Image Segmentation and Object Detection

Image Classification: more network architectures

- Efficientnet
- Efficientnetv2: architecture learned by another deep learning algorithm (2021)
- Vit: vision transformer (2021)

NLP: Natural language processing: analyze texts

Outline

1. Introduction
2. Image enhancement
3. Frequency domain operations
4. Image descriptors
5. Machine learning
6. Neural networks
7. **Segmentation and object detection**
8. Morphological processing
9. Geometric transformations
10. Motion analysis and optical flow
11. Compression
12. Other topics

Image Segmentation

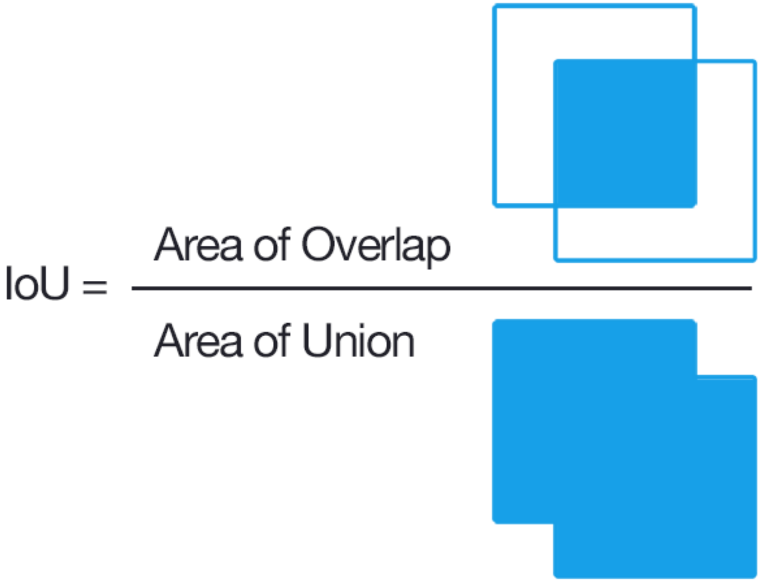
Image Segmentation

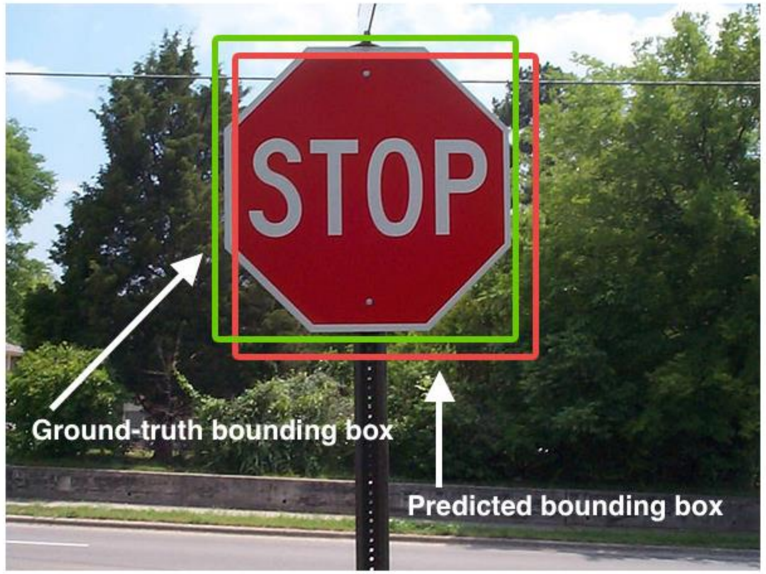
Segmentation Accuracy

- Pixel Accuracy: the number of pixels segmented correctly. This can be misleading. Assume that the background is 90% of a given image, then labelling the whole image is background will get 90% accuracy

Segmentation Accuracy

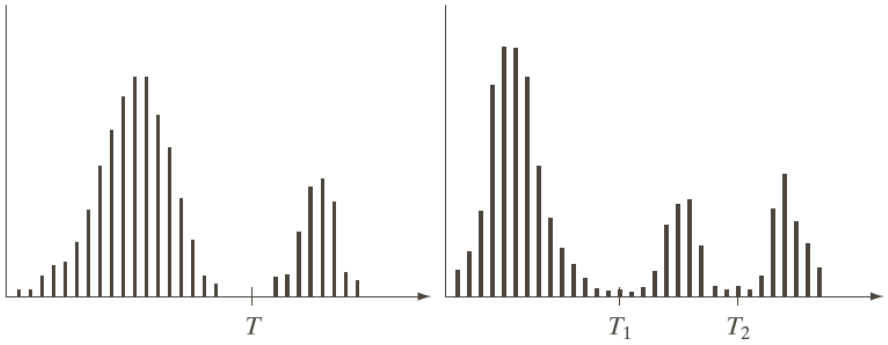
- IOU (intersection over union)





