Convolutional Neural Networks 3 (Ch 14)

Wojtek Kowalczyk

wojtek@liacs.nl

Agenda

- Transfer learning
- Classification and Localization
- Object Detection
- Semantic Segmentation
- Fully Convolutional Networks

Transfer Learning

- Does it make sense to apply a network that is trained on the ImageNet collection to other image recognition problems? **YES!**

- How to deal with:
  - different image resolutions
  - black&white/gray levels vs. color images
  - various object types (medical imaging, steel inspection data, archeological objects, …)
  - various sensors: spectral imaging, radar/sonar/laidar,…
  - …

Transfer Learning = adapting a model trained on one task to another task

How?

- Use the first few layers (their weights) of a network that is pretrained on the ImageNet data (or another big collection of images), e.g., the ResNet

=> *they contain "knowledge" of "important/low level" features*

- Rescale (when needed) your new data to fit the ResNet input format

- In case of black&white/gray images => triplicate them! (e.g., 256x256 -> 3x256x256) -> it works! WHY?

Main advantages of transfer learning

- Huge savings network training/development time
- Superior accuracy on relatively small training sets (usually, labeled(!) collections of images are small)

Classification and Localization

- How to find a specific item in an image (e.g, a traffic sign, a face, a flower, etc.) ?
- We need a training set: annotated images: locations of objects and their class labels
- Localization => the center of the bounding box:(x,y) and its height and width (h,w)

- Images must be annotated: equipped with localizations and class labels (that's a difficult part!) => regression problem => MSE

