https://www.sli.do/
#073374

Deep Learning for Computer Vision (IV)

Learning Objectives

- Learn the basics of advanced computer vision tasks beyond image classification: image segmentation and object detection.

Beyond image classification: object detection and image segmentation

C-S David Chen, Department of Civil Engineering, National Taiwan University

Single-label multi-class classification

- (•) Biking
- ( ) Running
- ( ) Swimming

Multi-label classification

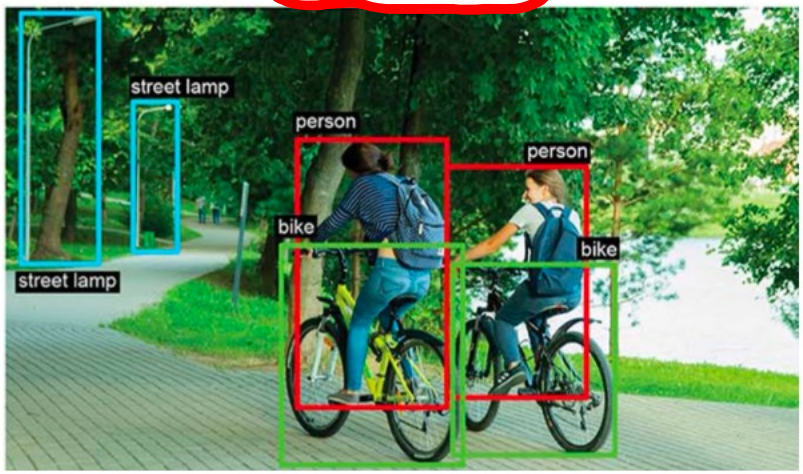
- [x] Bike
- [x] Person
- [ ] Boat
- [x] Tree
- [ ] Car
- [ ] House

Label 非常简单

Image segmentation

x
y

Object detection



Object Detection

- The goal is to draw rectangles (called *bounding boxes*) around objects of interest in an image, and associate each rectangle with a class.
- A self-driving car could use an object-detection model to monitor cars, pedestrians, and signs in view of its cameras, for instance.

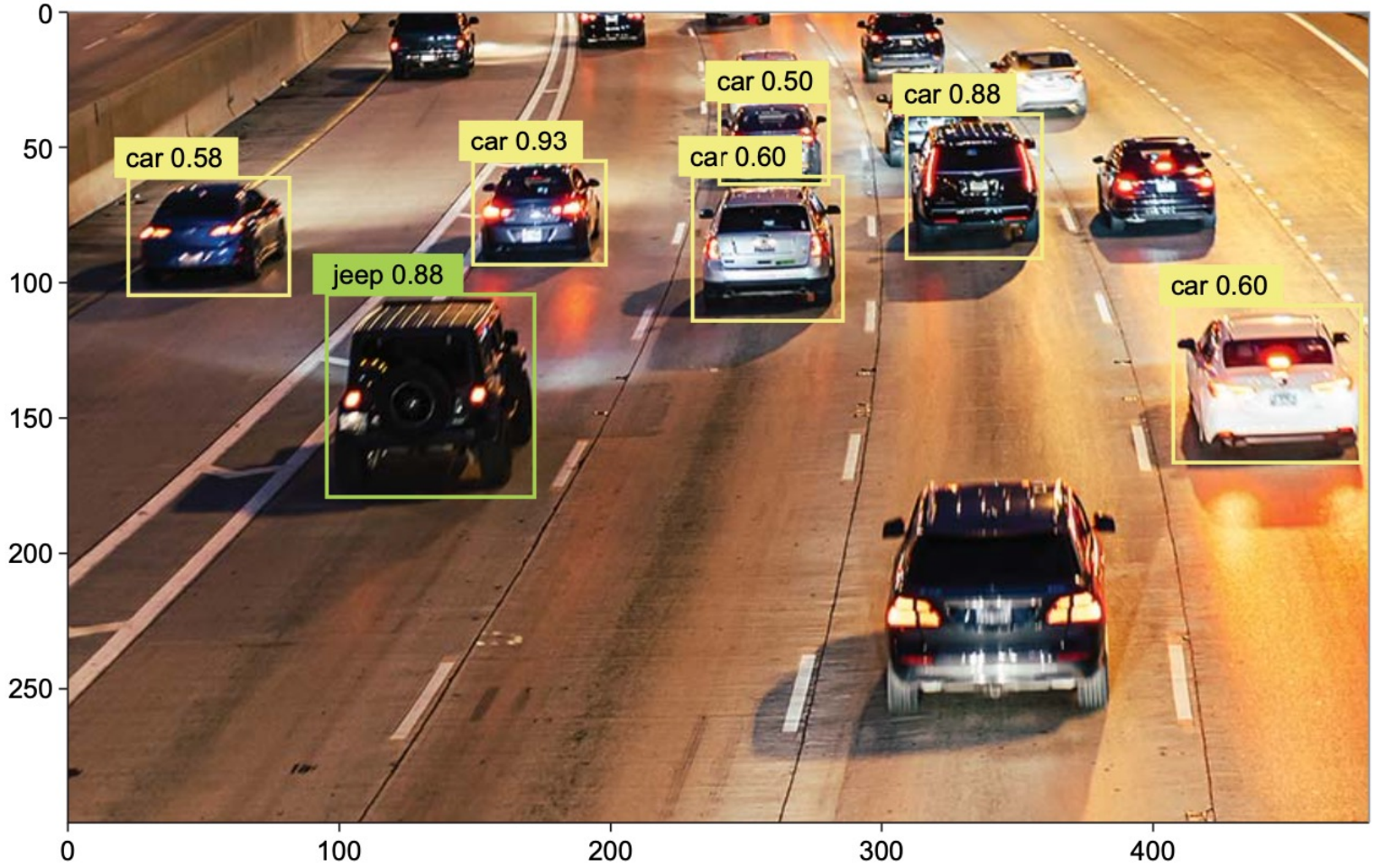
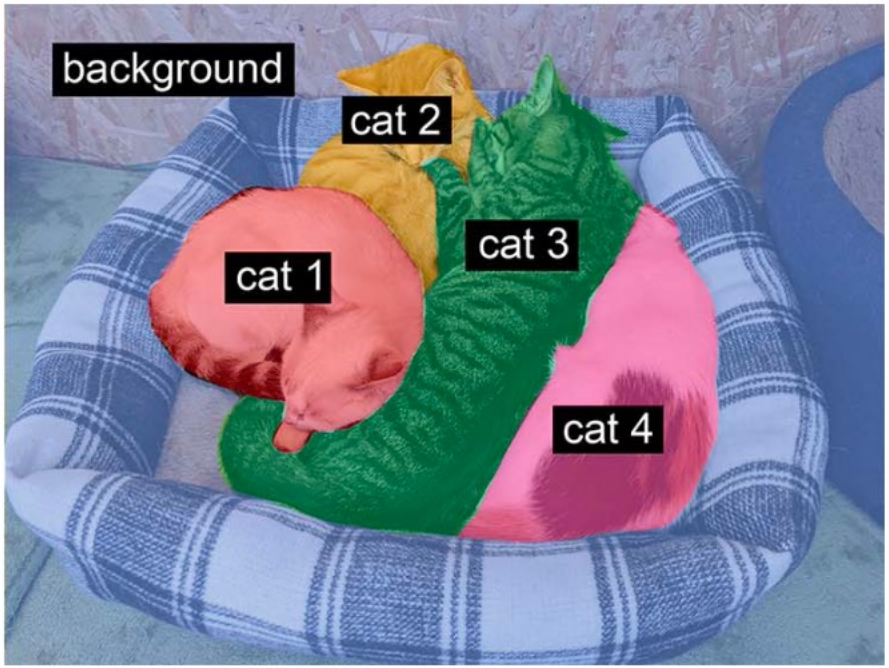


Image Segmentation

- The goal is to “segment” or “partition” an image into different areas, with each area usually representing a category.

