AdTrans project, 2023-1-PL01-KA220-HED-000158917

ADTRANS
ADVANCED DATA ANALYSIS FOR TRANSPORTATION AND MOBILITY ACADEMIC TEACHERS

Session 5

Image Segmentation & Object Detection

Paraskevi Fasouli

Co-funded by the European Union

Part 1:

Image Segmentation

Overview

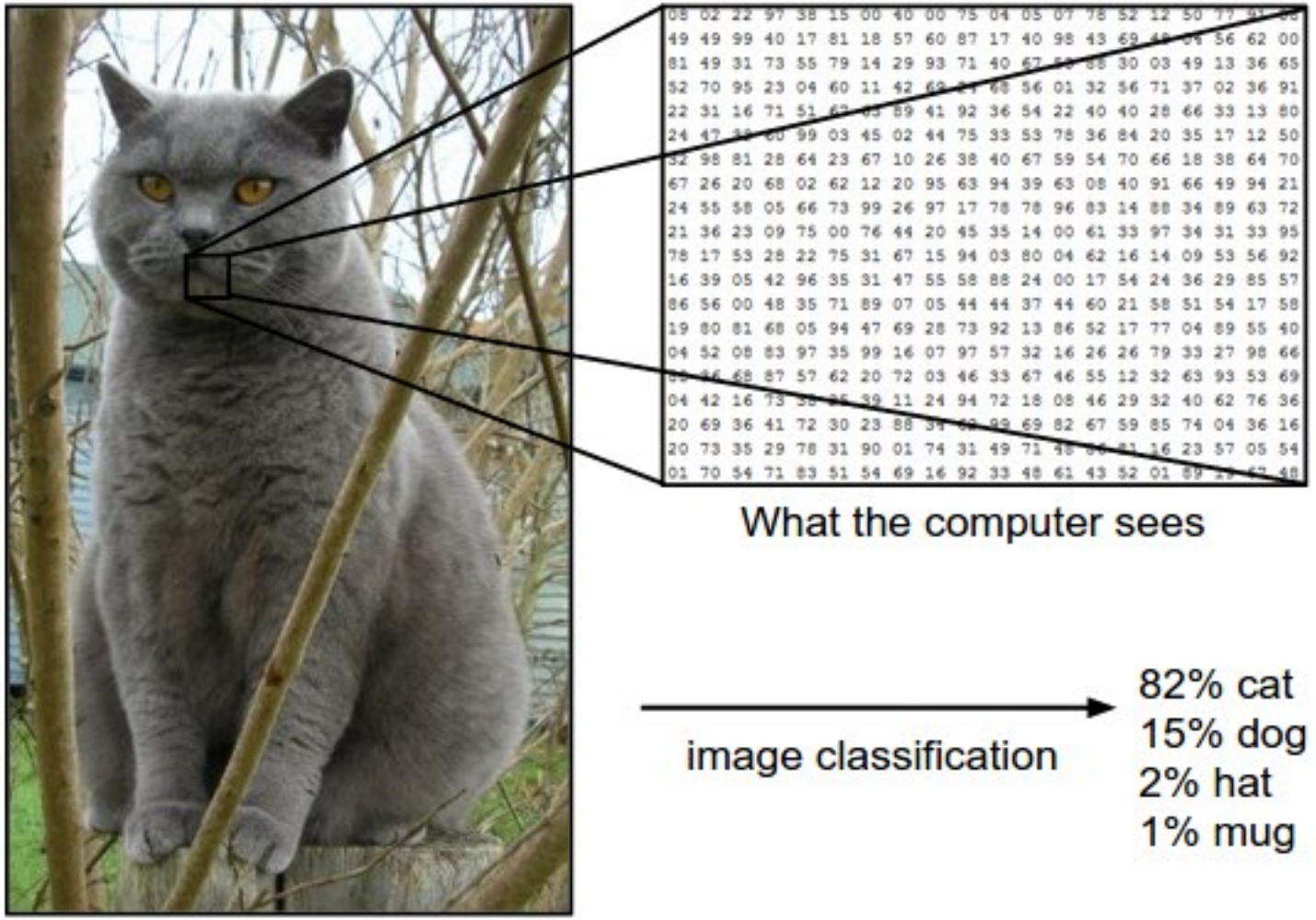
Image Segmentation Concept

Segmentation Models

Object Recognition Concept

Object Recognition Models

Image Classification



Task: Predict a single label (or a distribution over labels) for a given image.

Images are 3D arrays of integers from 0 to 255, of size Width x Height x 3. The 3 represents the three-color channels Red, Green, Blue.

CNNs are used for this task.

Image Classification: Challenges

Viewpoint variation

- A single instance of an object can be oriented in many ways with respect to the camera.

Scale variation

- Visual classes often exhibit variation in their size (size in the real world, not only in terms of their extent in the image).

Deformation

- Many objects of interest are not rigid bodies and can be deformed in extreme ways.

Occlusion

- The objects of interest can be occluded. Sometimes only a small portion of an object (as little as few pixels) could be visible.

Illumination conditions

- The effects of illumination are drastic on the pixel level.

Background clutter

- The objects of interest may blend into their environment, making them hard to identify.

Intra-class variation

- The classes of interest can often be relatively broad, such as chair. There are many different types of these objects, each with their own appearance.