- If we have multiple objects, average IOU over the objects we have

Thresholding



a b

FIGURE 10.35 Intensity histograms that can be partitioned (a) by a single threshold, and (b) by dual thresholds.

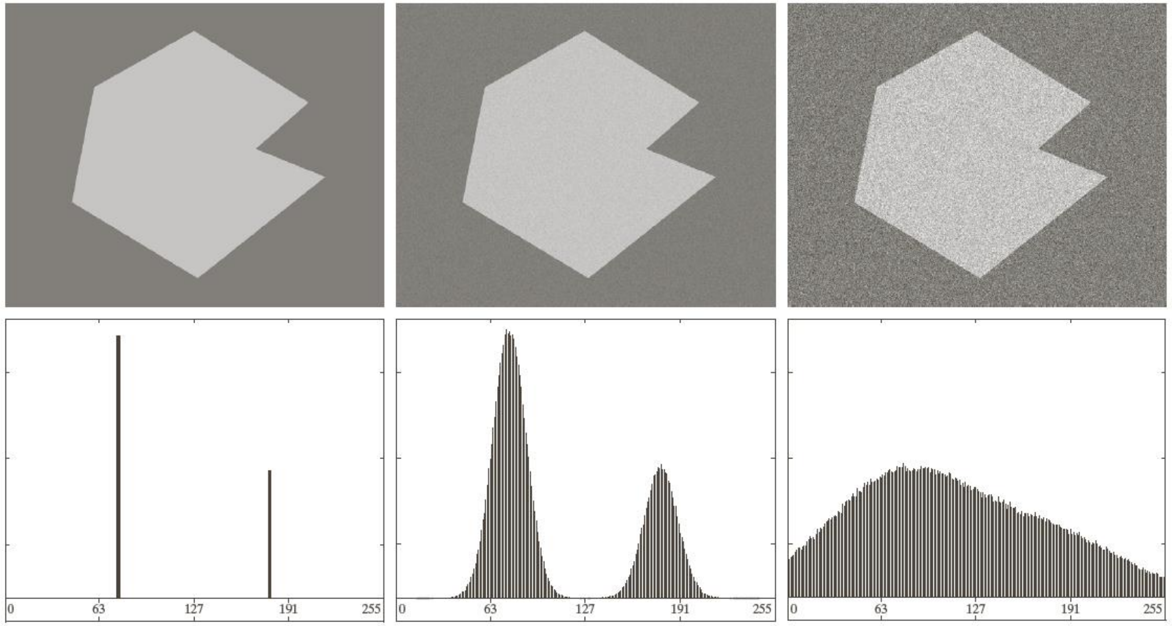
Thresholding histogram (a)

$$g(x,y) = \begin{cases} 1 \ if \ f(x,y) > T \\ 0 \ if \ f(x,y) < T \end{cases}$$

Thresholding histogram (b)

$$g(x,y) = \begin{cases} a & if \ f(x,y) > T_2 \\ b & if \ T_1 < f(x,y) < T_2 \\ c & if \ f(x,y) < T_1 \end{cases}$$