Semantic segmentation



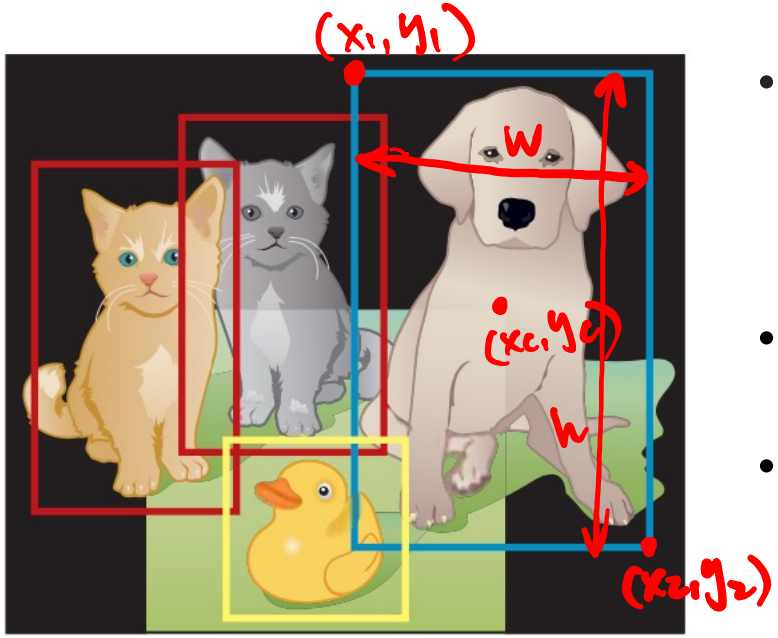
Instance segmentation

Learn the basics of object detections

C-S David Chen, Department of Civil Engineering, National Taiwan University

Object Detection

- **Goal**: predict the location of objects in an image via bounding boxes and the classes of the located objects
- **Input**: an image with one or more objects
- **Output**: one or more bounding boxes (defined by coordinates $[x_1, y_1, x_2, y_2]$ or $[x_c, y_c, w, h,]$) and a class label for each bounding box



Cat, Cat, Duck, Dog

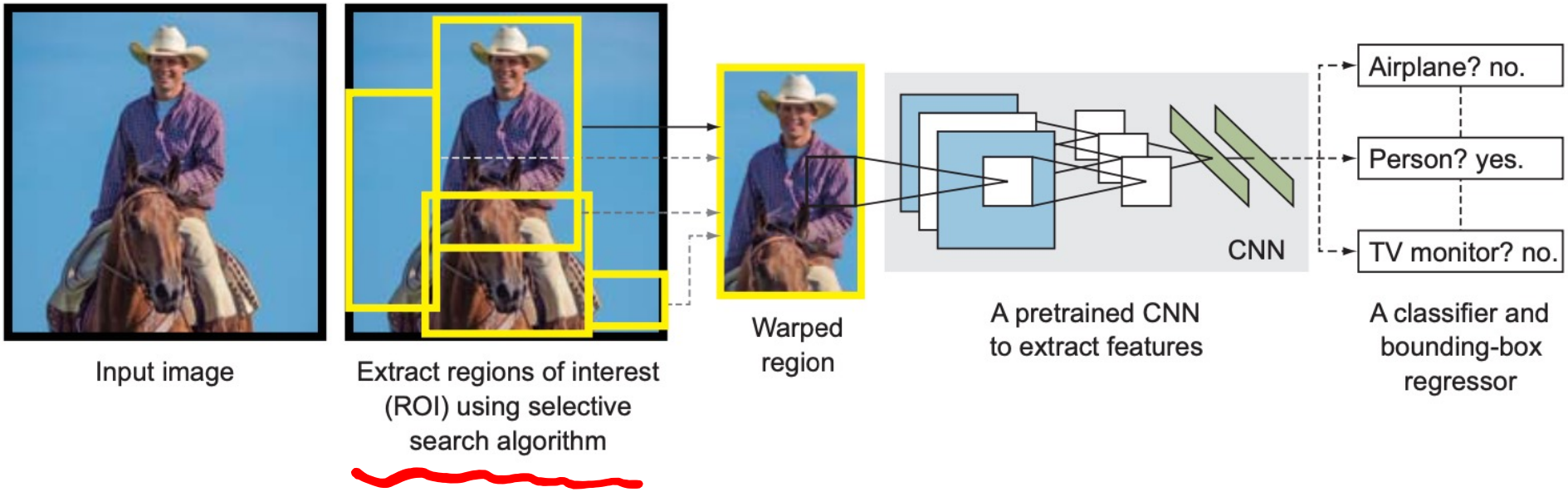
- The two most popular object detection systems are the **R-CNN** family of networks and the **YOLO** family of networks.
- R-CNN is a multi-stage detector and YOLO is a single-stage detector.
- In general, single-stage detectors tend to be less accurate than two-stage detectors but are significantly faster.

Object Detection: R-CNN Family

- The R-CNN family of networks has three main variations: R-CNN, Fast R-CNN, and Faster R-CNN. R-CNN and Fast R-CNN use **a selective search algorithm to propose RoIs**, whereas Faster R-CNN is an end-to-end DL system that uses **a region proposal network to propose RoIs**.