Image Classification: Challenges

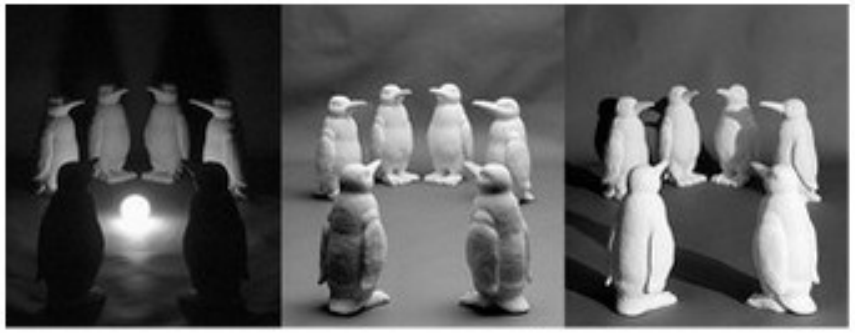
Viewpoint variation

Scale variation

Deformation

Occlusion

Illumination conditions

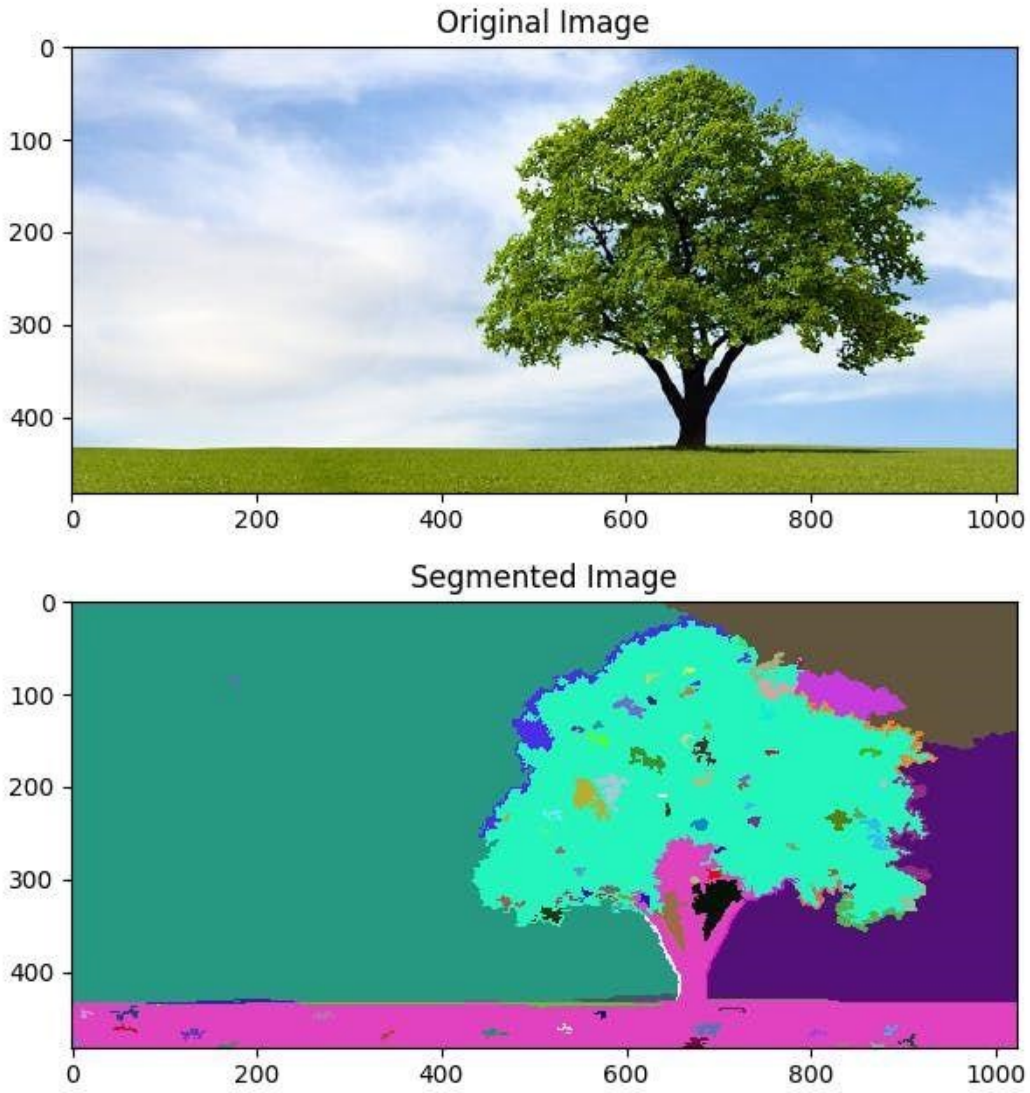


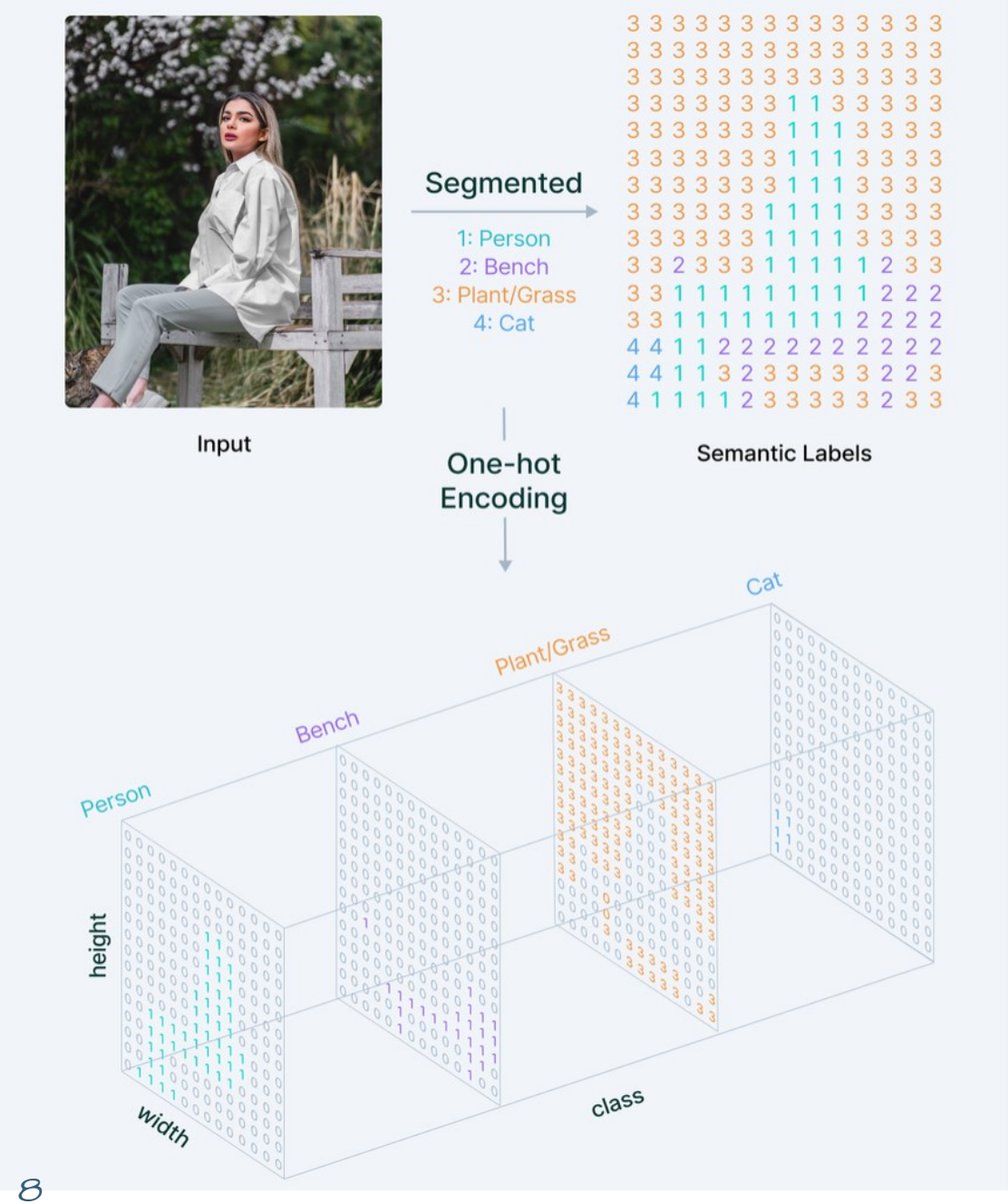
Background clutter

Intra-class variation

What is Image Segmentation?

- ✔ Computer vision task that involves **partitioning an image into multiple segments or regions**.
- ✔ Each pixel is assigned a **semantic label** that corresponds to the category of objects it contains.
- ✔ Key Part of Scene Understanding
- ✔ Also called Semantic Segmentation





What is Image Segmentation?

- ✔ A classification problem of pixels with semantic labels
- ✔ Performs pixel-level labeling with a set of object categories (e.g., human, car, tree, sky), so it is generally harder than image classification

Key Characteristics

Pixel-level Labeling:

- A semantic label is assigned to each pixel in an image, indicating the category of the object or region to which it belongs. This enables fine-grained analysis of the image content, allowing for precise delineation of object boundaries and shapes.

Semantic Understanding:

- By segmenting an image into semantically meaningful regions, semantic segmentation provides a deeper understanding of the scene's content and context. It allows computer vision systems to recognize objects, understand their spatial relationships, and infer their significance within the scene.

Multi-class Classification:

- It involves multiple object classes or categories, each represented by a distinct label. Common classes include "person," "car," "road," "building," "tree," and so on. The goal is to accurately classify every pixel in the image into one of these predefined categories.

Applications

- Autonomous driving
- Scene understanding
- Image editing
- Augmented reality
- Medical imaging
- Satellite imagery analysis

In autonomous driving, semantic segmentation is used to identify and classify objects in the vehicle's surroundings, such as *pedestrians, vehicles, road markings, and traffic signs*, to facilitate safe navigation and decision-making.

Classification vs Segmentation

Image Classification (ImageNet)

- Image level prediction
- Location Invariant
- Low Resolution (224x224 input)
- Computation requirements: ~10 GFLOPs
- Networks are designed for tasks and trained from scratch

Image Segmentation (Cityscapes)

- Pixel level prediction
- Location Variant
- High Resolution (1024x2048 input)
- Computation requirements: ~1 TFLOP
- SS Networks are adapted from classification networks and then retrained

Classification vs Segmentation

Is this a cat?

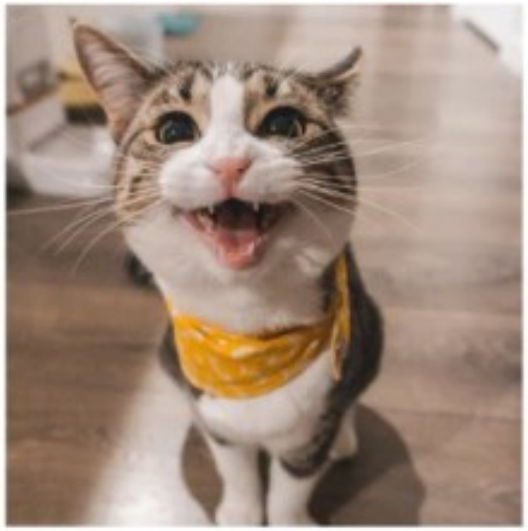


Image Classification

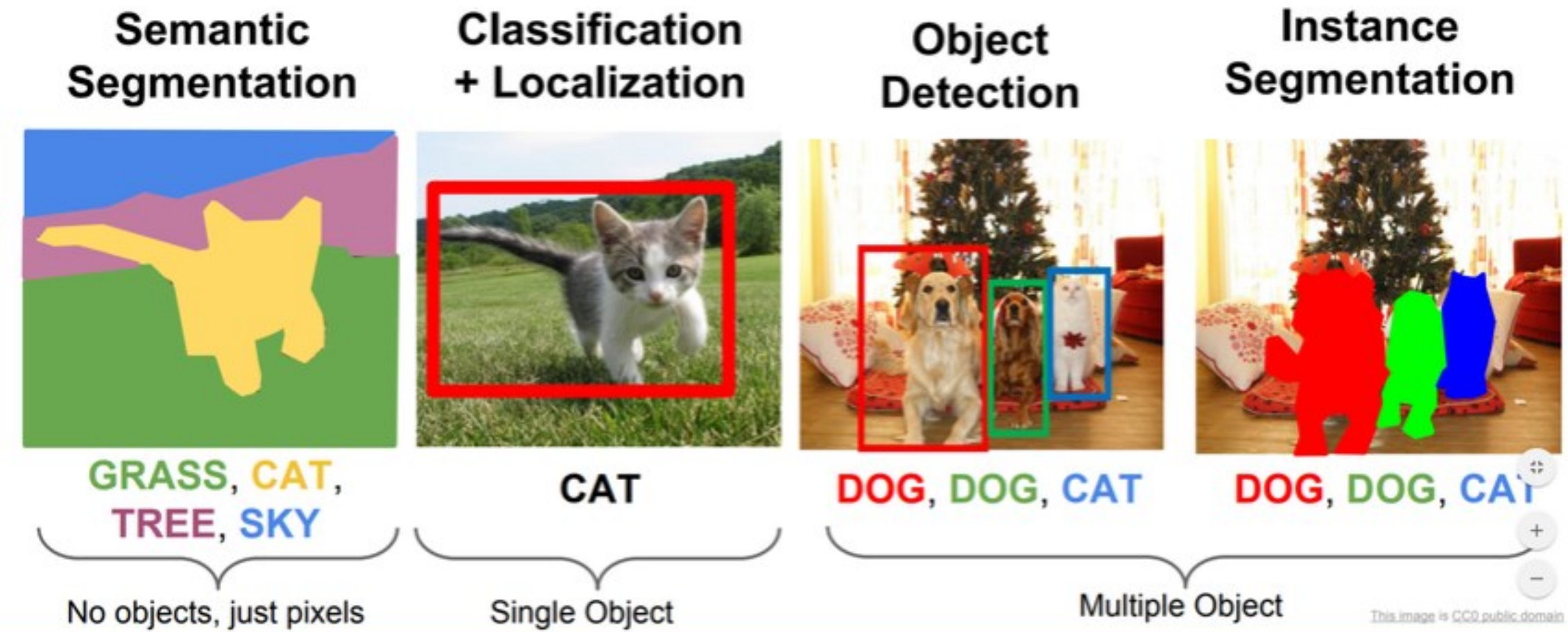
What is there in the image and where?

Object Detection

Which pixels belong to which object

Image Segmentation

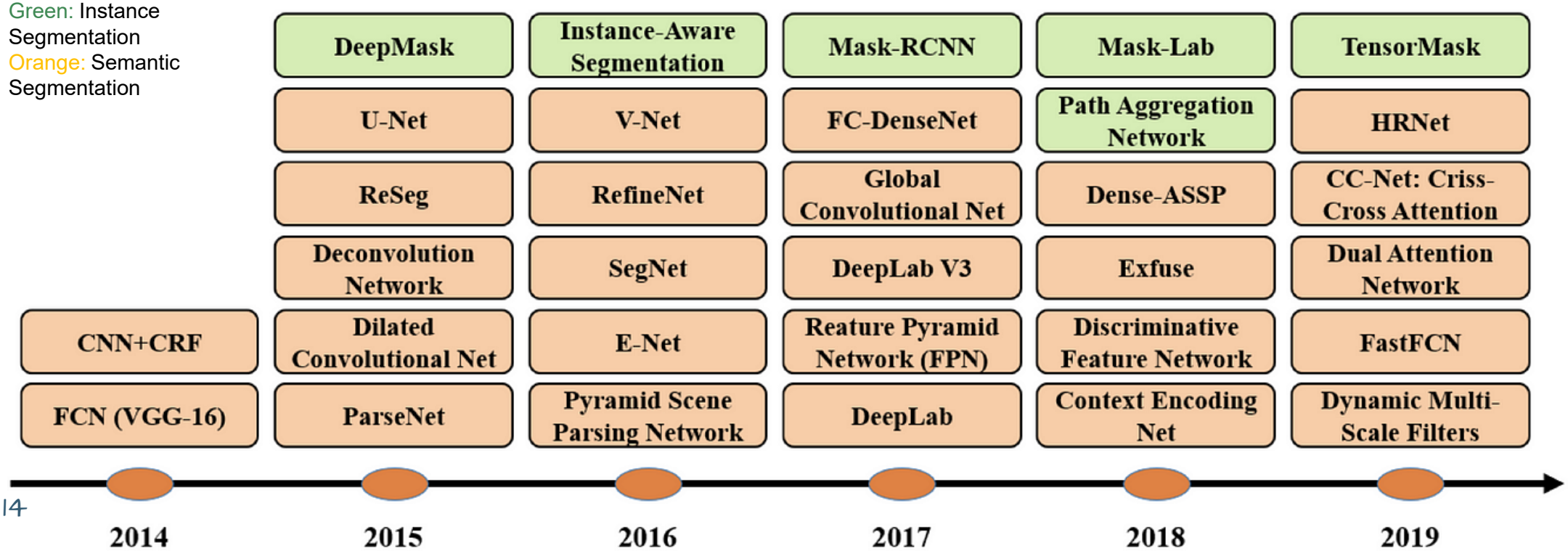
Different Tasks Overview



DL Image Segmentation Models

- ✔ Mostly CNN models
- ✔ Reason: They learn to capture both low-level image features (such as edges & textures) and high-level semantic information (such as object shapes & categories) through convolutional operations and feature aggregation.