Noise in image thresholding



a b c
d e f

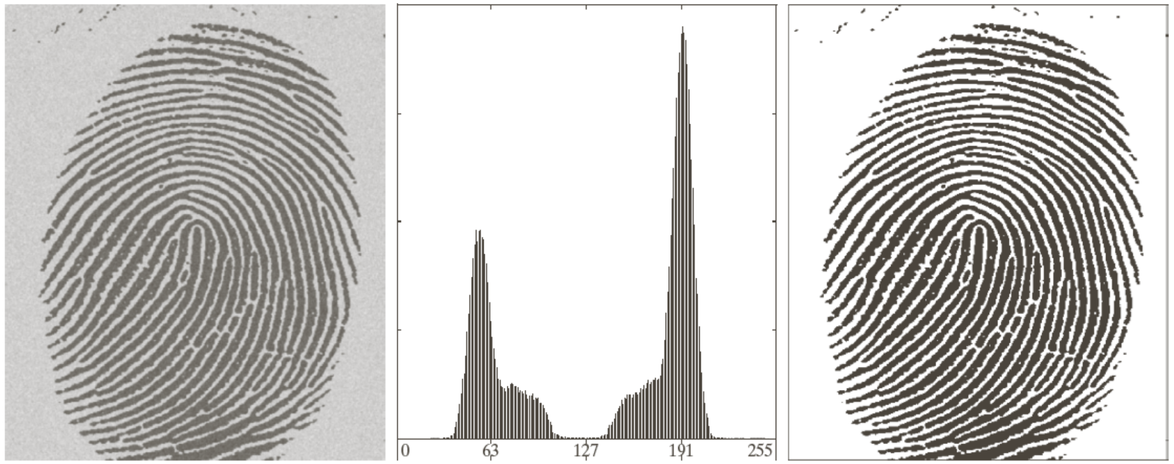
FIGURE 10.36 (a) Noiseless 8-bit image. (b) Image with additive Gaussian noise of mean 0 and standard deviation of 10 intensity levels. (c) Image with additive Gaussian noise of mean 0 and standard deviation of 50 intensity levels. (d)–(f) Corresponding histograms.

Basic Global Thresholding

- An algorithm for semi-automatic threshold selection.
- Assume one Threshold value *T* is going to separate the two objects we are interested in.

1. Select an estimate for the global threshold *T*
2. Segment the image using *T*
3. Compute the average intensity value m_1 and m_2 for pixels representing the two objects O_1 and O_2
4. Compute a new threshold value $T = \frac{1}{2}(m_1 + m_2)$
5. Repeat steps (2) through (4) until convergence.

Basic Global Thresholding



a b c

FIGURE 10.38 (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (the border was added for clarity). (Original courtesy of the National Institute of Standards and Technology.)

Otsu’s method

- A fast algorithm used for binary thresholding
- Idea: Best threshold is the value that achieves maximum separation between classes.
- Maximize the variance between the two classes.

Otsu’s method

For *every* possible *Threshold*:

A. Calculate between-class variances:

   1. probability of being in group 1; probability of being in group 2
   2. determine average intensity of group 1; determine average intensity of group 2
   3. calculate average intensity for the entire image.
   4. calculate between-class variance

B. Report which *Threshold* gave rise to maximum.

Otsu's method

For *every* possible *Threshold*:

A. Calculate between-class variances:

1. probability of being in group 1 $q_1(T)$; probability of being in group 2 $q_2(T)$
2. determine average intensity of group 1 $m_1(T)$; determine average intensity of group 2 $m_2(T)$
3. calculate average intensity for the entire image m_G.
4. calculate between-class variance, defined as

$$\sigma_B^2(T) = q_1(T)(m_1(T) - m_G)^2 + q_2(T)(m_2(T) - m_G)^2$$

Probability of being in group 1

Average intensity of the entire image

Otsu’s method

For *every* possible *Threshold*:

A. Calculate between-class variances:
   1. probability of being in group 1; probability of being in group 2
   2. determine average intensity of group 1; determine average intensity of group 2
   3. calculate average intensity for the entire image.
   4. calculate between-class variance, defined as