Region of Interests

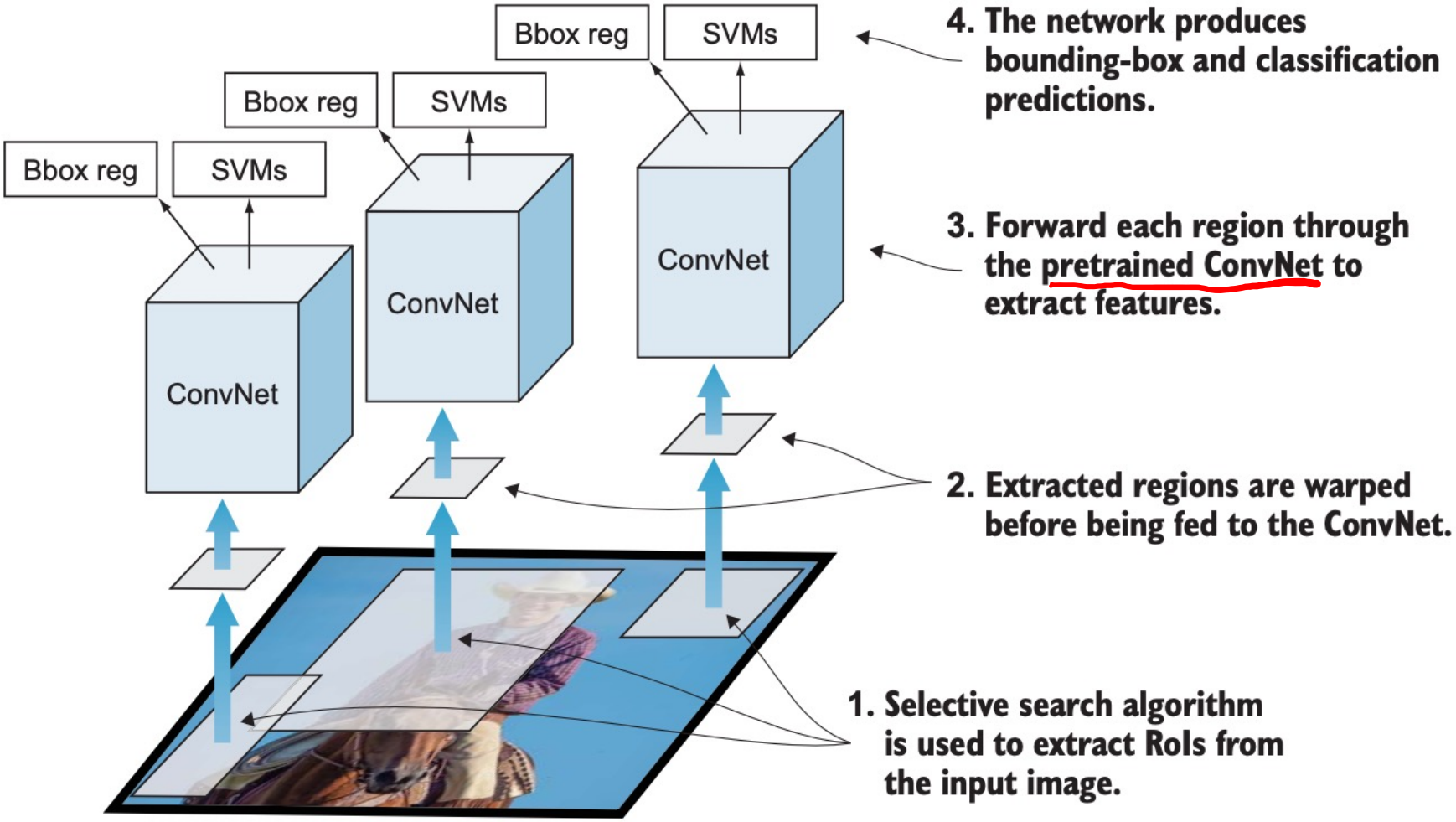
R-CNN



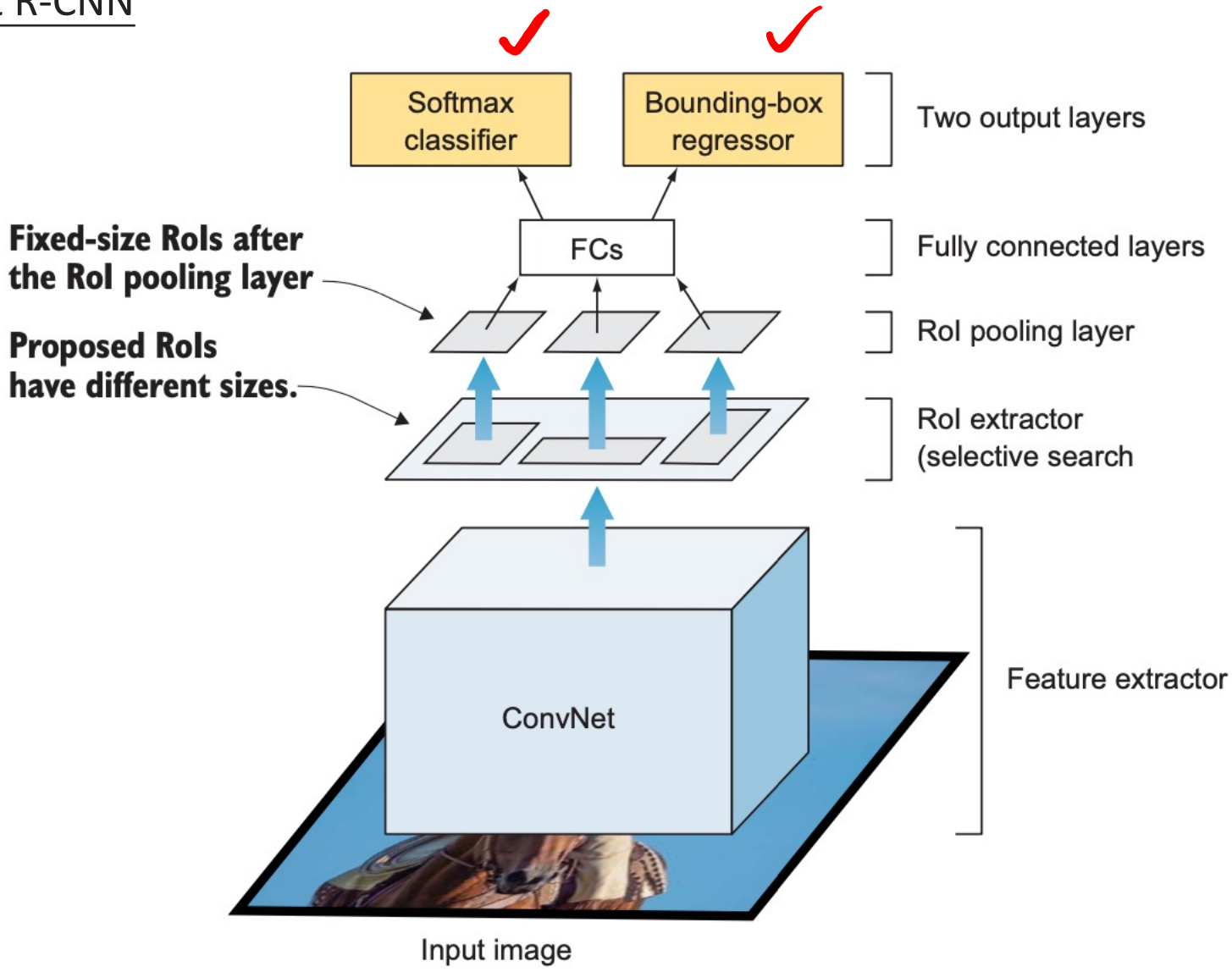
Selective Search Algorithm

- A greedy search algorithm that is used to provide region proposals that potentially contain objects.
- It tries to find areas that might contain an object by combining similar pixels and textures into rectangular boxes.

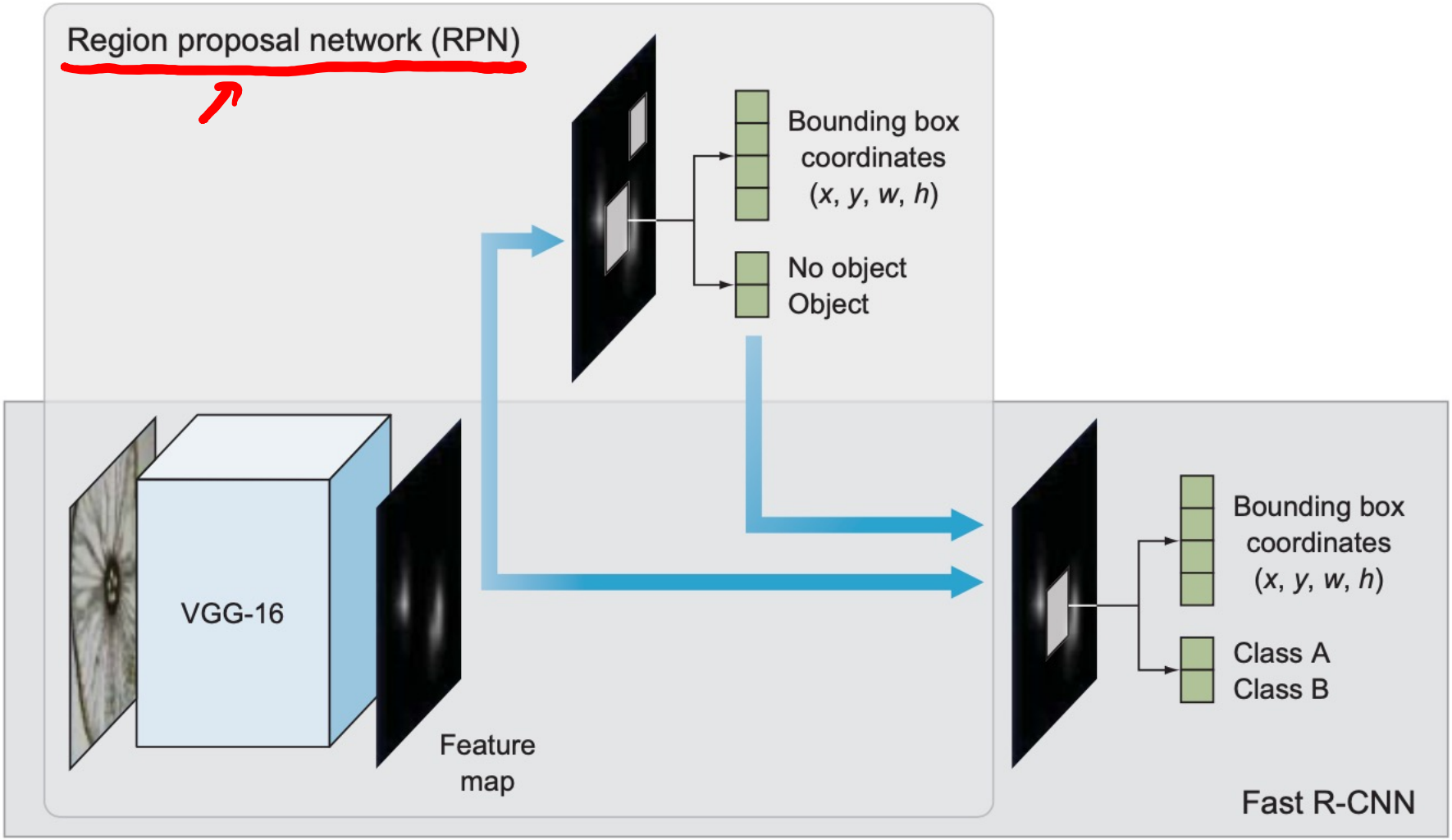
R-CNN


Bbox reg
SVMs
Bbox reg
SVMs
Bbox reg
SVMs
ConvNet
ConvNet
ConvNet
4. The network produces bounding-box and classification predictions.
3. Forward each region through the pretrained ConvNet to extract features.
2. Extracted regions are warped before being fed to the ConvNet.
1. Selective search algorithm is used to extract RoIs from the input image.