Green: Instance Segmentation
Orange: Semantic Segmentation



Metrics for Segmentation Models

1. Pixel Accuracy (PA):

$$\mathrm{PA} = \frac{\sum_{i=0}^{K} p_{ii}}{\sum_{i=0}^{K} \sum_{j=0}^{K} p_{ij}}$$

The ratio of pixels properly classified, divided by the total number of pixels.

2. Mean Pixel Accuracy (MPA):

$$\mathrm{MPA} = \frac{1}{K+1} \sum_{i=0}^{K} \frac{p_{ii}}{\sum_{j=0}^{K} p_{ij}}$$

The ratio of correct pixels is computed in a per-class manner and then averaged over the total number of classes.

Metrics for Segmentation Models

3. Intersection over Union (IoU):

$$\text{IoU} = J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

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

- Also called Jaccard Index
- The most used metrics in semantic segmentation. (mean-IoU/mIoU)
- A denotes the ground truth and B denotes the predicted segmentation maps.

4. Dice Coefficient:

$$\text{Dice} = \frac{2|A \cap B|}{|A| + |B|} \qquad \text{Dice} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} = \text{F1}$$

- Essentially identical to the F1 score.
- The Dice coefficient and IoU are positively correlated.

Metrics for Segmentation & Classification Models

5. Precision/ Recall/ F1 score:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1-score} = \frac{2\ \text{Prec}\ \text{Rec}}{\text{Prec} + \text{Rec}}$$

Predicted Value (predicted by the test)		Actual Value (as confirmed by experiment)	
		positives	negatives
	positives	**TP** True Positive	**FP** False Positive
	negatives	**FN** False Negative	**TN** True Negative

Datasets for Autonomous Cars

Cityscapes:

- Cityscapes is a large-scale dataset containing high-resolution images captured in urban environments from 50 cities of Europe. It includes pixel-level annotations of 30 classes grouped into 8 categories for semantic segmentation tasks, covering classes such as road, sidewalk, building, vehicle, pedestrian, and more.

KITTI Vision Benchmark Suite:

- The KITTI Vision Benchmark Suite is a dataset collected for autonomous driving research, consisting of images captured from a vehicle-mounted camera in urban and highway scenarios. It includes pixel-level annotations for tasks such as object detection, tracking, stereo, and optical flow, which can be used for semantic segmentation and scene understanding.

PandaSet:

- PandaSet was the first open-source AV dataset available for both academic and commercial use. It contains 48,000 camera images, 16,000 LiDAR sweeps, 28 annotation classes, and 37 semantic segmentation labels taken from a full sensor suite.

Datasets for Autonomous Cars

BDD100K:

- The Berkeley DeepDrive 100K (BDD100K) dataset is a large-scale driving dataset containing images captured from vehicle-mounted cameras in various driving conditions. It includes pixel-level annotations for semantic segmentation tasks, covering classes such as road, lane, vehicle, pedestrian, bicycle, and more.

A2D2 Dataset:

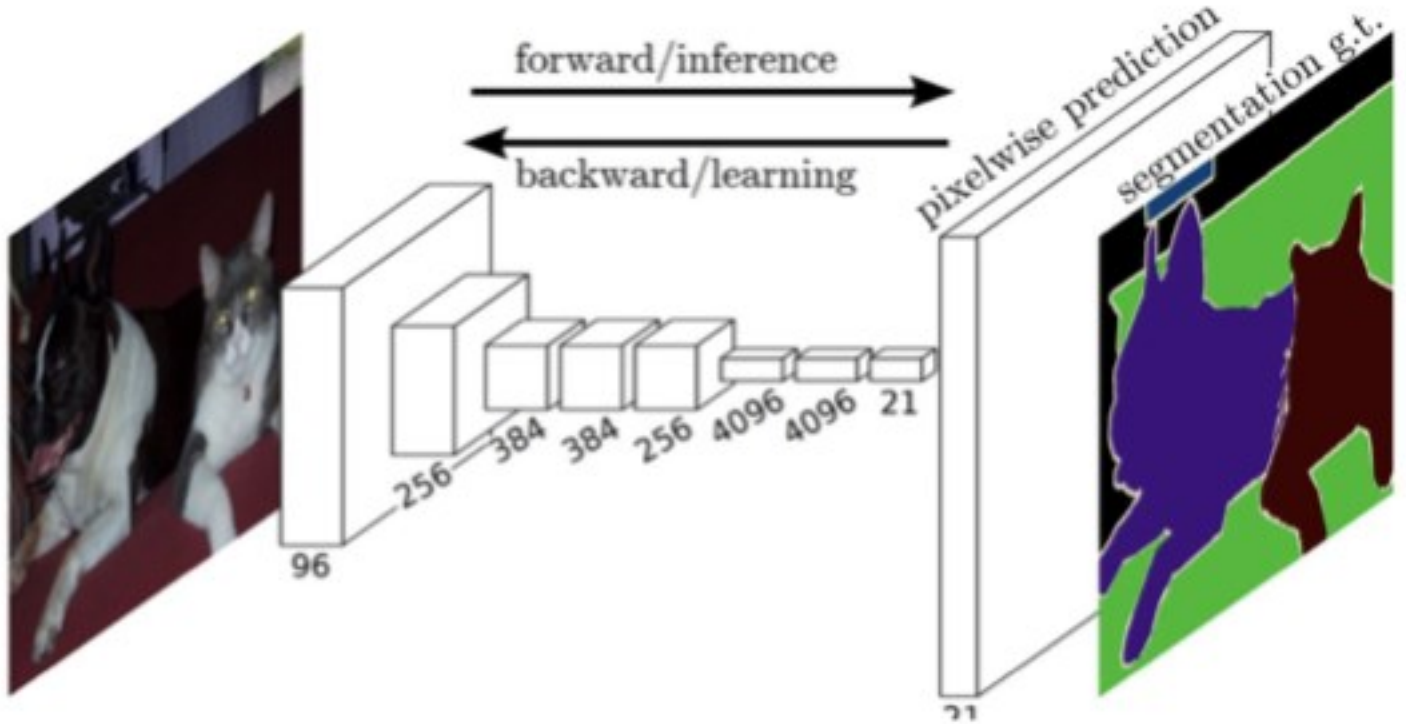
- The Audi Autonomous Driving Dataset (A2D2) features over 41,000 labeled with 38 features. Around 2.3 TB in total, A2D2 is split by annotation type (i.e. semantic segmentation, 3D bounding-box).

Waymo Open Dataset:

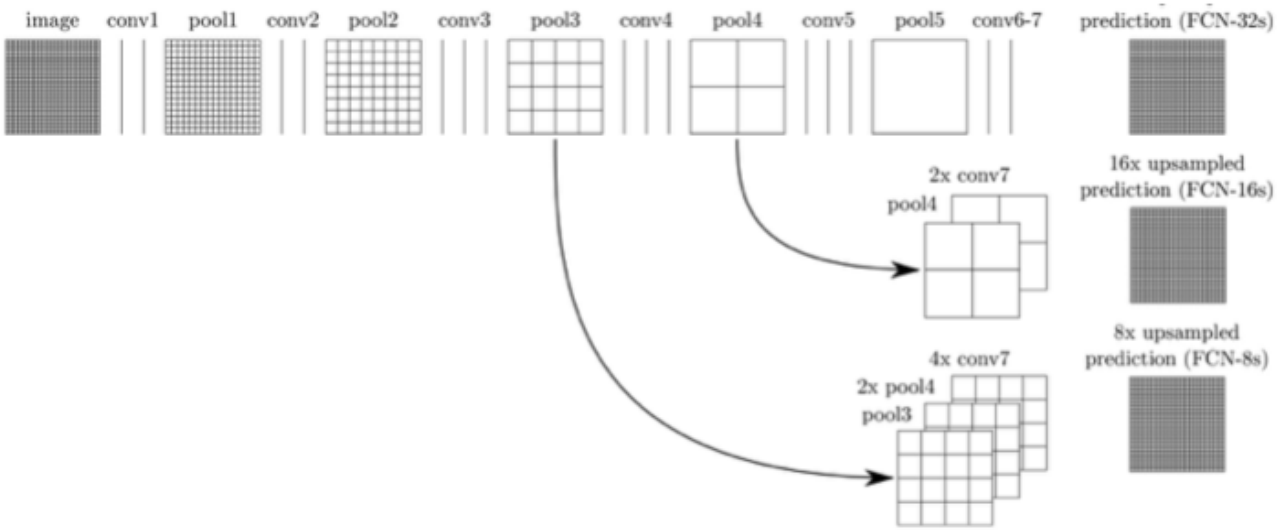
- The Waymo Open dataset is an open-source multimodal sensor dataset for autonomous driving. Extracted from Waymo self-driving vehicles, the data covers a wide variety of driving scenarios and environments. It contains 1000 types of different segments where each segment captures 20 seconds of continuous driving, corresponding to 200,000 frames at 10 Hz per sensor.

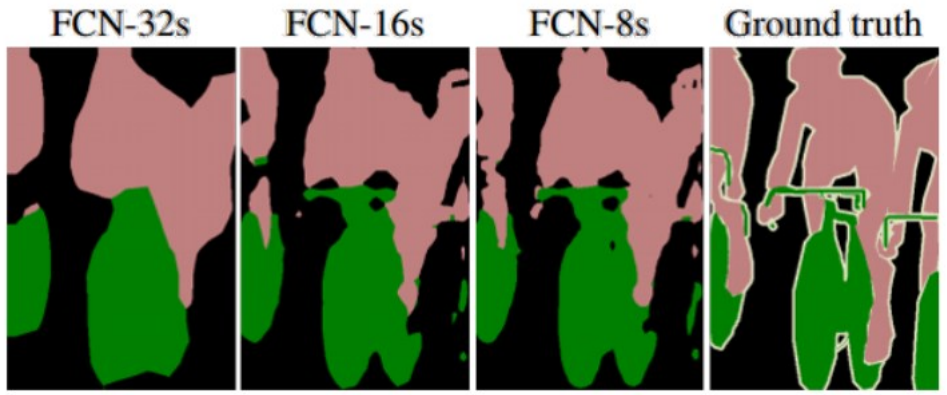
Fully Convolutional Networks (FCN)

- Take an image of arbitrary size & produce a same-size segmentation map.
- Replacing all fully-connected layers with the fully- convolutional layers.
- Outputs a **spatial segmentation map**, not classification scores.



Fully Convolutional Networks (FCN)





Skip connections allow feature maps from final layers to be up-sampled and fused with features maps of earlier layers. This helps the model to produce a very accurate and detailed segmentation.

Combines the semantic **high-information** from the **deep and coarse layers** with the appearance **low-information** from the **shallow and fine layers**.