- "Intersection over union" as a "true" model quality measure

Intersection over Union

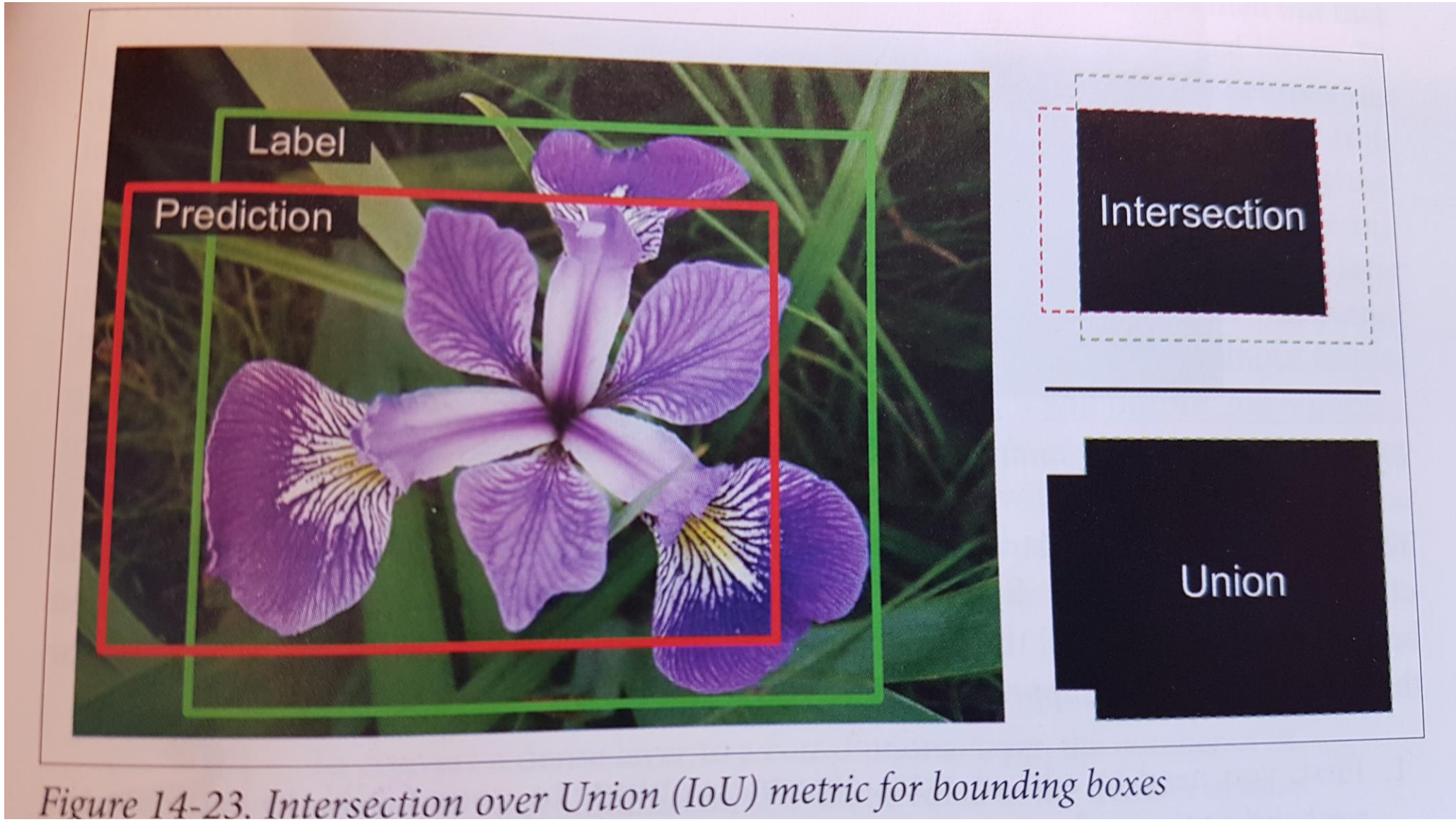

Figure 14-23. Intersection over Union (IoU) metric for bounding boxes

Object Detection

- How to find a specific object in an image? E.g. find “a traffic sign” in an image from your car camera

- Take a pretrained CNN (on the object(s) you are interested in) and apply it to small fragments of the image (“sliding window”), finding places where the CNN recognizes the desired object

$\Rightarrow$ multiple detections? Find the bounding box that maximizes the “objectness”/ “confidence” on a given fragment of the image