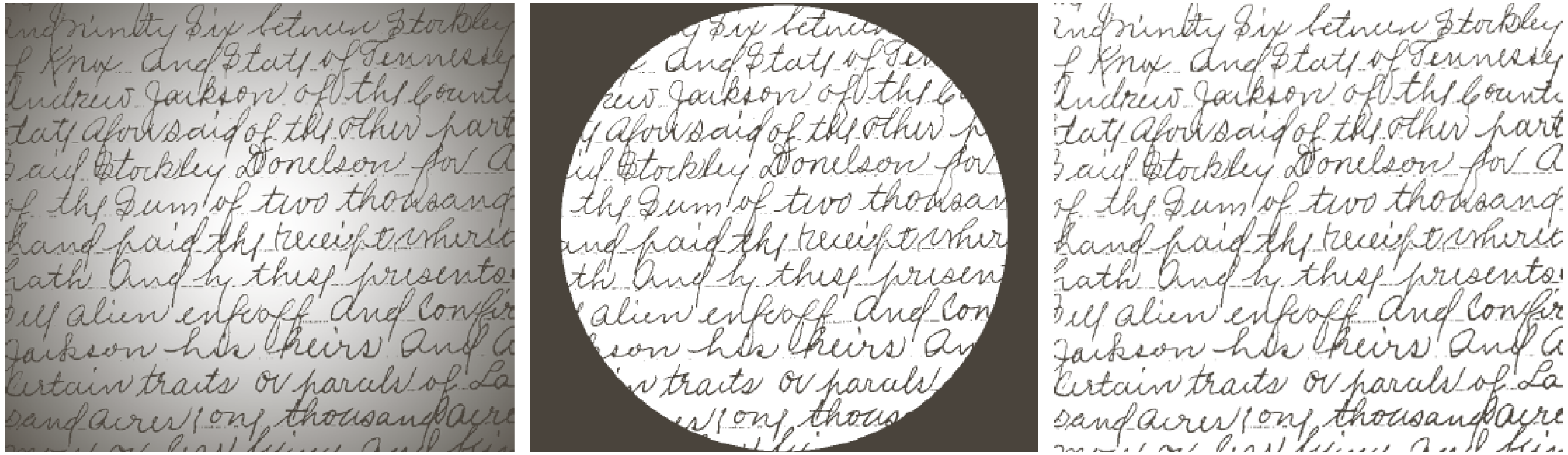
$$\sigma_B^2(T) = q_1(T)\left(m_1(T) - m_G\right)^2 + q_2(T)\left(m_2(T) - m_G\right)^2$$

Report which *Threshold* gave rise to maximum.

$$\arg\max \left\{ \sigma_B^2(T) \mid 0 \le T \le L-1 \right\}$$

Otsu’s method

- Global thresholding will not work in cases where the statistics of image intensities vary sharply.
- Solution: Use local thresholding



a b c

FIGURE 10.49 (a) Text image corrupted by spot shading. (b) Result of global thresholding using Otsu’s method. (c) Result of local thresholding using moving averages.

Otsu’s method

a b c

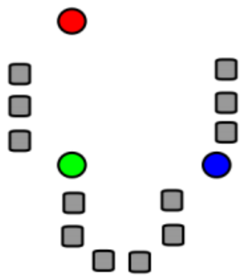
FIGURE 10.50 (a) Text image corrupted by sinusoidal shading. (b) Result of global thresholding using Otsu’s method. (c) Result of local thresholding using moving averages.

Clustering

- Group together similar points and represent them with a single index
- Applications
  - Summarizing data
  - Counting
  - Segmentation
- Methods
  - K-means
  - Mean-shift
  - Deep learning methods

K-means clustering

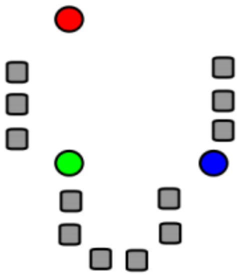
First the number of clusters should be specified. In this case n=3. Start by selecting 3 random centroids



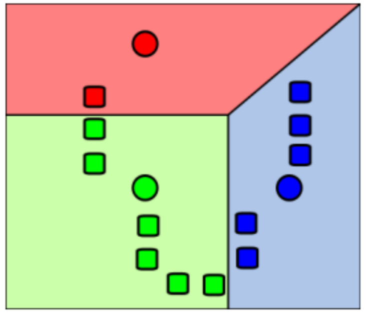
1. Select initial
centroids at random

K-means clustering

Euclidian distance
d= sqrt [(p1-q1)^2 + (p2-q2)^2]