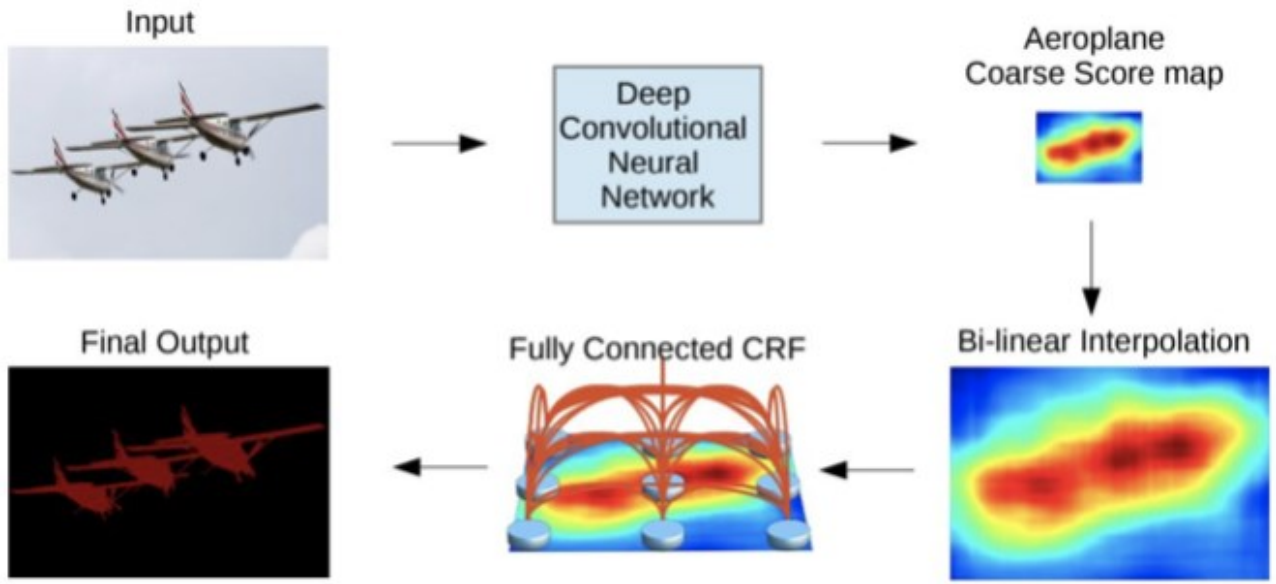
Convolutional Models with Graphical Models

FCN Limitations:

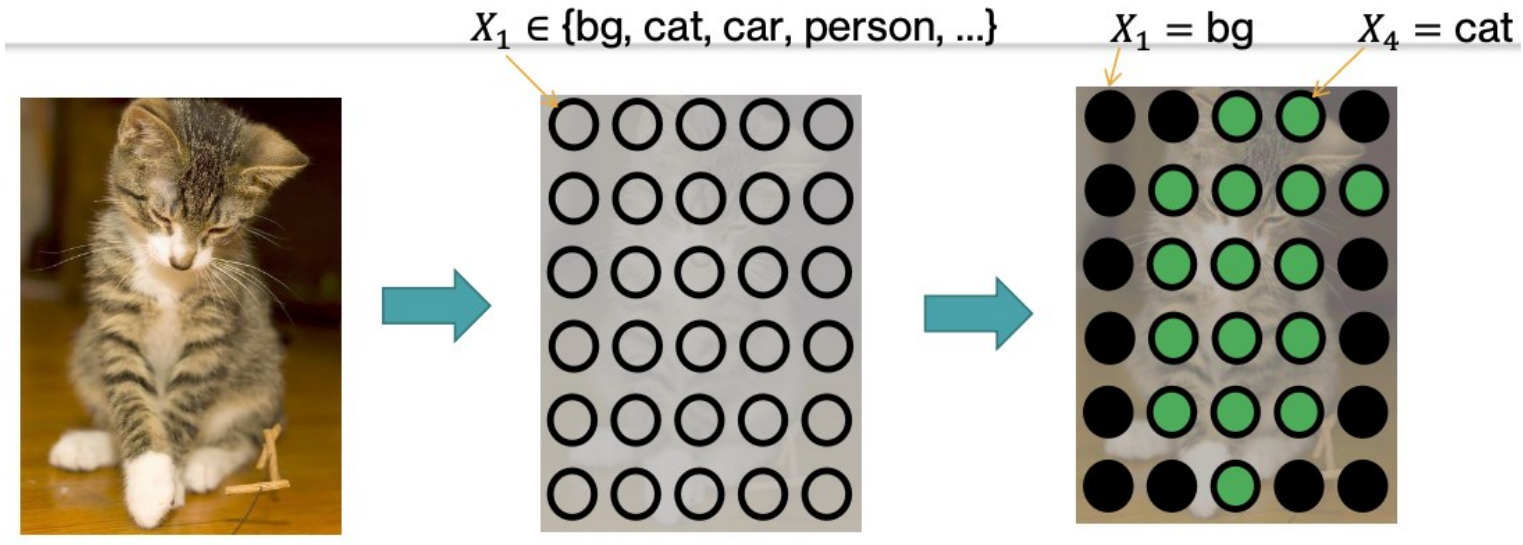
- Not fast enough for real-time inference
- Does not consider the global context information efficiently
- Not easily transferable to 3D images

Improvement: FCN+CRF

- Final CNN layer can be combined to a fully connected Conditional Random Field (CRF)
- Higher accuracy rate



Conditional Random Fields (CRF)

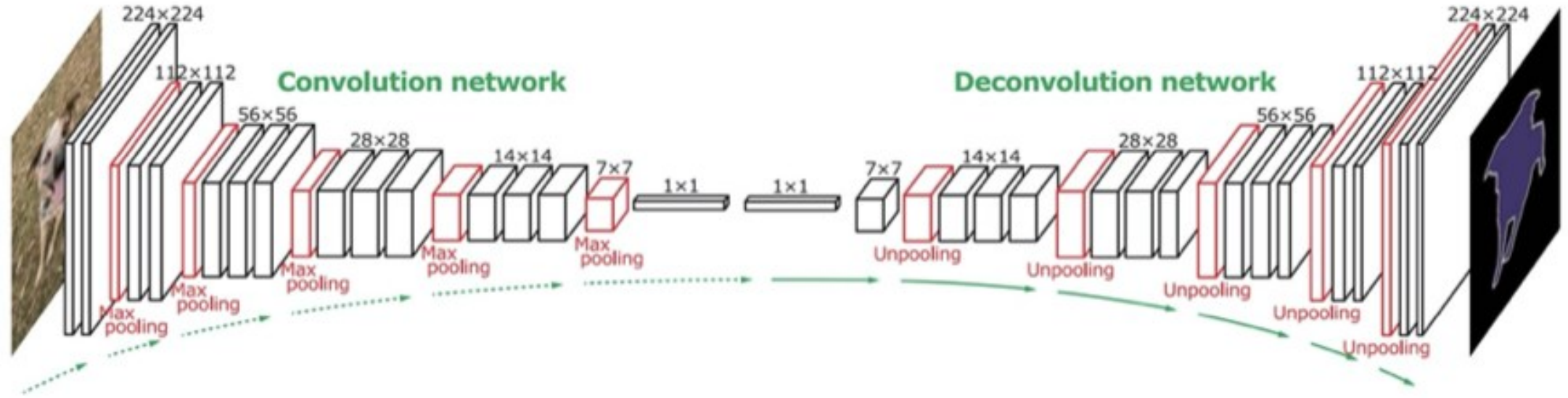


Define a discrete random variable, Xi, for each pixel i

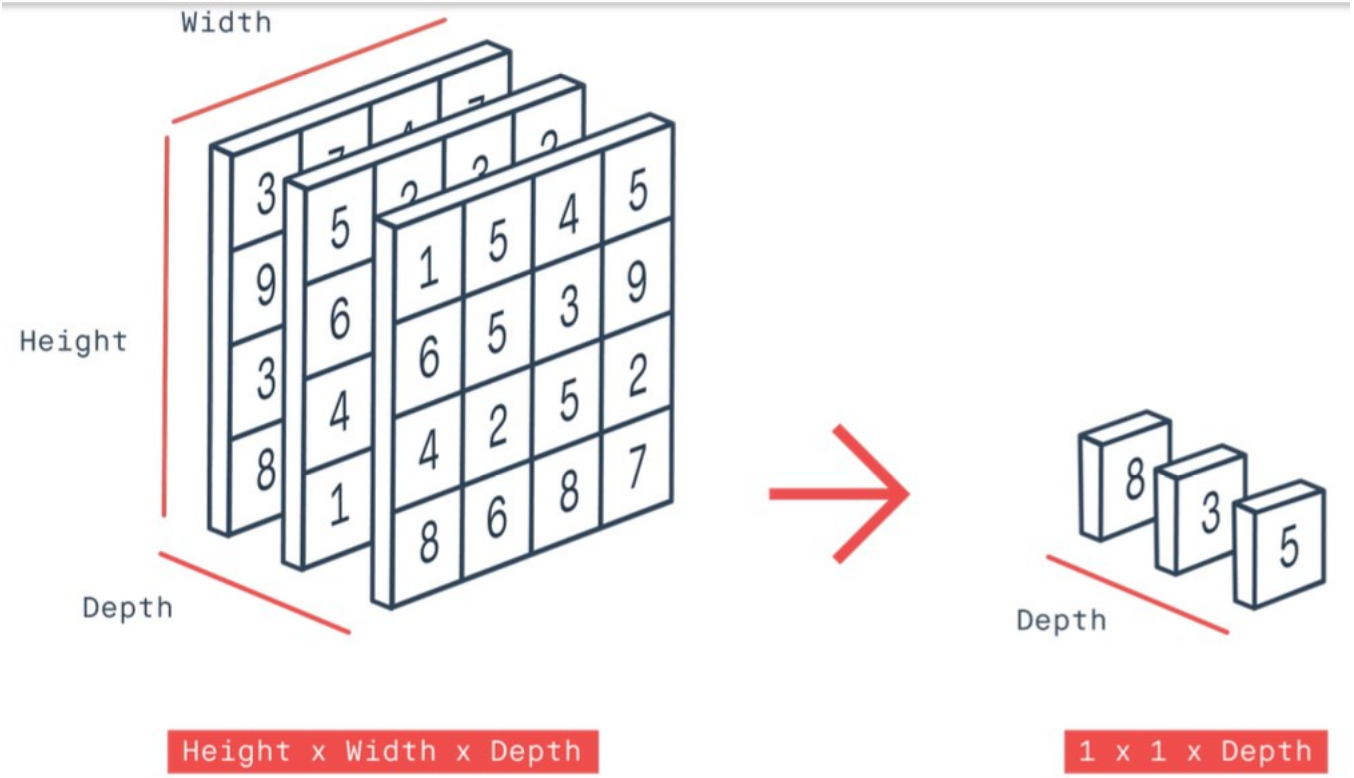
- Each Xi takes a value from the label set $\mathcal{L}$
- The random variables are connected to form a random field. The most probable assignment, conditioned on the image, is our semantic segmentation result.