=> make sure that the boxes do not overlap

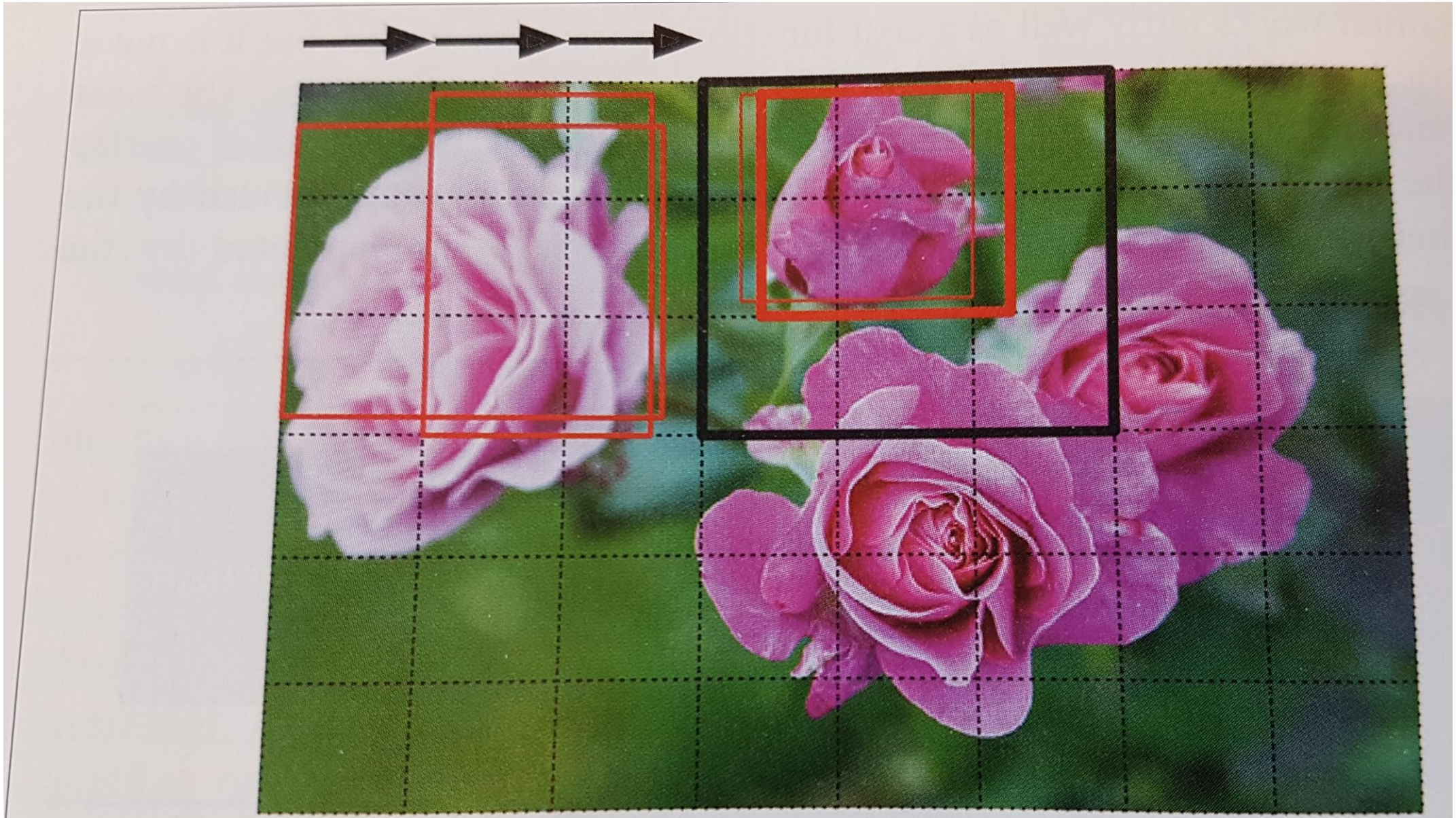

Figure 14-24. Detecting multiple objects by sliding a CNN across the image

Fully Convolutional Networks

- Main idea: the final dense layer can be viewed/specified as a convolutional layer with each filter looking at the whole “previous convolutional layer with *valid* padding”

- As the network becomes “fully convolutional” it can be applied to images of arbitrary (bigger) sizes!

- => much faster object detection
- => FCNs for Semantic Segmentation
- => FCN for Super Resolution

$\Rightarrow$ multiple detections? Find the bounding box that maximizes the “objectness”/ “confidence” on a given fragment of the image

