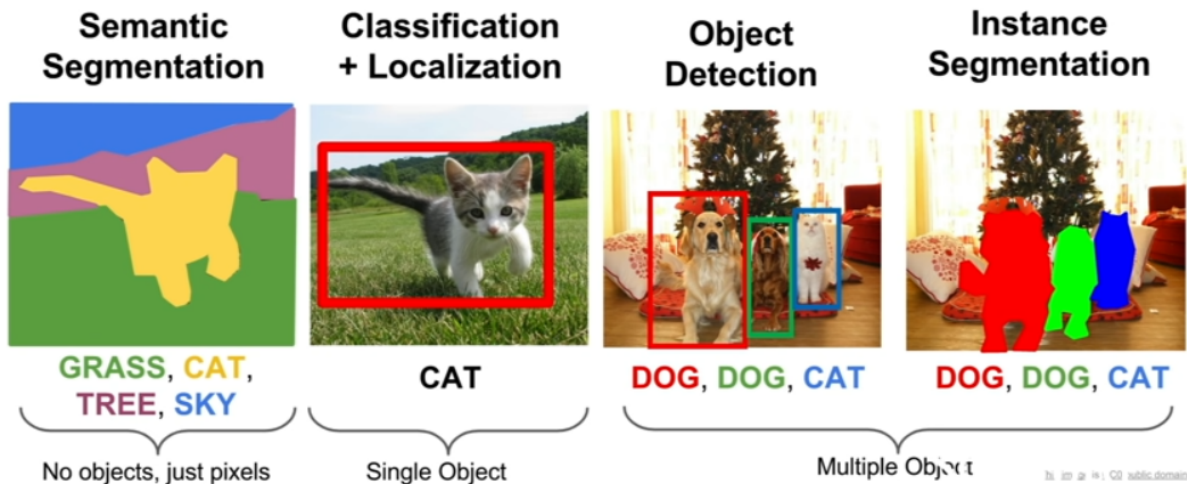


Detection & Segmentation

Other Computer Vision Tasks

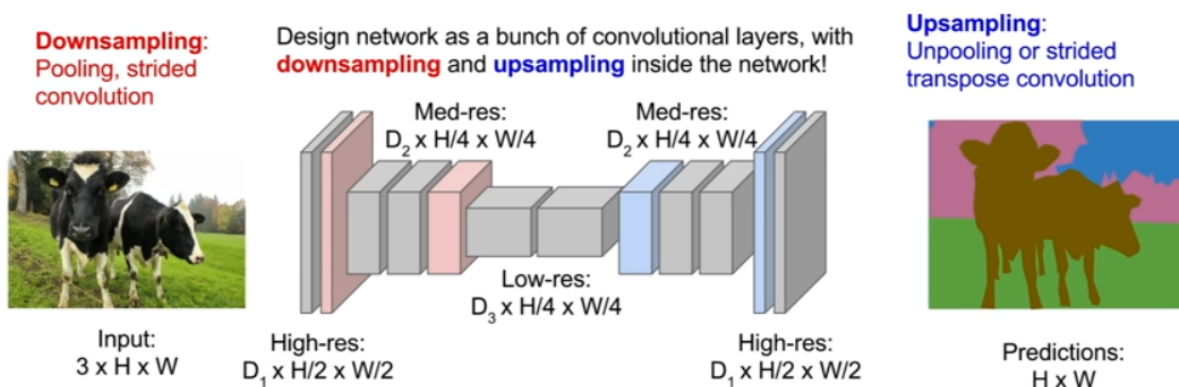


Semantic Segmentation

Label each pixel in the image with a category label

Ideas: sliding window (too expensive) → fully convolutional (downsampling & upsampling using average unpooling / max unpooling / transpose convolution)

Semantic Segmentation Idea: Fully Convolutional



Classification + Localization

Classification + Localization

The diagram illustrates a deep learning architecture for object detection, which combines classification and localization tasks. It starts with an input image of a cat, which is processed by a series of convolutional and pooling layers. The resulting feature map is then split into two parallel paths:

- Classification Path:** The feature map is passed through a fully connected layer (4096 to 1000) to produce **Class Scores** for various classes (e.g., Cat: 0.9, Dog: 0.05, Car: 0.01). These scores are then passed through a **Softmax Loss** function to determine the **Correct label** (Cat).
- Localization Path:** The feature map is also passed through a fully connected layer (4096 to 4) to produce **Box Coordinates** (x, y, w, h). These coordinates are then passed through an **L2 Loss** function to determine the **Correct box** (x', y', w', h').

The final output is a bounding box around the cat in the original image, labeled "Cat".

Classification + Localization

Correct label: Cat

Softmax Loss

Class Scores

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

Fully Connected: 4096 to 1000

Vector: 4096

Fully Connected: 4096 to 4

Box Coordinates (x, y, w, h)

L2 Loss

Correct box: (x', y', w', h')

Treat localization as a regression problem!

This image is CC0 public domain

Object Detection

Object detection as classification: sliding window (too expensive)

Region proposals: R-CNN, $\sim 2k$ regions of interest \rightarrow CNN, separated

Fast R-CNN, whole image \rightarrow convolutional feature map \rightarrow CNN

Faster R-CNN, insert Region Proposal Network (RPN) - faster region proposal

Detection without Proposals: YOLO (you only look once), SSD (single shot detector)

$$\text{Intersection over Union (IoU)} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Mean Average Precision (mAP) metric:

- Calculate the precision and recall metrics.
- Calculate the area under the precision-recall curve.
- Measure the average precision.

With YOLO, you only look once at an image to perform detection

- We split the image into a grid
- Each cell predicts boxes and confidences: $P(\text{Object})$
- Each cell also predicts a class probability
- Conditioned on object: $P(\text{Car} \mid \text{Object})$

- Then we combine the box and class predictions
- Finally we do Non-maximum Suppression (NMS) and threshold detections

R-CNN: Region based ConvNets for Object Detection, Regions of Interest (RoI) from a proposal method (~2k) on input images

Fast R-CNN: Forward whole image through ConvNet, RoI from a proposal method on “conv5” feature maps (after ConvNet)

Faster R-CNN: Solely based on CNN, no external region proposals, Region Proposal Net after feature map instead

Mask R-CNN: Faster R-CNN with FCN on RoIs, mask other areas