Fast R-CNN

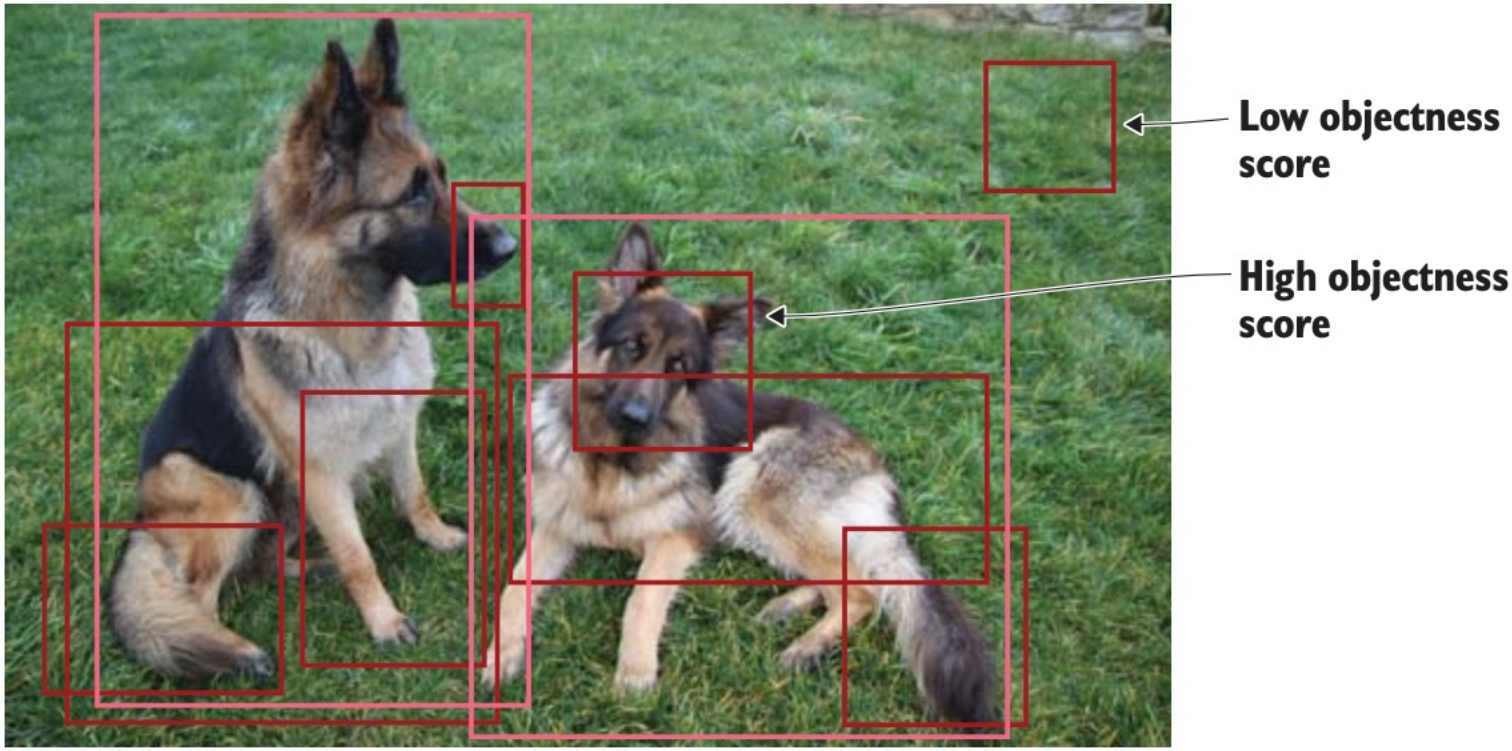

Softmax classifier
Bounding-box regressor
Two output layers
FCs
Fully connected layers
Fixed-size RoIs after the RoI pooling layer
RoI pooling layer
Proposed RoIs have different sizes.
RoI extractor (selective search
ConvNet
Feature extractor
Input image

Faster R-CNN (RPN + Fast R-CNN)



Region Proposals

Regions of interest (RoIs) proposed by the system. Regions with high objectness score represent areas of high likelihood to contain objects (foreground), and the ones with low objectness score are ignored because they have a low likelihood of containing objects (background).



Object Detection: R-CNN Family

- *R-CNN*—Bounding boxes are proposed by the selective search algorithm. Each is warped, and features are extracted via a deep convolutional neural network such as AlexNet, before a final set of object classifications and bounding box predictions is made with linear SVMs and linear regressors.
- *Fast R-CNN*—A simplified design with a single model. An RoI pooling layer is used *after* the CNN to consolidate regions. The model predicts both class labels and RoIs directly.
- *Faster R-CNN*—A fully end-to-end DL object detector. It replaces the selective search algorithm to propose RoIs with a region proposal network that interprets features extracted from the deep CNN and learns to propose RoIs directly.

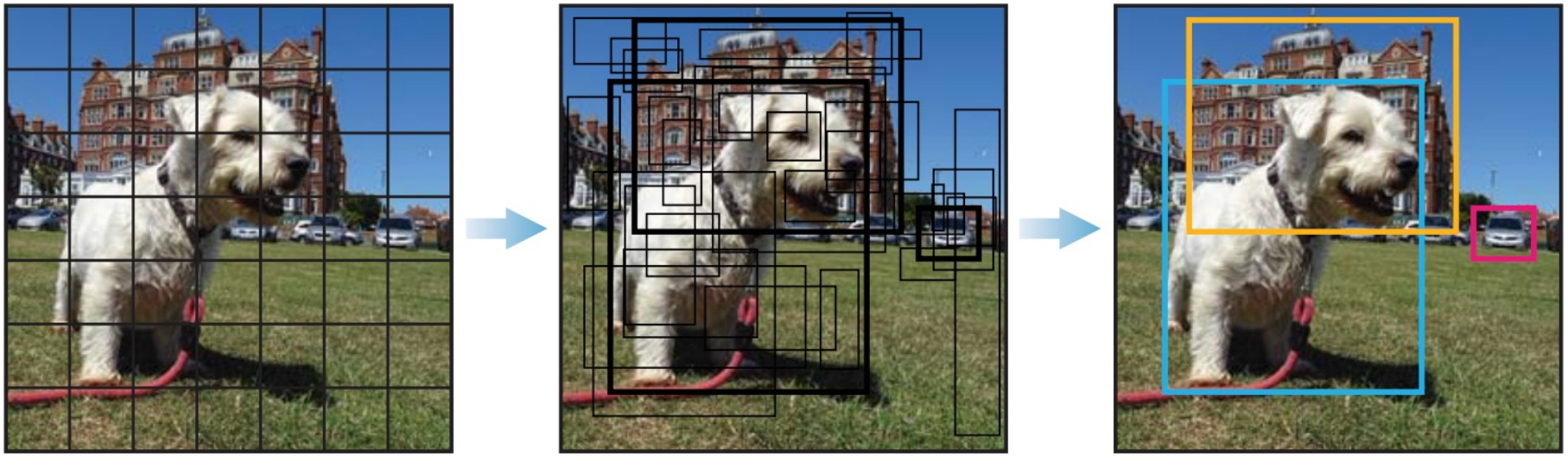
Detectron2

1. **Detectron2**
   a. **Data preparation (dataset registration)**
      `DatasetCatalog`
      `MetadataCatalog`
   b. **Configuration setup (hyper-parameters)**
      `cfg = get_cfg()`
   c. **Build a trainer**
      `DefaultTrainer(cfg)`
   d. **COCO evaluation**
      `COCOEvaluator`
   e. **Inference by predictor**
      `DefaultPredictor`
2. **Rebar hook angle detection**