Encoder-Decoder Models
Deconvolutional Models

- Two parts: an encoder and a decoder
- An encoder uses convolutional layers to generate a feature map
- A decoder uses a deconvolutional network which generates a map of pixel-wise class probabilities based on the input feature vector

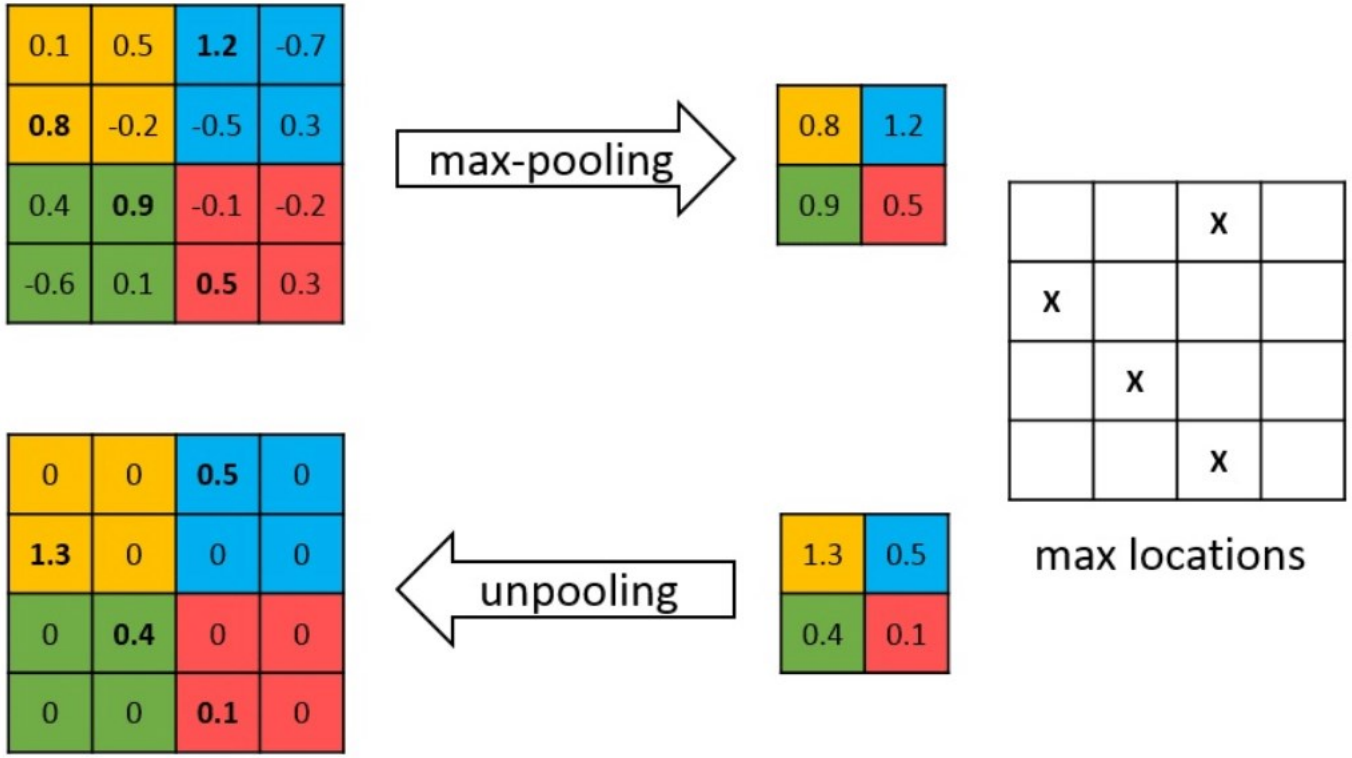


Encoder-Decoder Models
Deconvolutional Models

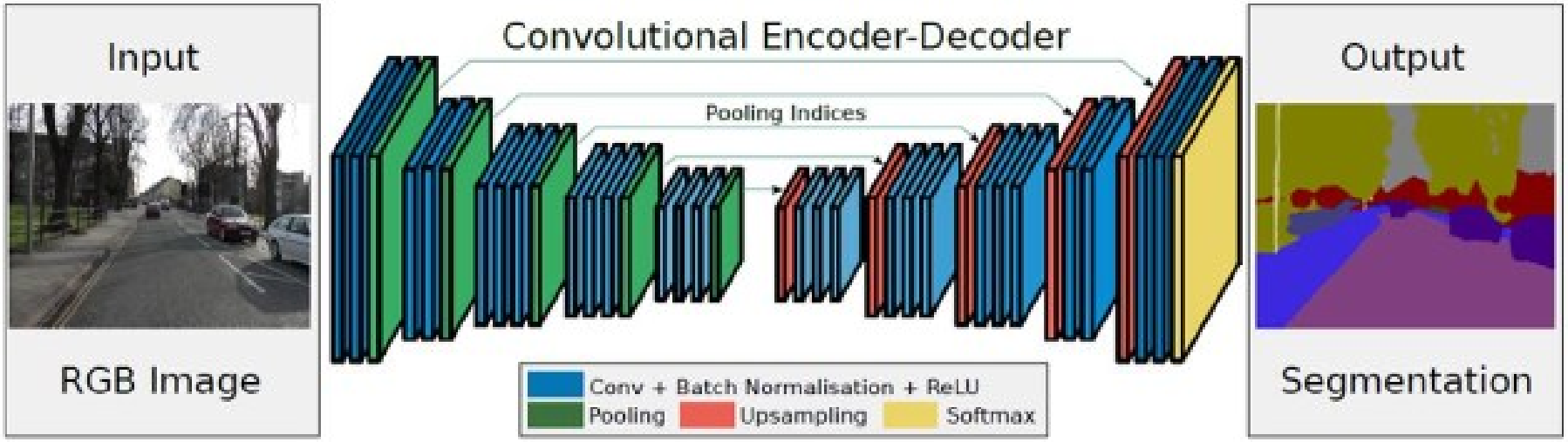


Encoder-Decoder Models
Deconvolutional Models

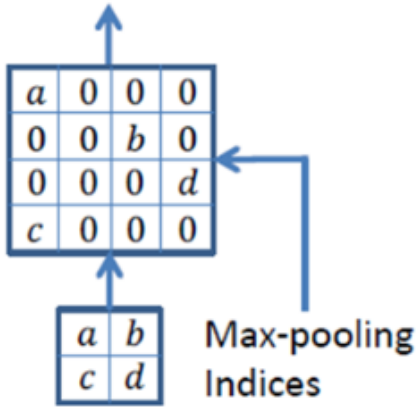
Unpooling Layer



Encoder-Decoder Models
SegNet



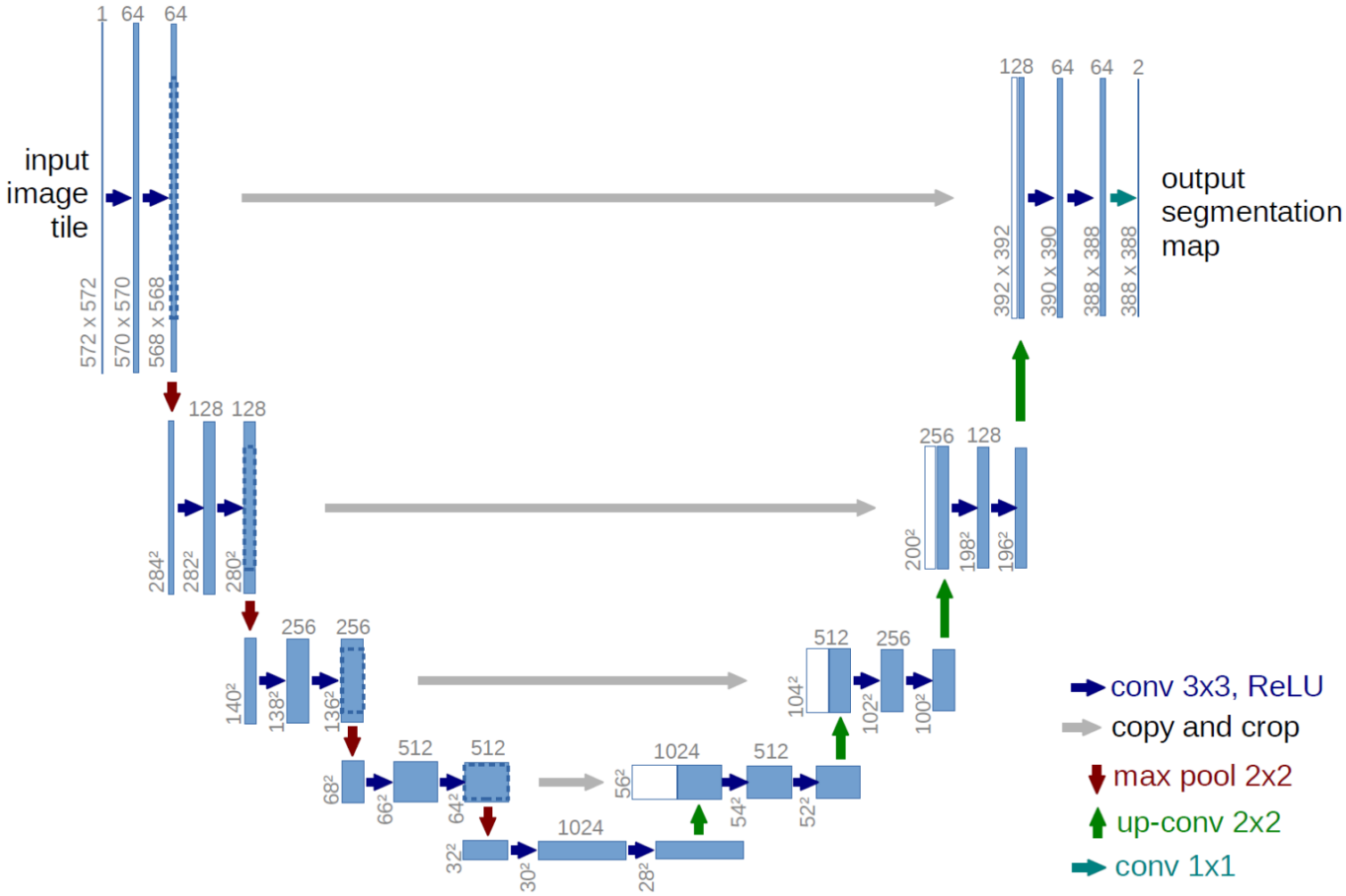
- Uses a series of convolutional layers for feature extraction and up sampling layers for pixel-wise classification.
- Eliminates the need for learning to up sample.