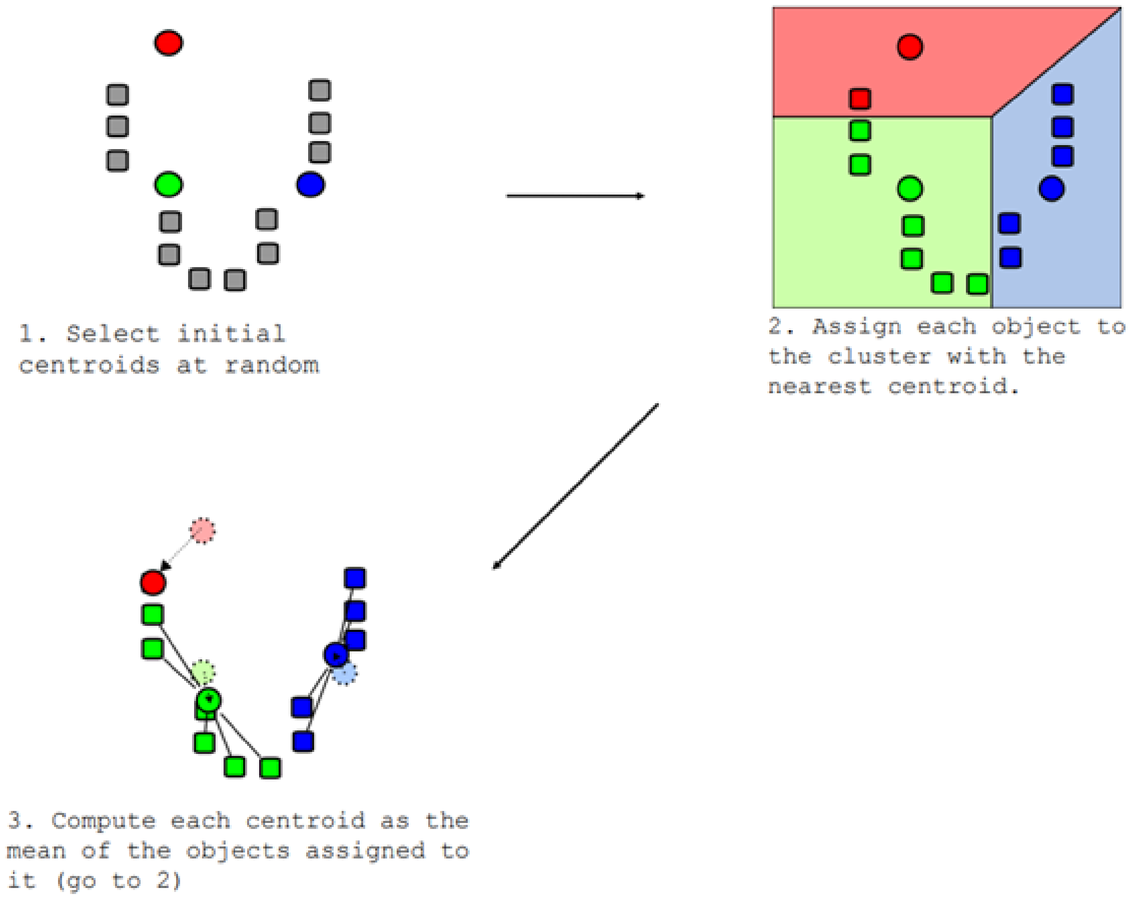


1. Select initial centroids at random

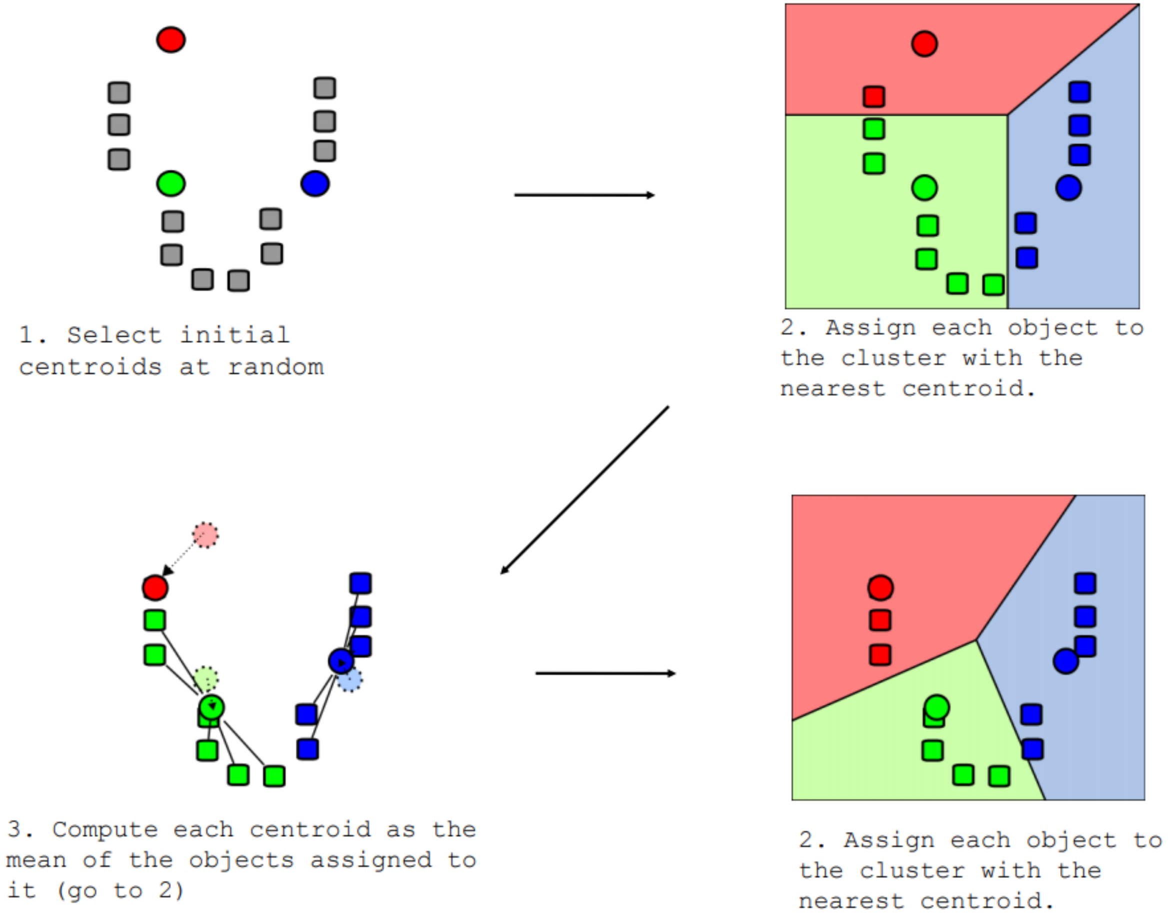


2. Assign each object to the cluster with the nearest centroid.

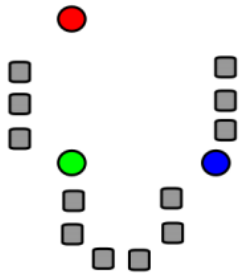
K-means clustering



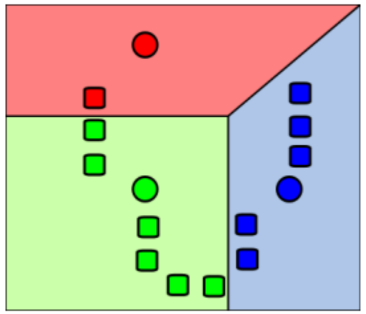
K-means clustering



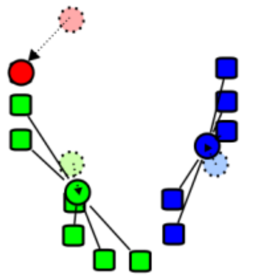
K-means clustering



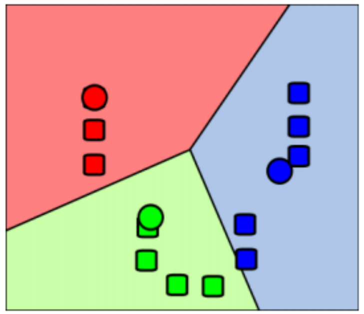
1. Select initial centroids at random



2. Assign each object to the cluster with the nearest centroid.



3. Compute each centroid as the mean of the objects assigned to it (go to 2)



2. Assign each object to the cluster with the nearest centroid.

Repeat previous 2 steps until no change

K-means clustering

K-means clustering

- Pros
  - Simple, fast, and easy to implement
- Cons
  - Require the number of clusters to be given
  - Sensitive to outliers
  - Can be slow