   Train your first Detectron2 detector

Object Detection: YOLO Family

- The YOLO family is a series of end-to-end DL models designed for fast object detection, and it was among the first attempts to build a fast real-time object detector.
- Although the accuracy of the models is close but not as good as R-CNNs, they are popular for object detection because of their detection speed, often demonstrated in real-time video or camera feed input.
- For a good introduction, see https://aiacademy.tw/yolo-v4-intro



Splits the image into grids

Predicts bounding boxes and classifications

Final predictions after non-maximum suppression

Darknet (YOLO v4)

1. **Darknet**
   a. **Data preparation (dataset registration)**
      `.jpg/.txt (images/annotations)`
      `train.txt/valid.txt`
   b. **Configuration setup (hyper-parameters)**
      `Obj.data`
      `Obj.names`
      `yolov4.cfg`
   c. **Training**
      `./darknet detector train obj.data yolov4.cfg yo.weights -map`
   d. **Evaluation**
      `./darknet detector map obj.data yolov4.cfg yo.weights -iou_thresh 0.5`
   e. **Image prediction**
      `./darknet detector test yolov4.cfg yo.weights image`
2. **Construction machine detection**

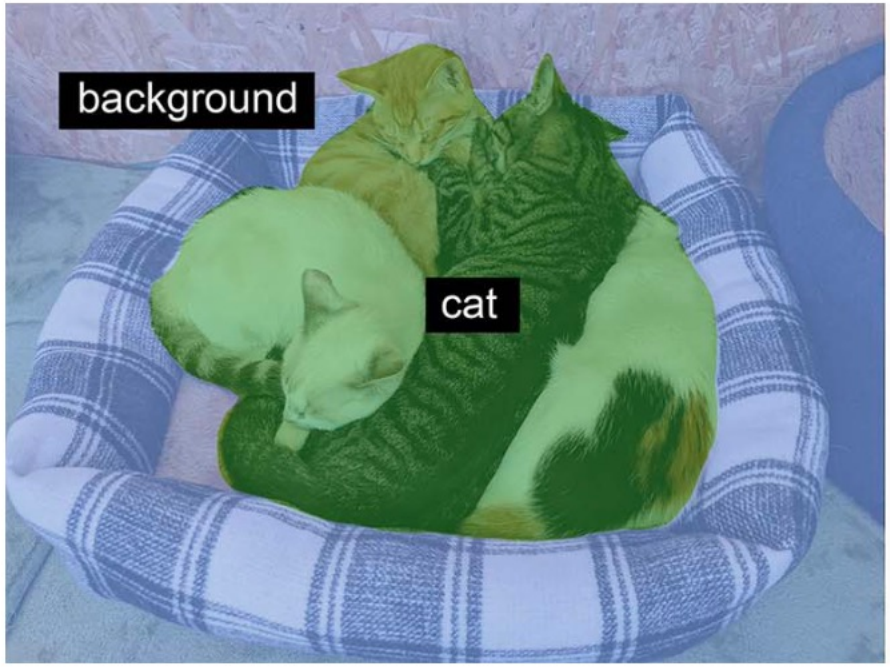
   Train your first YOLO v4 detector

Learn the basics of image segmentation

C-S David Chen, Department of Civil Engineering, National Taiwan University

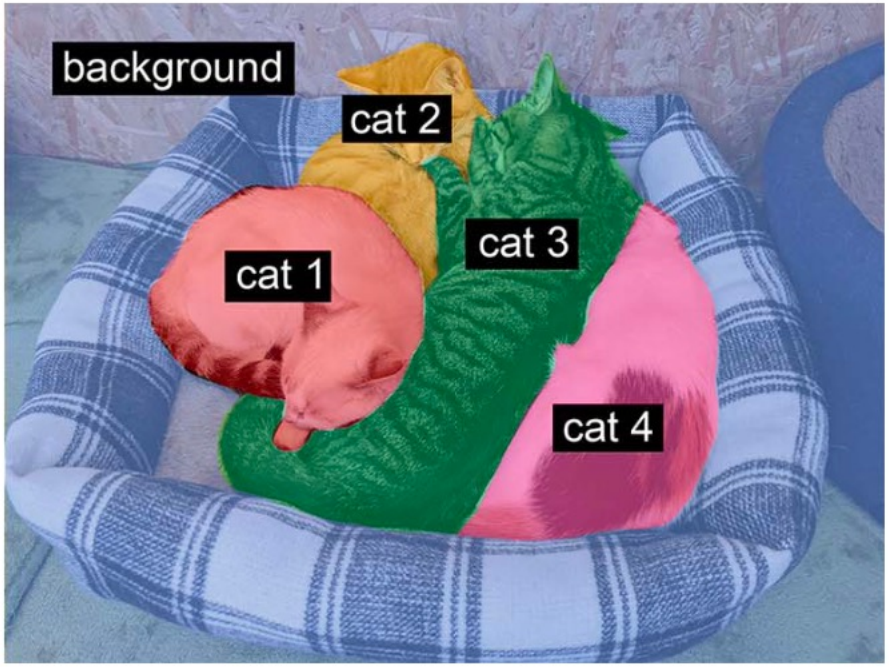
Image Segmentation

- The goal is to “segment” or “partition” an image into different areas, with each area usually representing a category.



Semantic segmentation

each pixel is independently classified into a semantic category, like “cat.” If there are two cats in the image, the corresponding pixels are all mapped to the same generic “cat” category