Encoder-Decoder Models

U-Net

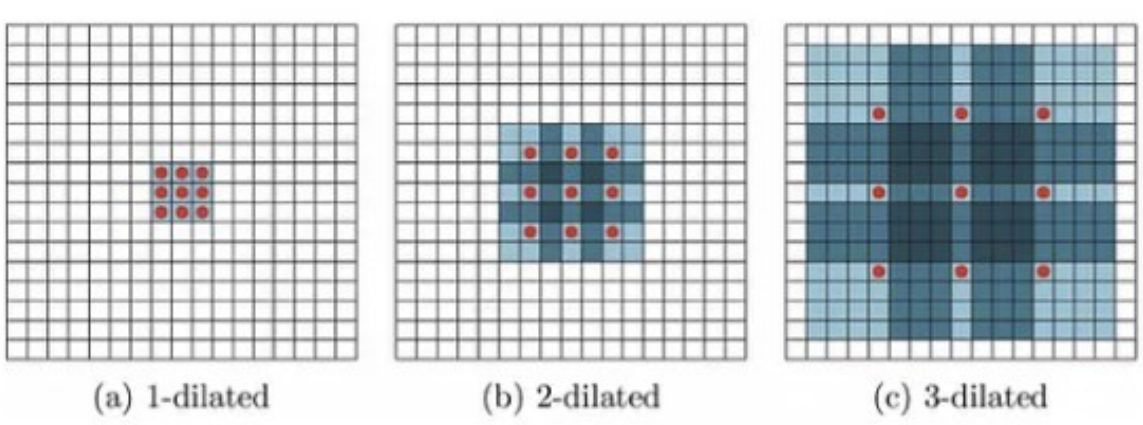
- ✔ Consists of a contracting path (encoder) for feature extraction and a symmetric expanding path (decoder) for pixel-wise classification.
- ✔ Feature maps from the down-sampling part are copied to the up-sampling part to avoid losing pattern information.
- ✔ Used for tasks such as road segmentation, obstacle detection, and lane marking extraction.

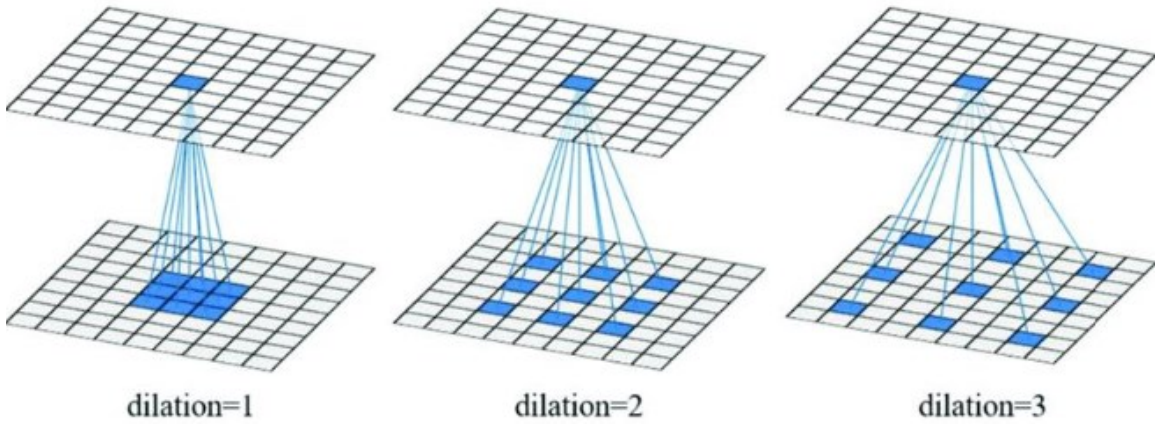


Encoder-Decoder Models
DeepLab Family Models

Dilated (Atrous) Convolution

- An additional parameter is added to convolutional layers known as dilation rate which defines a spacing between the weights of the kernel.
- DeepLab v1, DeepLab v2 and DeepLab v3 are the state-of-the-art models for image segmentation approaches.





Part 2:

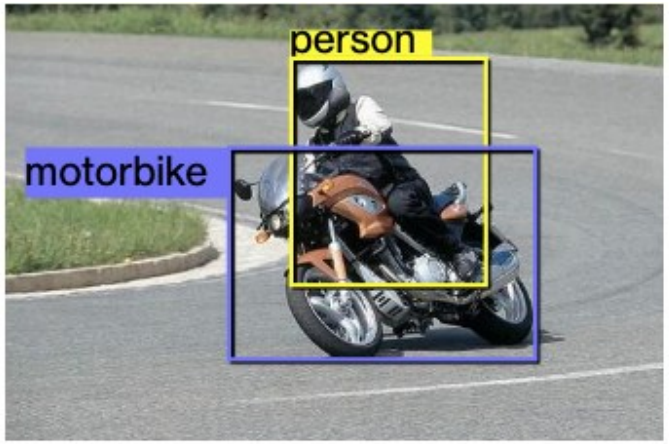
Object Detection

Why is detection hard?

- Precise localization
- Much larger impact of pose
- Occlusion makes localization difficult
- Counting
- Small objects

{ airplane, bird, motorbike, person, sofa }

Input



Desired output

Detection as Classification

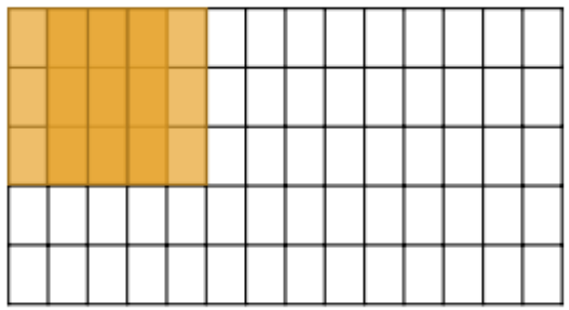
Run through every possible box and classify

How many boxes?

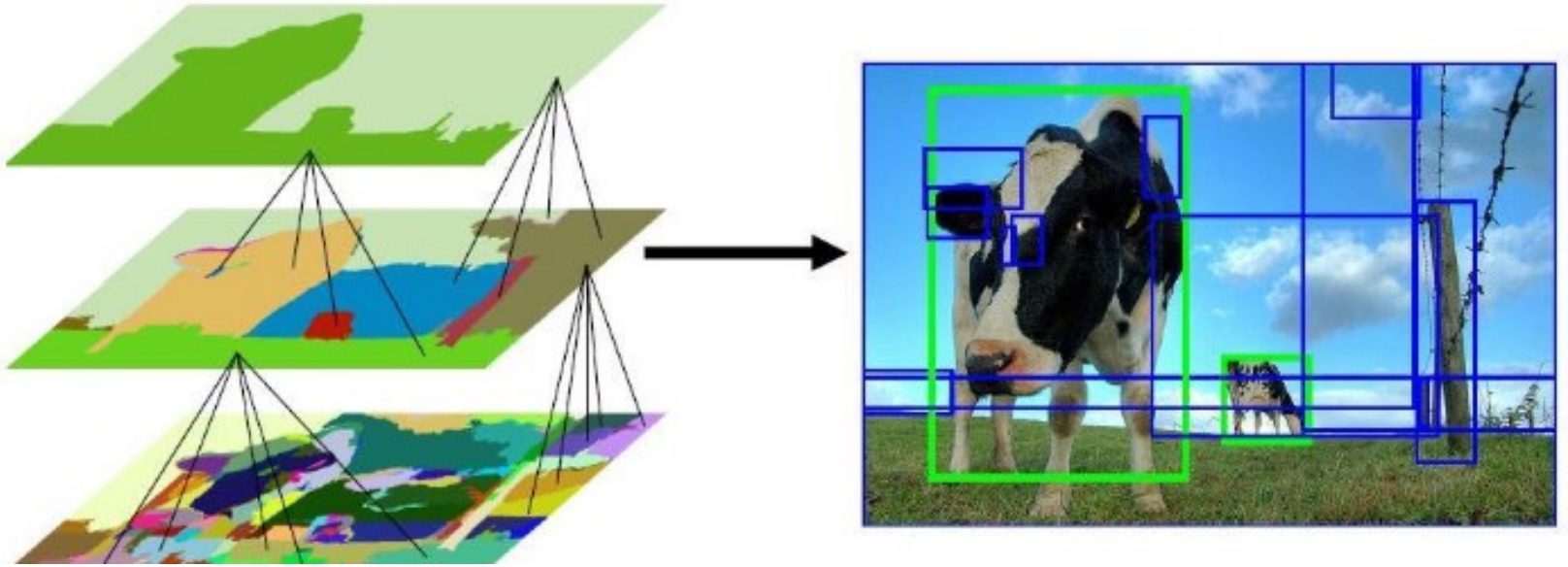
- Every pair of pixels = 1 box
- $\binom{N}{2} = O(N^2)$
- For 300 x 500 image, N = 150K
- 2.25×10^{10} boxes!
- Classifies millions of boxes, so must be very fast
- Needs ultra-fine sampling of scales and object sizes, can still miss outlier sizes

Scanning Window

- Fixed size
  - Can take a few different sizes
- Fixed stride
  - Convolution with a filter
    - Classic: compute HOG features over entire image



Object Proposals

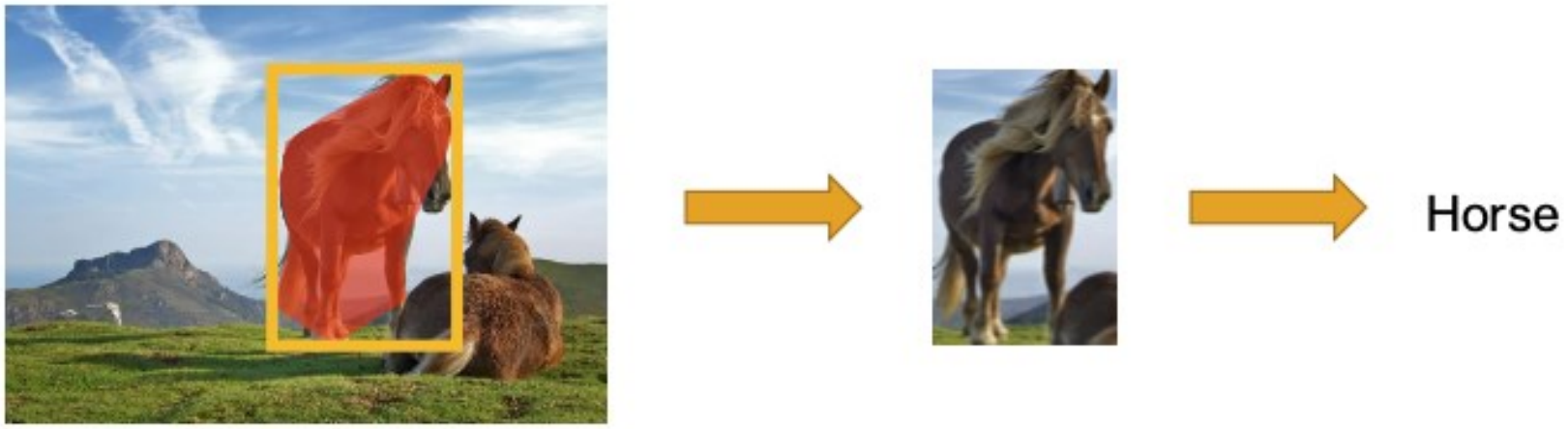


- Use segmentation to produce ~5K candidates
- Many different segmentation algorithms (k-means on color, k-means on color & position, N-cuts….)
- Every cluster is a candidate object
- Thousands of segmentations -> thousands of candidate objects

- Many hyperparameters (number of clusters, weights on edges)

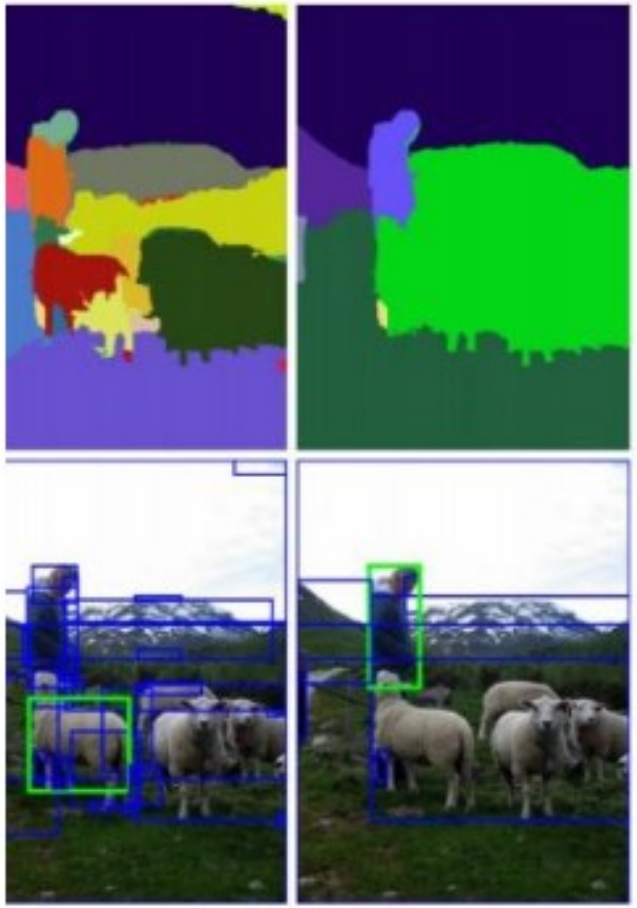
What are the proposals?