Region growing

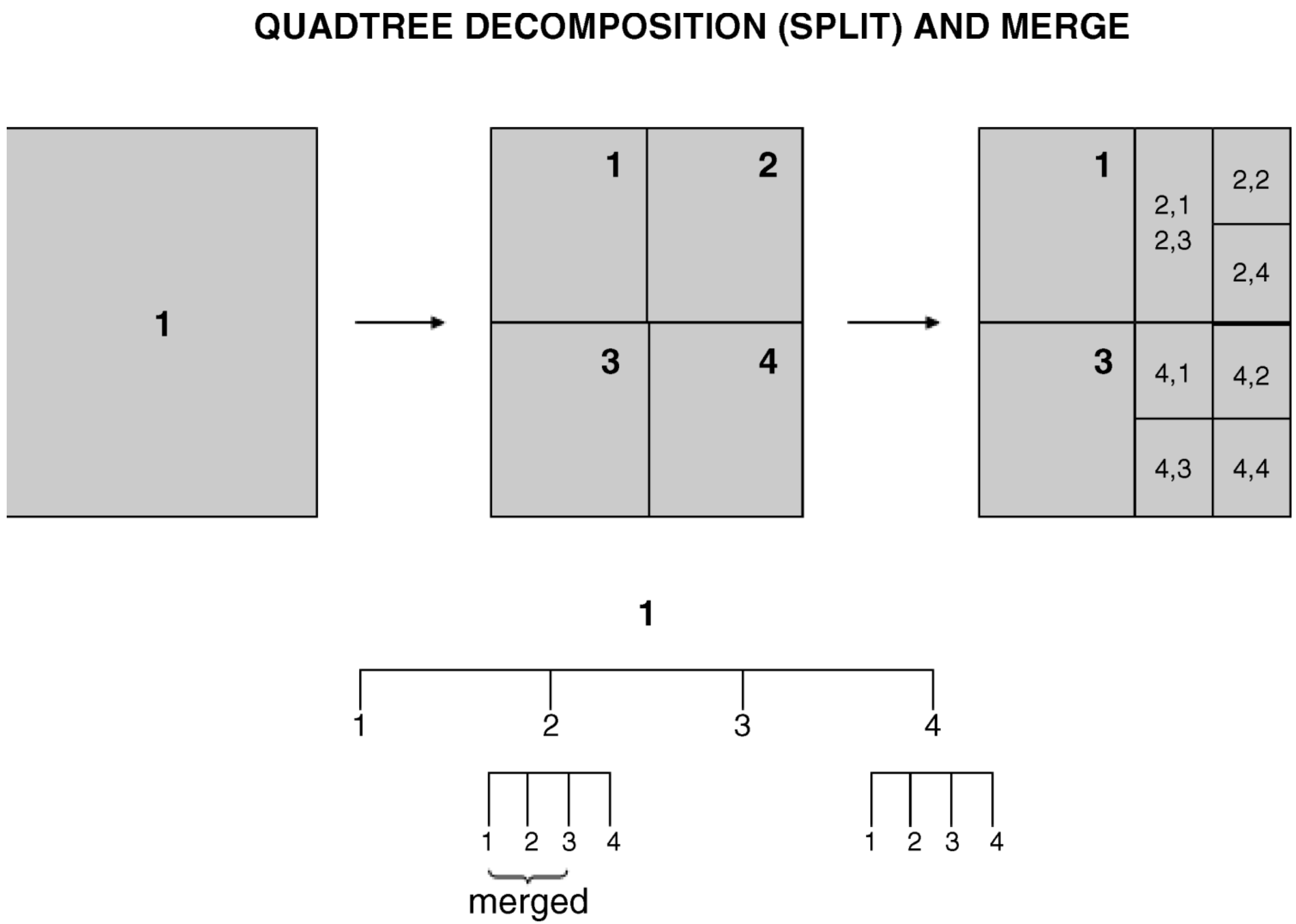
- A well-known segmentation technique.
- Start with a seed pixel and append pixels to it if they satisfy a similarity criteria.
- Examples of similarity criteria
  - ➢ Absolute intensity difference between a candidate pixel and the seed pixel must be within a specified range.
  - ➢ Absolute intensity difference between a candidate pixel and the running average intensity of the growing region must be within a specified range.
- Region growth video

https://youtu.be/SokLiR6PA0I

Split-and-merge algorithm

- Applies an approach that is opposite to region growth.
- Consider the image as a whole to the initial segment/area of interest.
- Do all pixels in the region satisfy a similarity condition
- If TRUE the area of interest corresponds to a single region in the image is labelled as such.
- If FALSE, then split the area of interest to smaller areas, and consider each of the sub-areas as the area of interest.
- Next, each region gets assigned the mean intensity value
- Finally, merge similar regions according to a designated criteria

Split-and-merge algorithm



Split-and-merge algorithm

- Default similarity function used: Split based on abs(max(i) – min(i))

