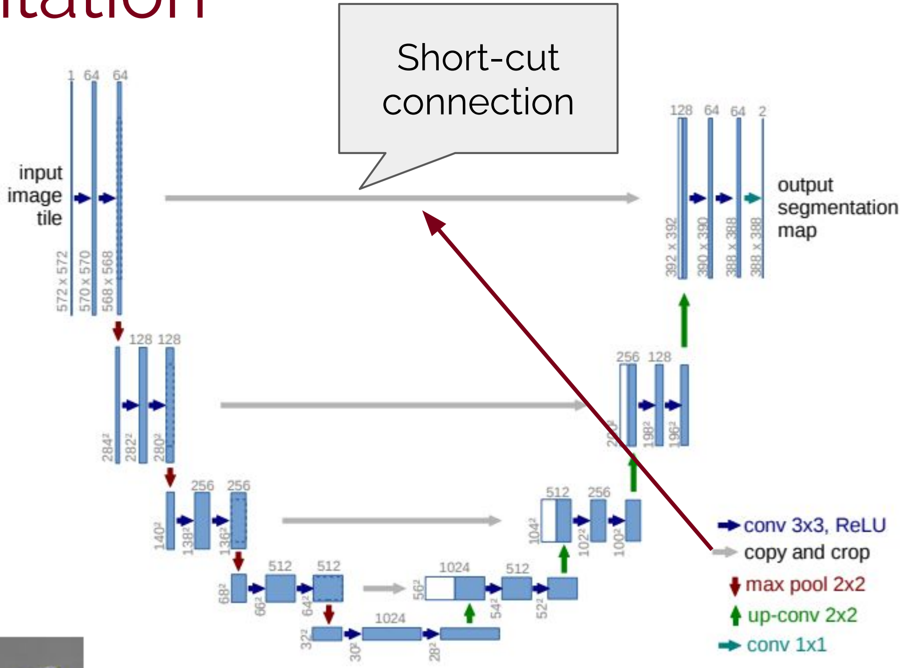
State of the Art Segmentation CNNs

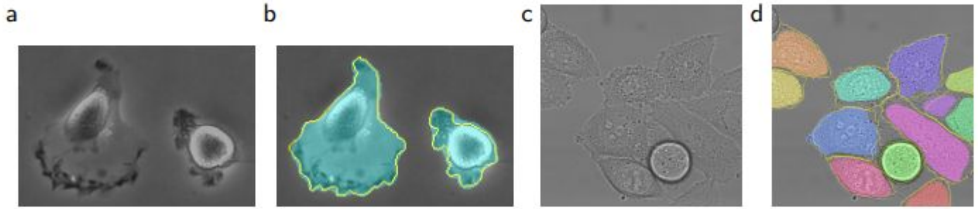


(Islam et al., "Semantic Segmentation of Underwater Imagery: Dataset and Benchmark," 2020)
Paper from UMN IRVLab http://irvlab.dl.umn.edu/

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





(Ronneberger et al., 2015)

Mask R-CNN for instance segmentation

Instance segmentation = detection + segmentation

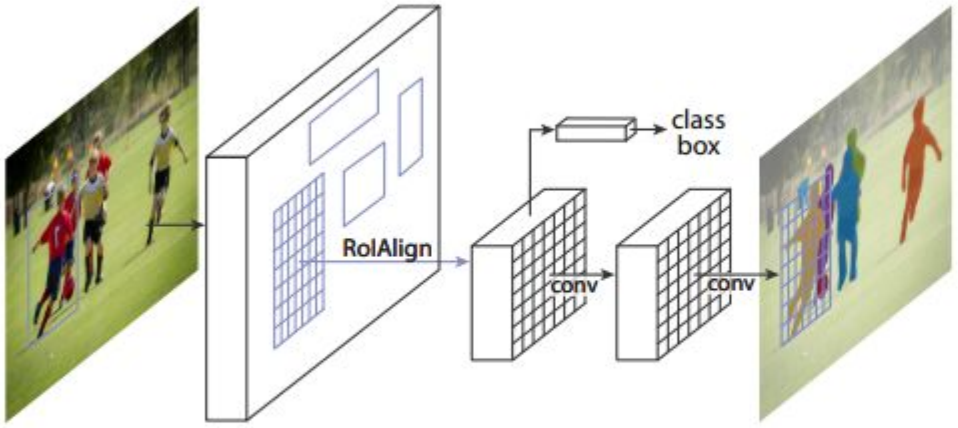


Figure 1. The **Mask R-CNN** framework for instance segmentation.