=> make sure that the boxes do not overlap

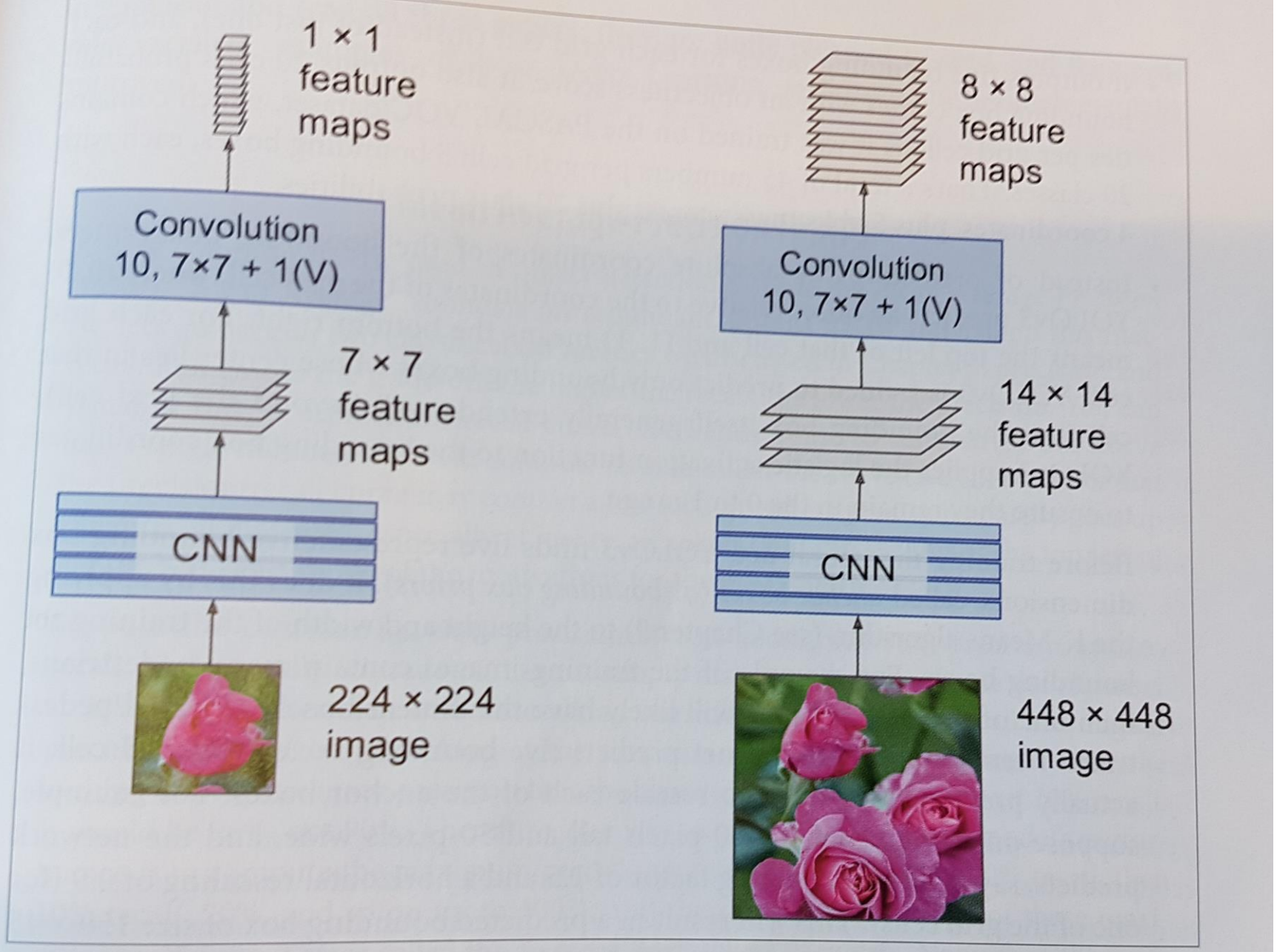

Figure 14-25. The same fully convolutional network processing a small image (left) and a large one (right)

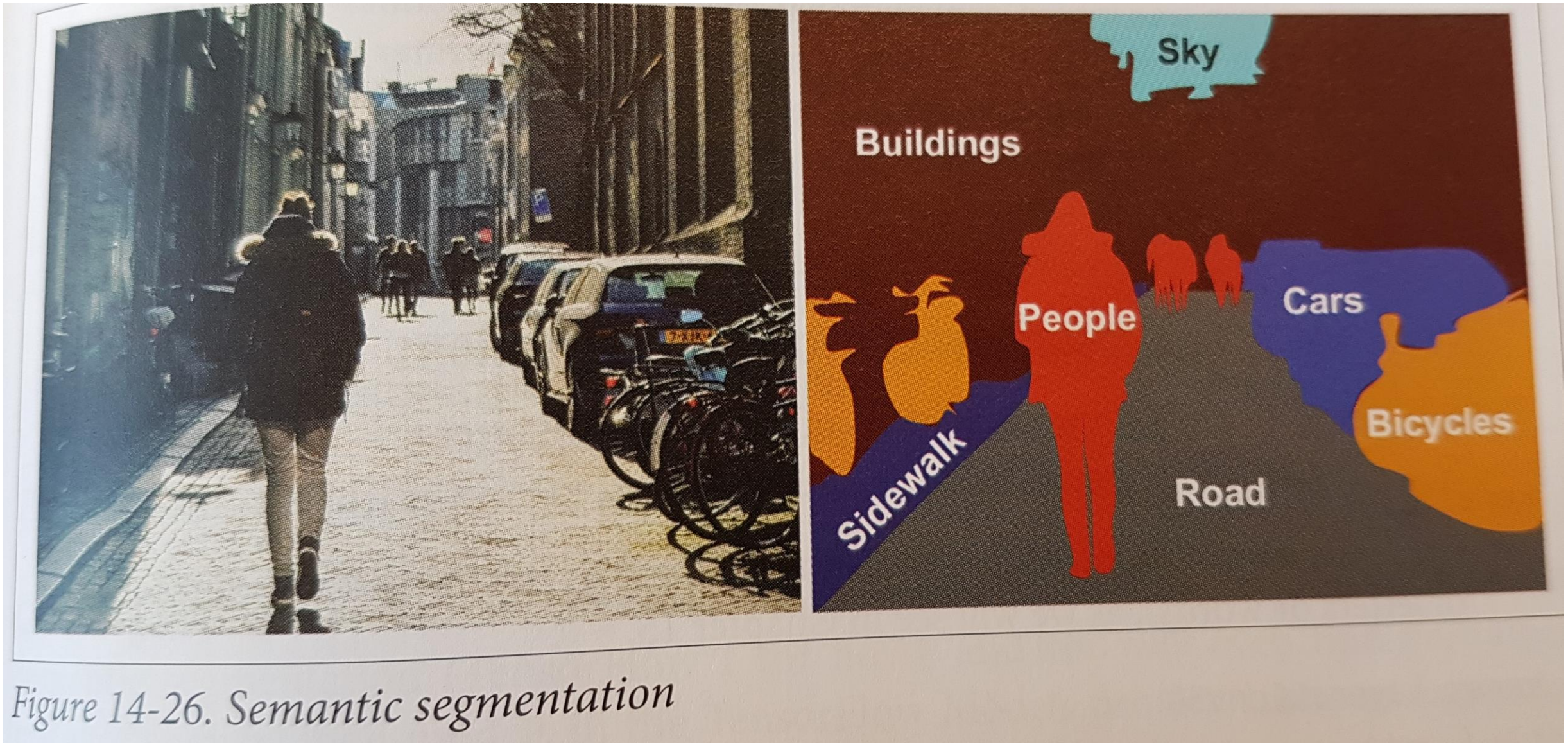

Figure 14-26. Semantic segmentation