- Each proposal is a group of pixels
- Take tight fitting box and classify it
- Can leverage any image classification approach



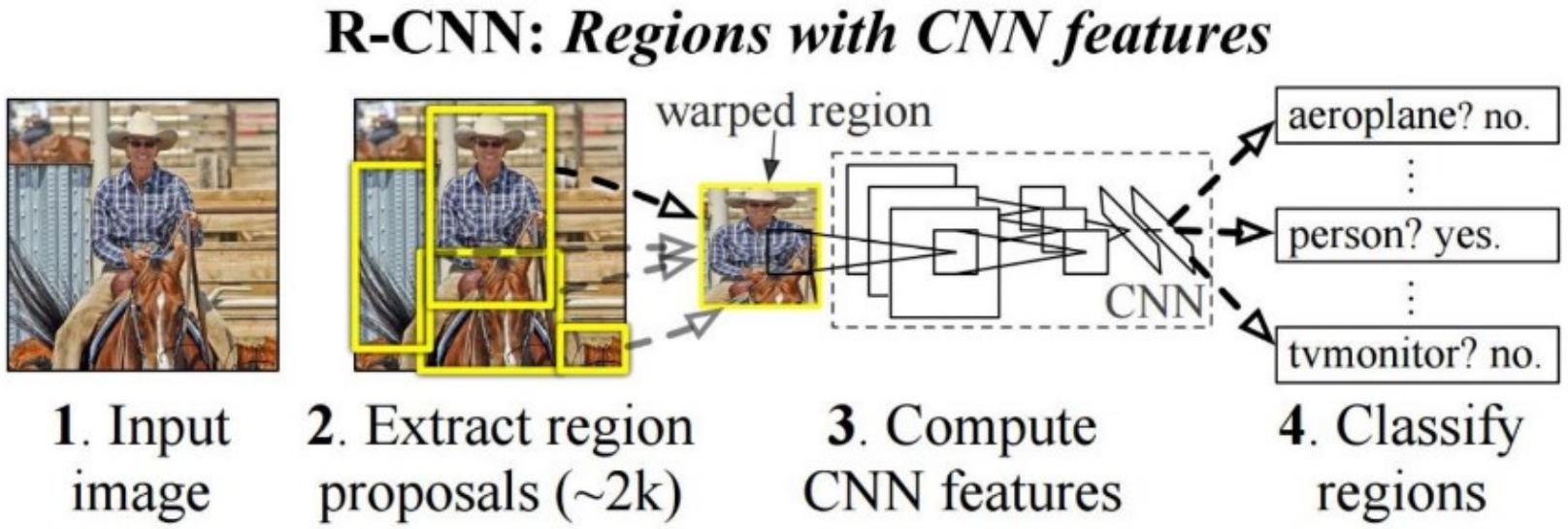
Region Proposal Method: Selective Search

- **Divides the image into multiple regions at different scales and sizes**.
- Hierarchical grouping strategy: merge adjacent regions based on various similarity measures, such as color histograms, texture gradients, and pixel intensities. This results in a large number of candidate regions covering different object instances and scales in the image.
- **Algorithm**: starts by initializing small regions based on pixel similarity and gradually merges them into larger regions using a region merging strategy.
- Uses a variety of similarity measures and segmentation techniques to determine the similarity between regions, including color histograms, texture gradients, and spatial proximity.
- Stops when a set of candidate regions covering the entire image is obtained.



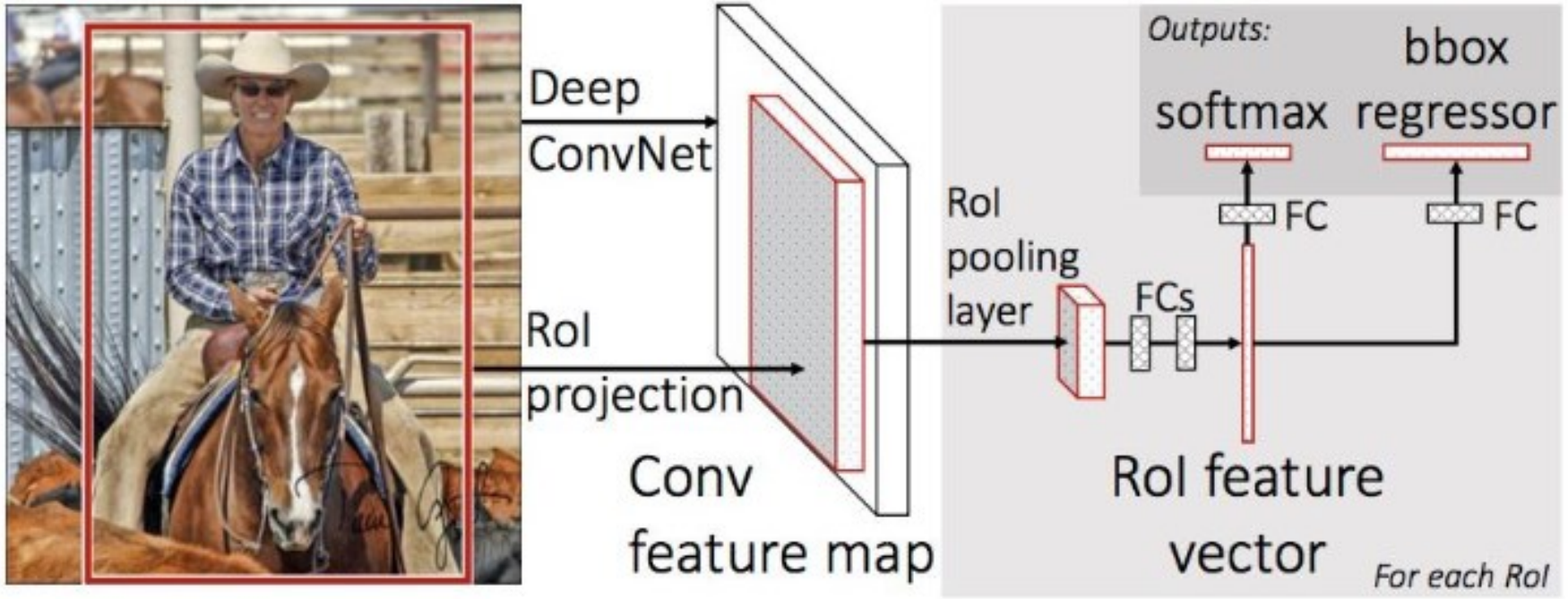
RCNN

Rich feature hierarchies for accurate object detection and semantic segmentation.



Fast - RCNN

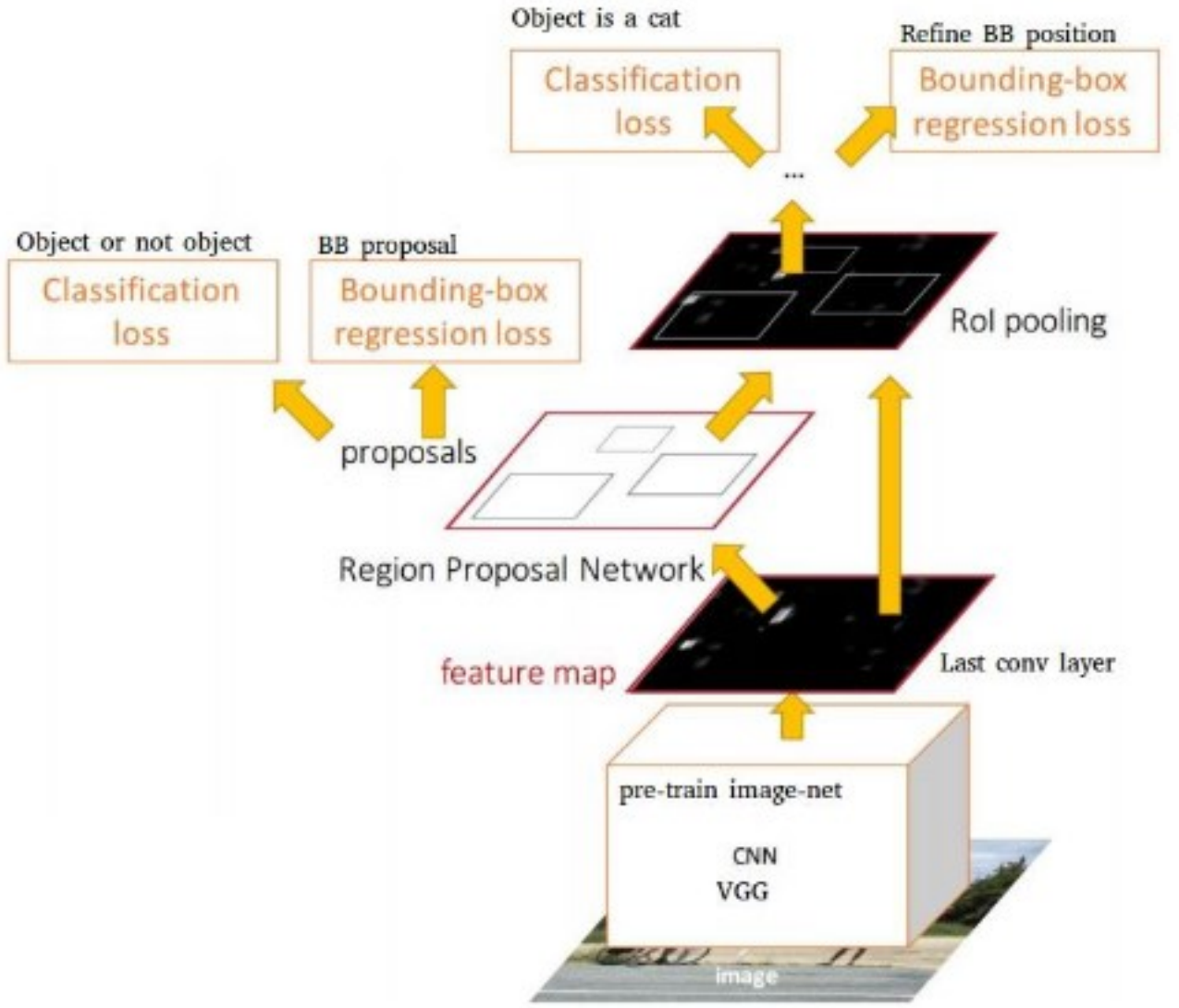
Idea: No need to recompute features for every box independently.