- Extension of Faster R-CNN
- Adds a masking network after the output of Faster R-CNN
- Masking network outputs a segmentation mask for each object instance

(He et al., 2018)

Mask R-CNN for instance segmentation

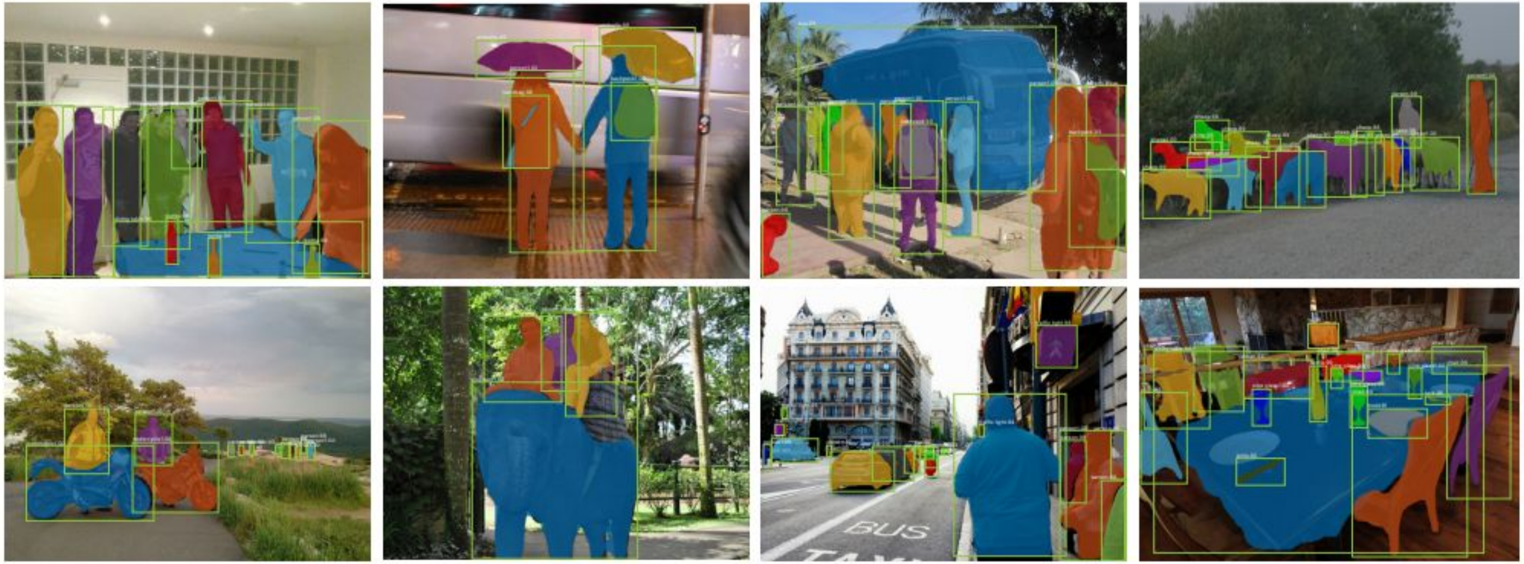


Figure 2. **Mask R-CNN** results on the COCO test set. These results are based on ResNet-101 [19], achieving a *mask* AP of 35.7 and running at 5 fps. Masks are shown in color, and bounding box, category, and confidences are also shown.