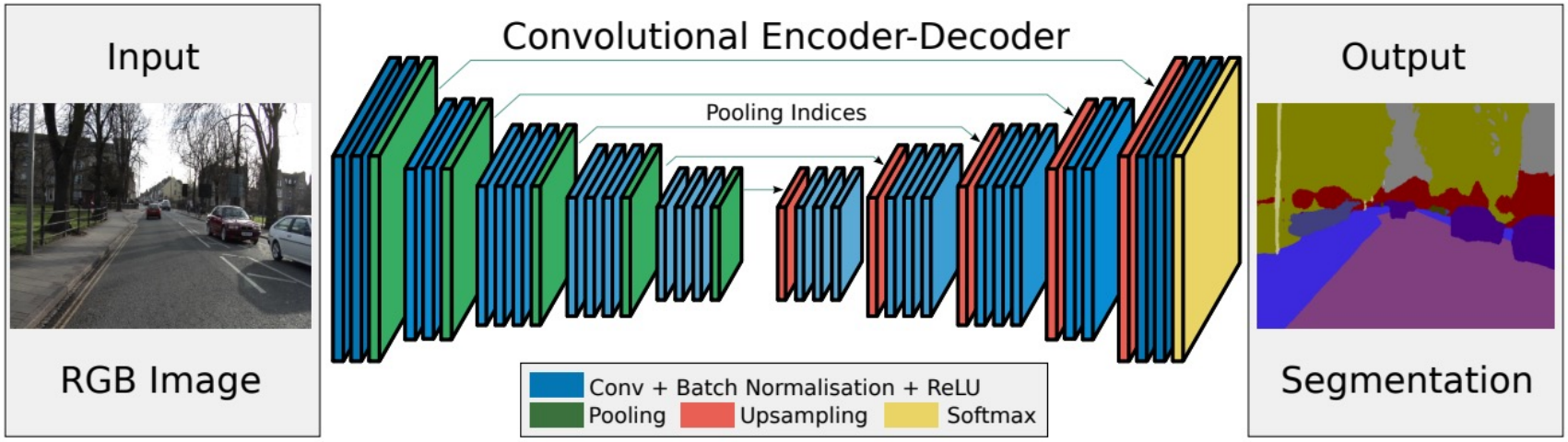


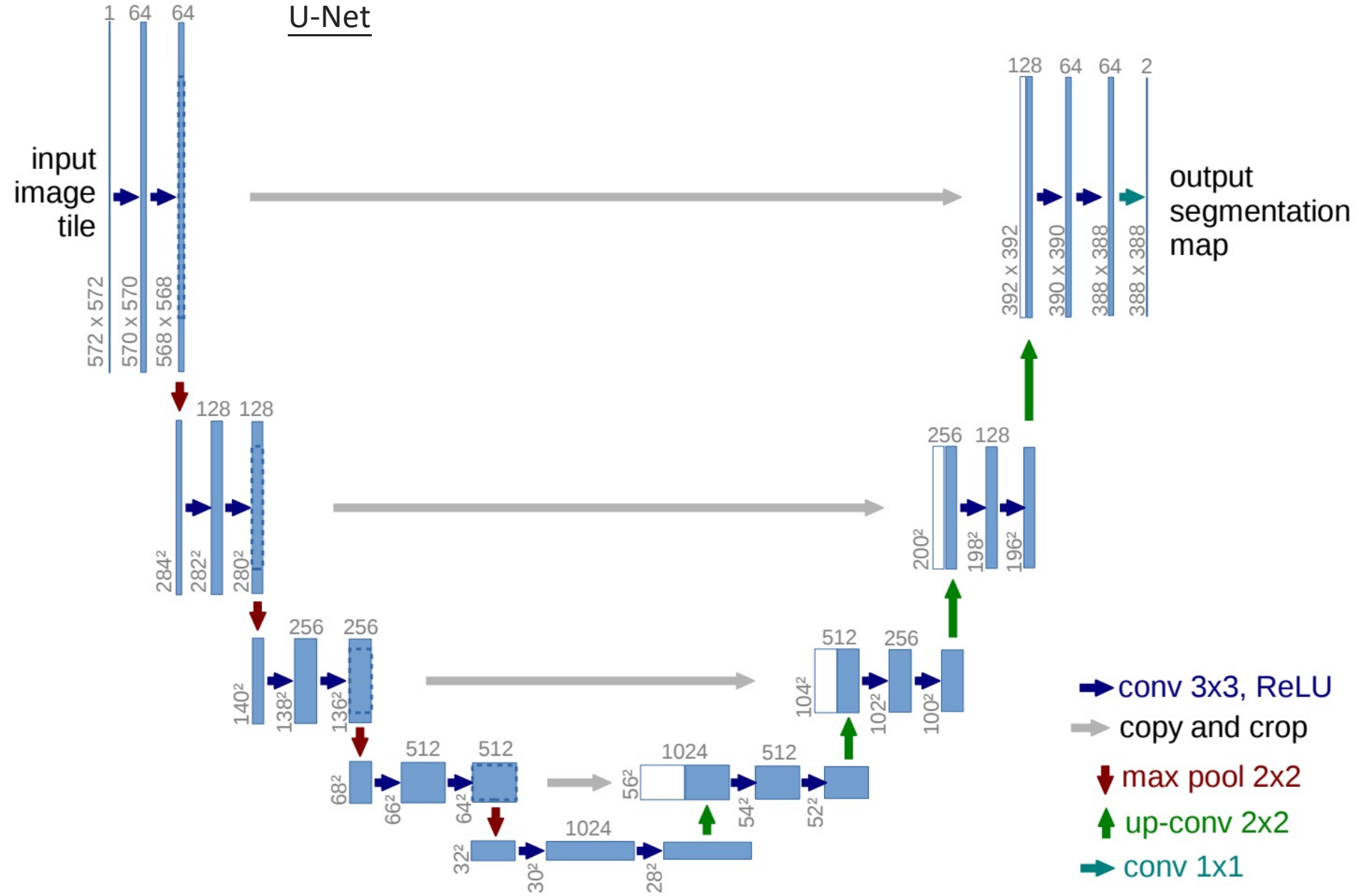
Instance segmentation

seeks not only to classify image pixels by category, but also to parse out individual object instances. In an image with two cats in it, instance segmentation would treat “cat 1” and “cat 2” as two separate classes of pixels

Semantic Segmentation

- Convolution **Encoder**: closely resembles the kind of convnet you'd use for image classification: a stack of Conv2D layers, with gradually increasing filter sizes. The purpose is to encode the images into smaller feature maps, where each spatial location (or pixel) contains information about a large spatial chunk of the original image. You can understand it as a kind of compression.
- Convolution **Decoder**: apply a kind of inverse of the transformations we've applied so far—something that will upsample the feature maps instead of downsampling them.
- A Simple Encoder-Decoder Example for Semantic Segmentation



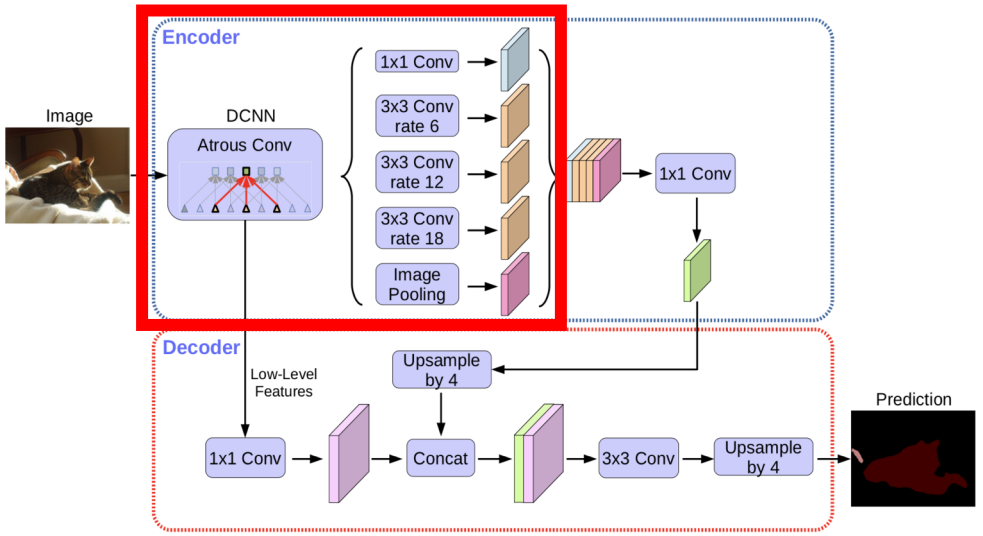

U-Net
input image tile
output segmentation map
1 64 64
572 x 572
570 x 570
568 x 568
128 128
284²
282²
280²
256 256
140²
138²
136²
512 512
68²
66²
64²
1024
32²
30²
28²
1024
56²
512
54²
512
52²
512 256
104²
102²
100²
256 128
200²
198²
196²
128 64 64 2
392 x 392
390 x 390
388 x 388
388 x 388
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

Semantic Segmentation (Pixel Level Recognition)
Google DeepLab V3+

Encoder-Decoder Structure

Encoder: gradually reduces the feature maps and captures higher semantic information