Regress refined bounding box coordinates.

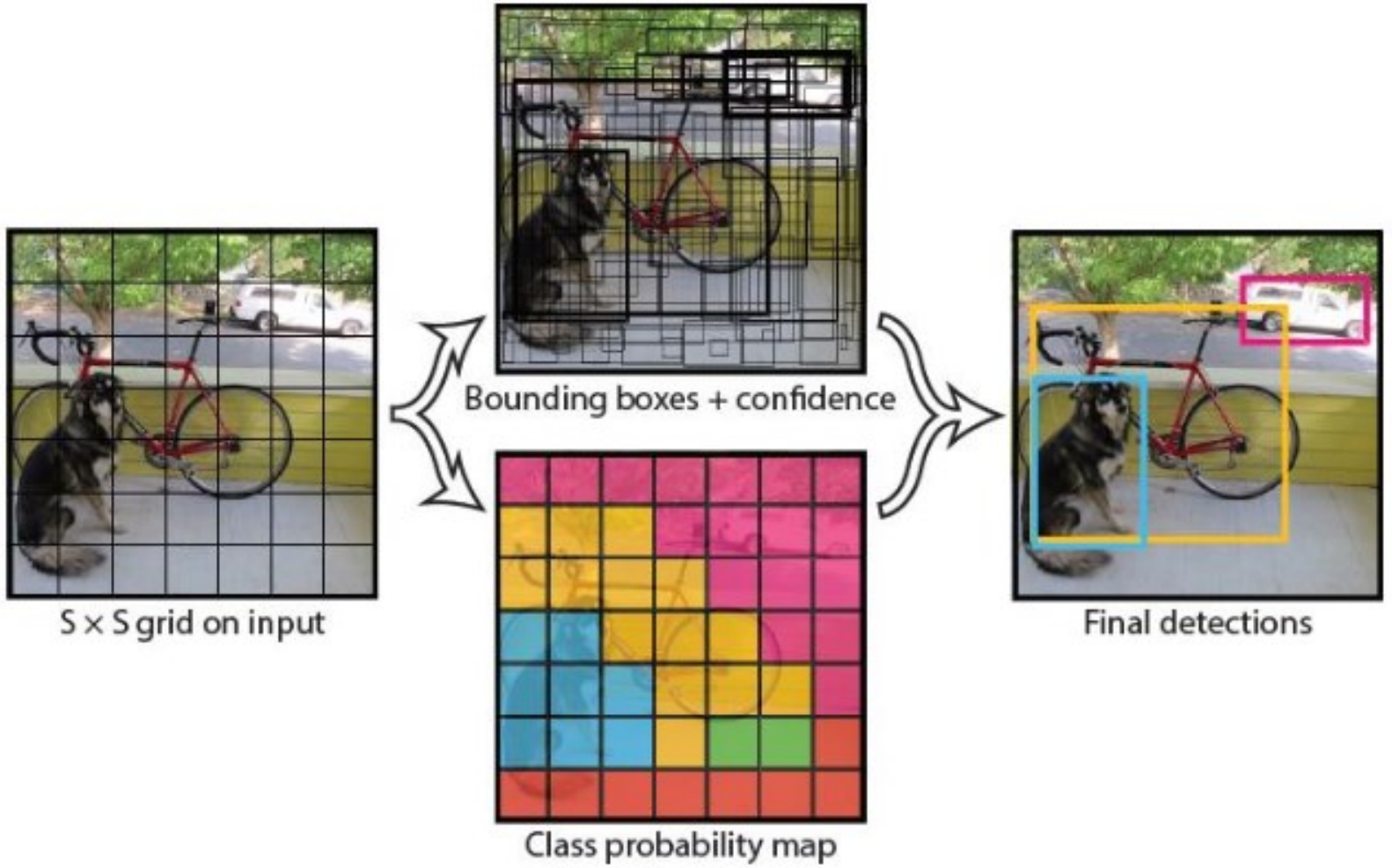
Fast - RCNN

Idea: Integrate the Bounding Box Proposals as part of the CNN predictions

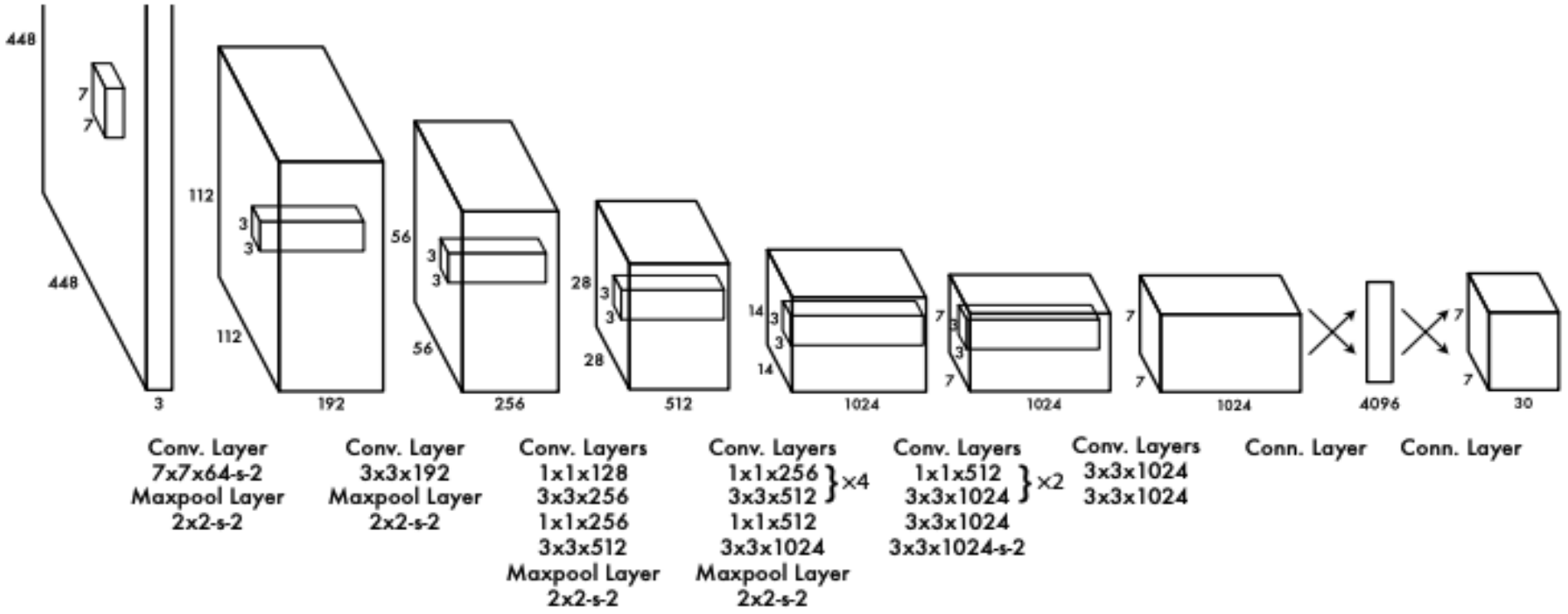


YOLO – You Only Look Once

Idea: No bounding box proposals.
Predict a class and a box for every location in a grid.



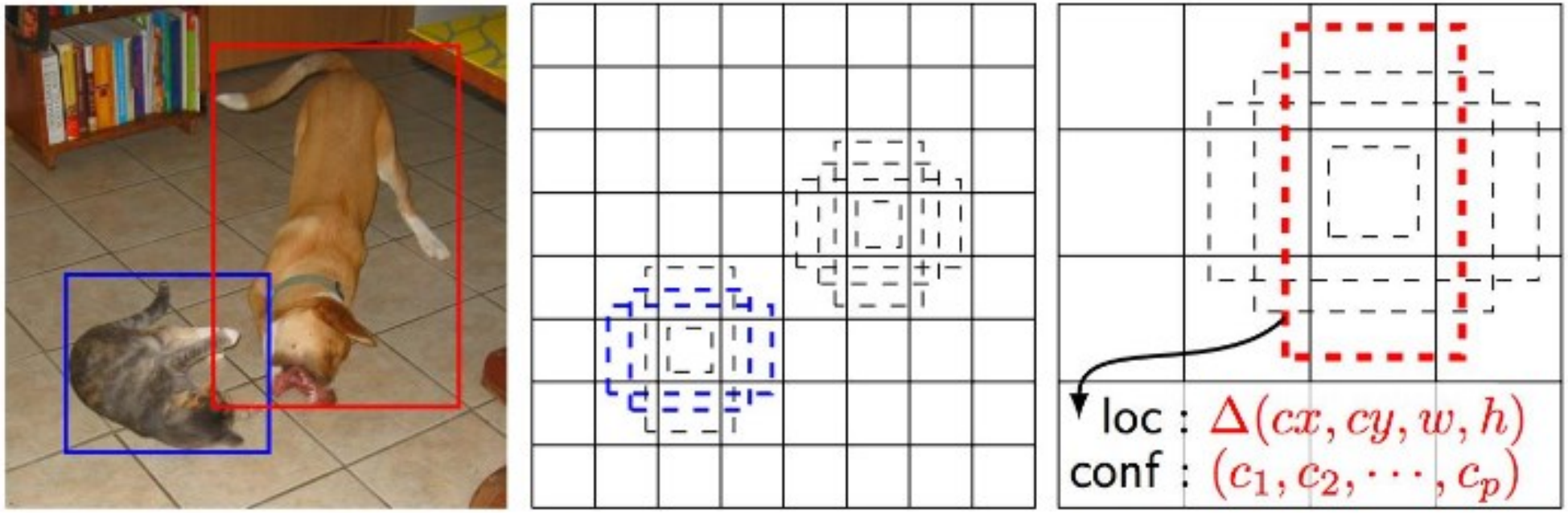
YOLO – You Only Look Once



Divide the image into 7x7 cells.
Each cell trains a detector.
The detector needs to predict the object's class distributions.
The detector has 2 bounding-box predictors to predict bounding-boxes and confidence scores.

SSD: Single Shot Detector

Idea: Similar to YOLO, but denser grid map, multiscale grid maps.



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

Comparison

	VOC 2007 (mAP)
R-CNN	54.2%
R-CNN + bbox regression	58.5%
R-CNN Faster	73.2%
Yolo	63.4%
Yolo V2	78.6%
Yolo V3	83.68%
SSD	81.06%

Non-max Suppression

How do we deal with multiple detections on the same object?

- Post-processing technique commonly used to eliminate redundant bounding boxes generated by the detection algorithm.
- The algorithm aims to select the most confident bounding boxes while removing overlapping or redundant ones.

1. Go down the box list of detections starting from highest scoring (IoU)

2. Eliminate any detection box that overlaps highly with a higher scoring detection

3. Repeat for every box

4. Select the most confident and non-overlapping boxes

End of Session 5