(He et al., 2018)

DeepLab by Google

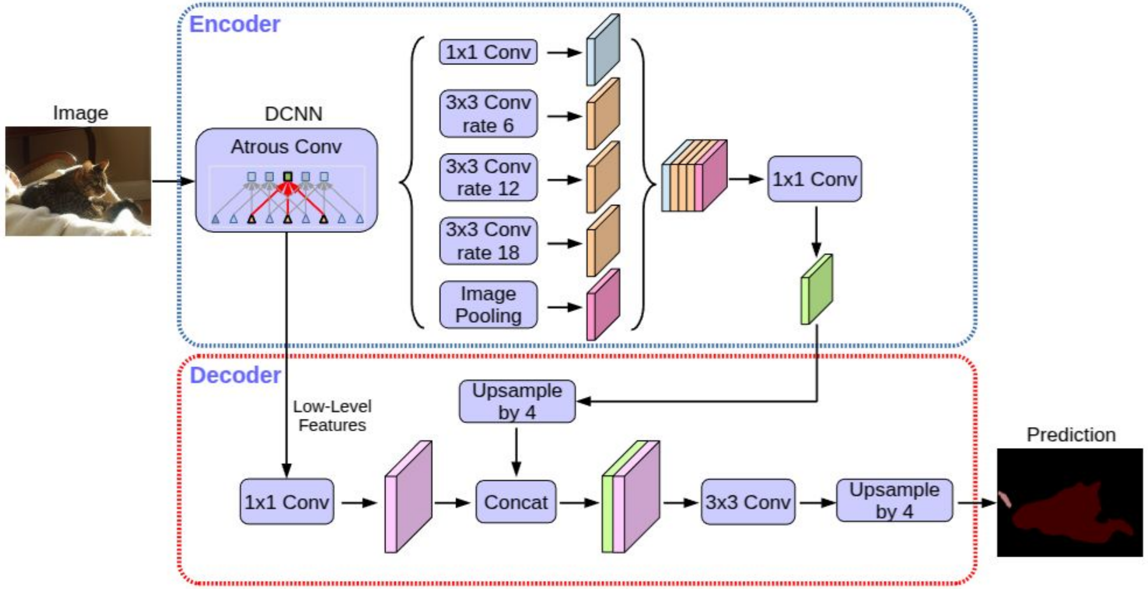


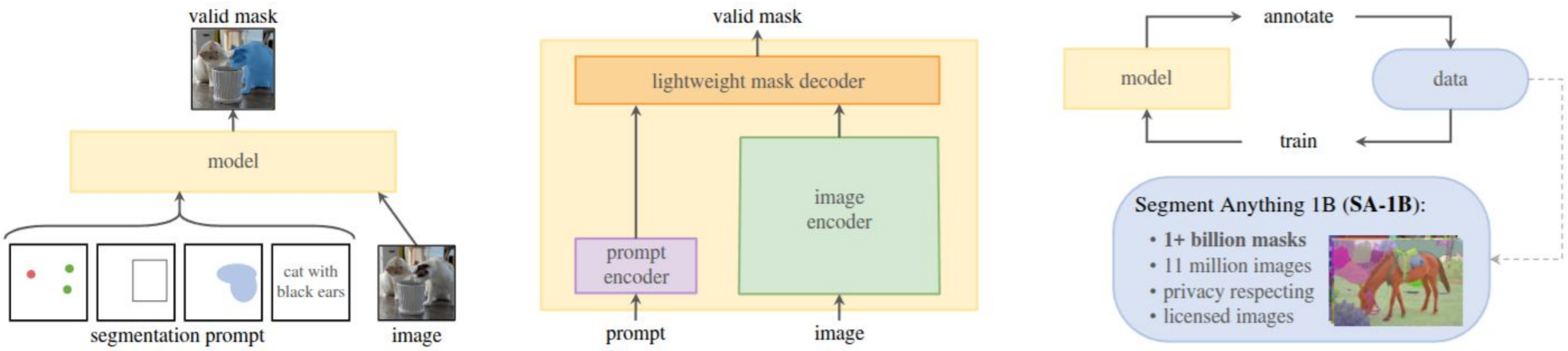
Fig. 2. Our proposed DeepLabv3+ extends DeepLabv3 by employing a encoder-decoder structure. The encoder module encodes multi-scale contextual information by applying atrous convolution at multiple scales, while the simple yet effective decoder module refines the segmentation results along object boundaries.

https://arxiv.org/abs/1802.02611

Segment anything (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://segment-anything.com/

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.