Decoder: gradually recovers the spatial information



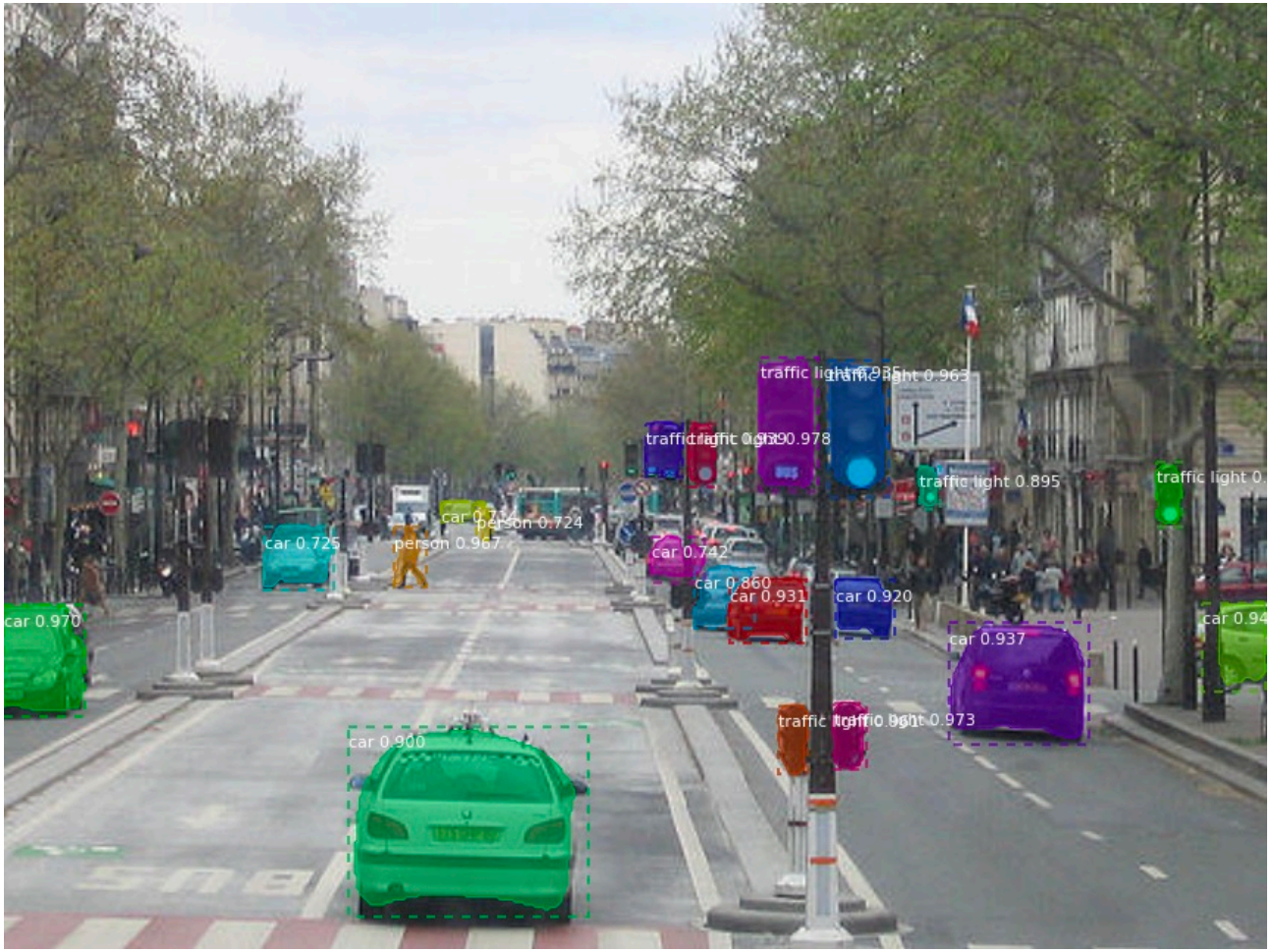
Atrous Spatial Pyramid Pooling

Apply several parallel atrous convolution with different rates

Depthwise separable convolution

Reduce the computation cost and number of parameters while maintaining similar performance.

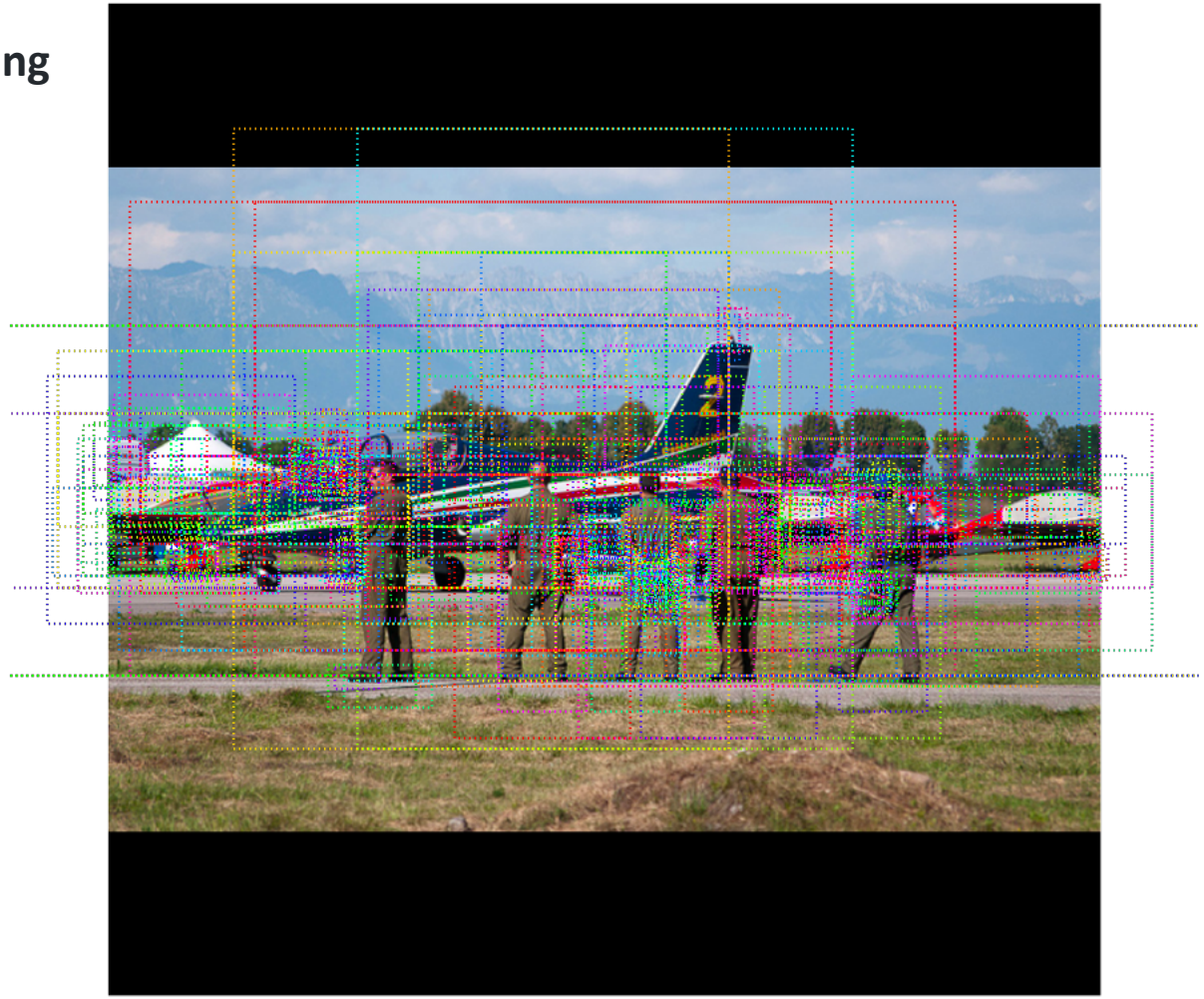
Instance Segmentation (Pixel Level Recognition) Mask R-CNN



Kaiming He et al. (2018). Mask R-CNN. axXiv:1703.06870

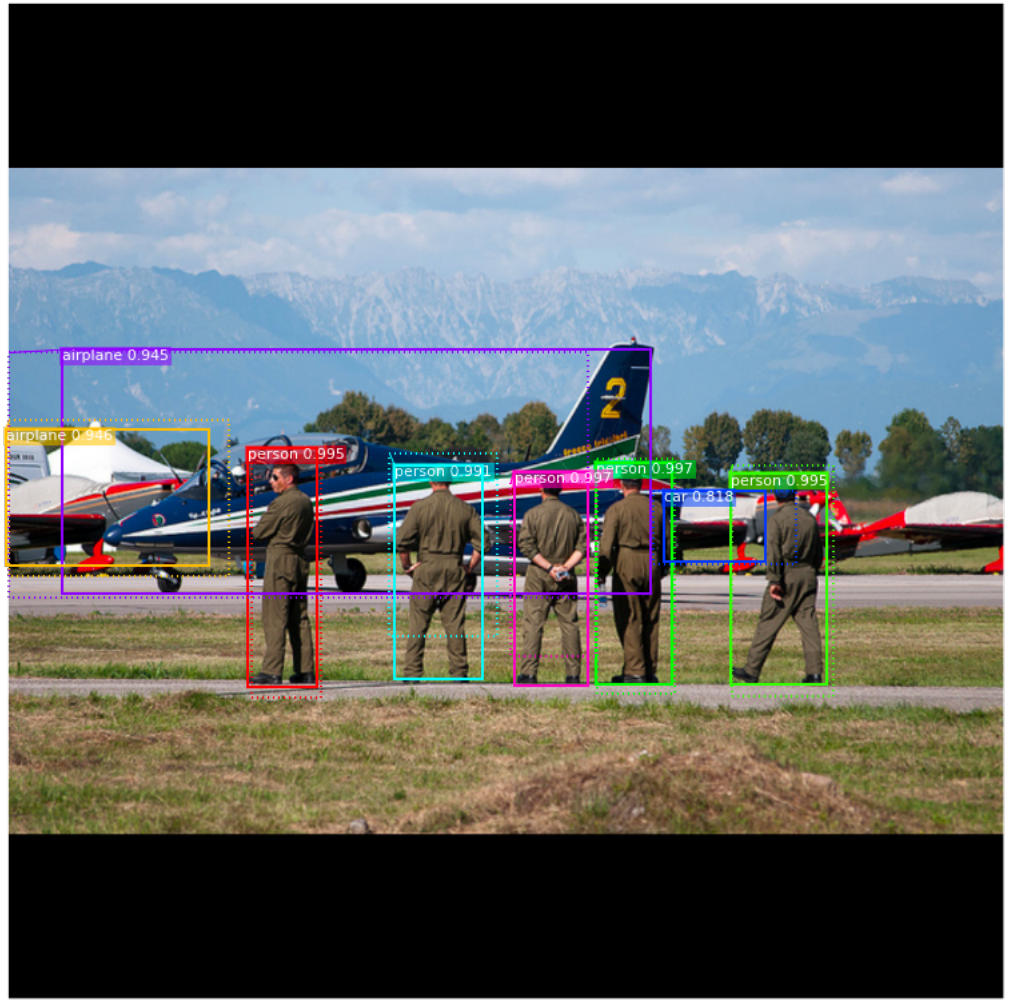
Instance Segmentation (Pixel Level Recognition) Mask R-CNN

1. Anchor sorting and filtering



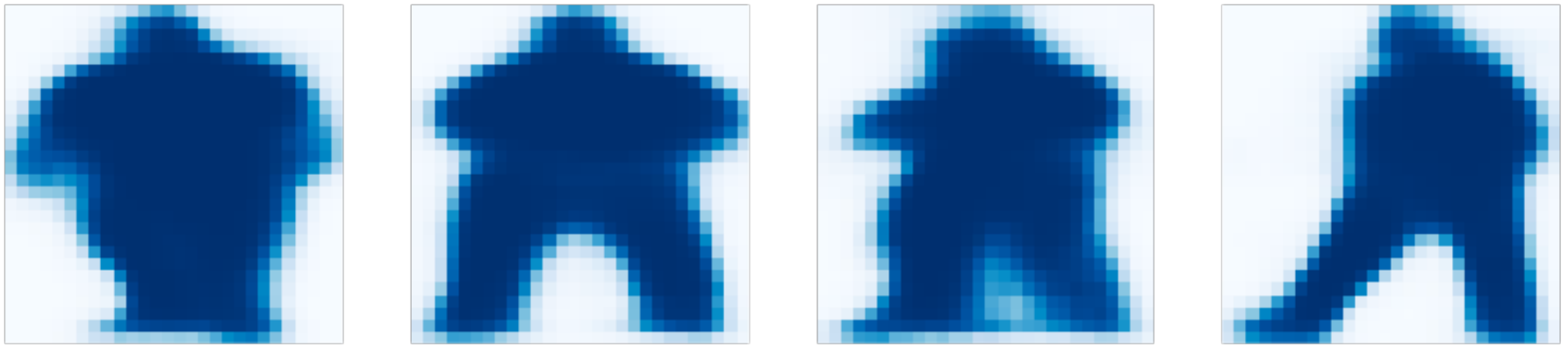
Instance Segmentation (Pixel Level Recognition) Mask R-CNN

2. Bounding Box Refinement



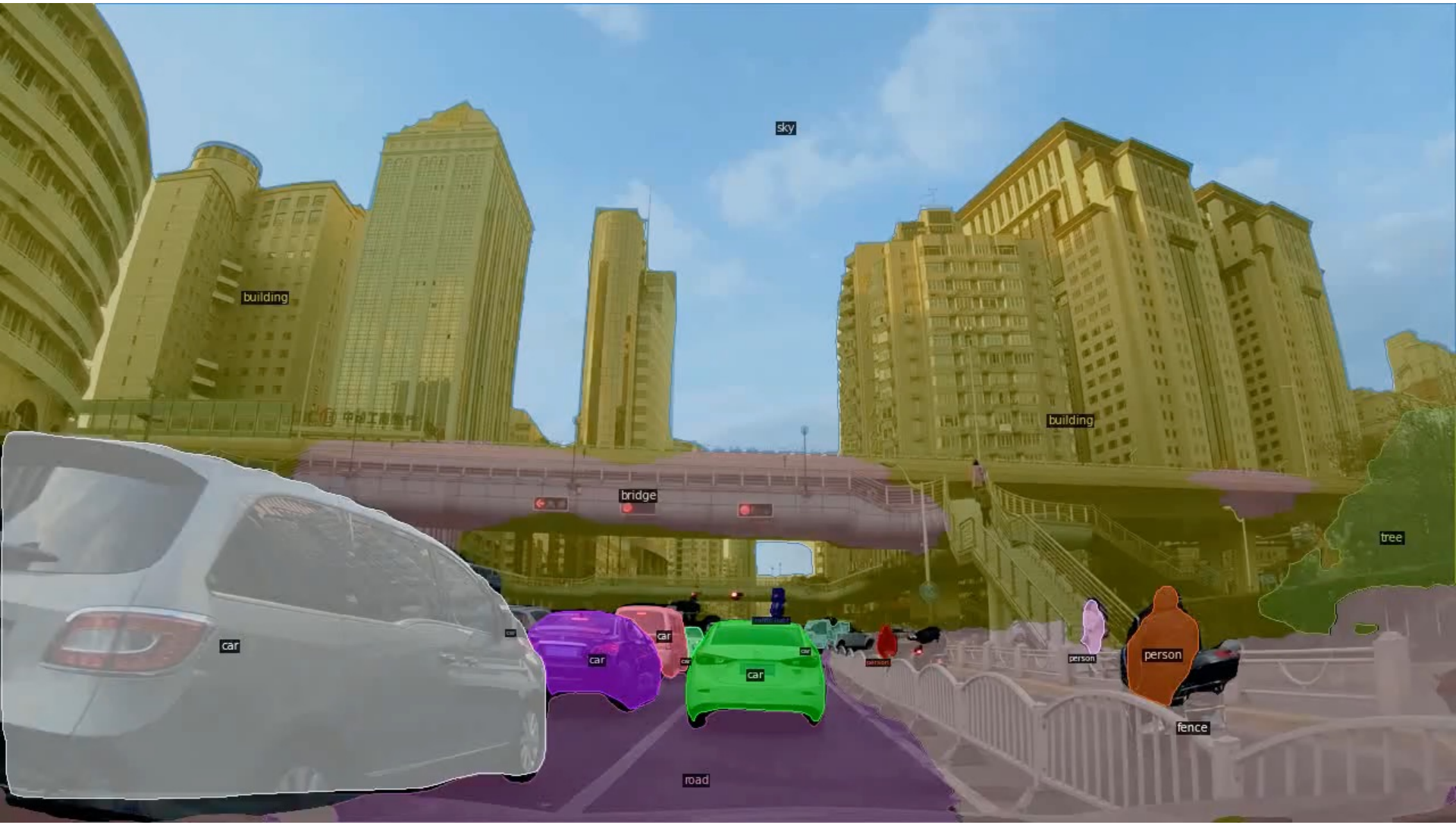
Instance Segmentation (Pixel Level Recognition) Mask R-CNN

3. Mask Generation



Instance Segmentation (Pixel Level Recognition) Mask R-CNN

4. Composing the different pieces into a final result


sky
building
building
bridge
tree
car
car
car
car
person
person
fence
road

HW7