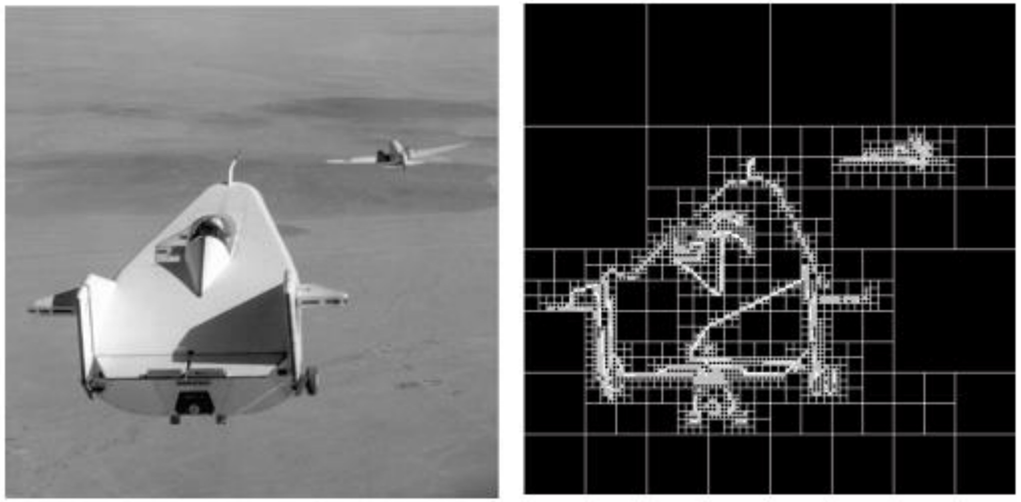


Image courtesy of NASA

Split-and-merge algorithm

Original image

(https://vgg.fiit.stuba.sk/2016-06/split-and-merge/)

Split-and-merge algorithm

After splitting

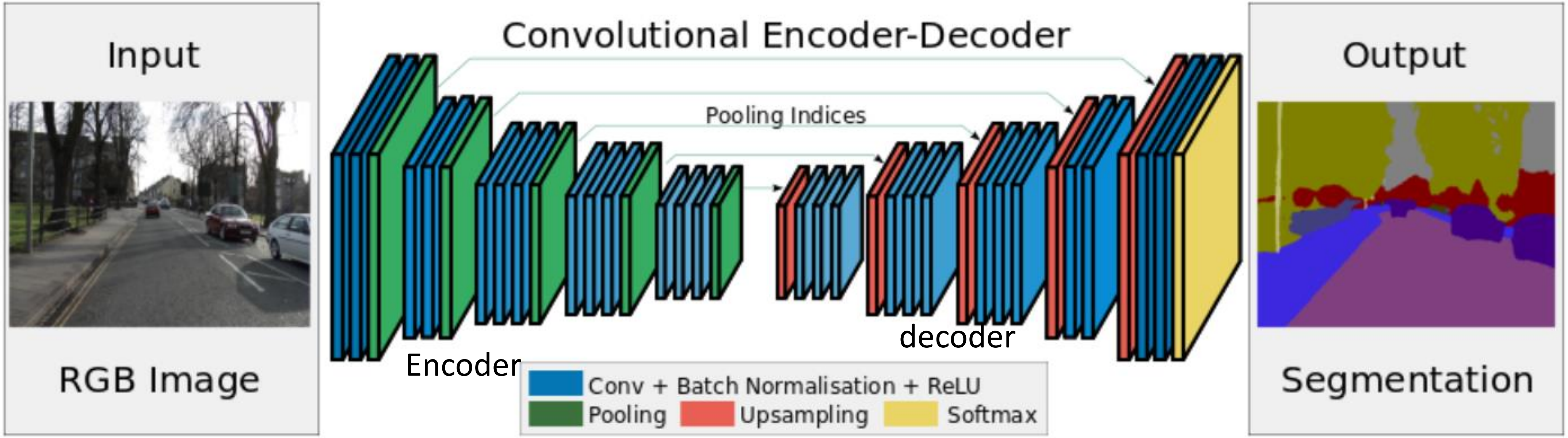
Split-and-merge algorithm

After merging

Split-and-merge algorithm

Final result where the boundaries are smoothed using morphological operations (sequence of erosion and dilations)

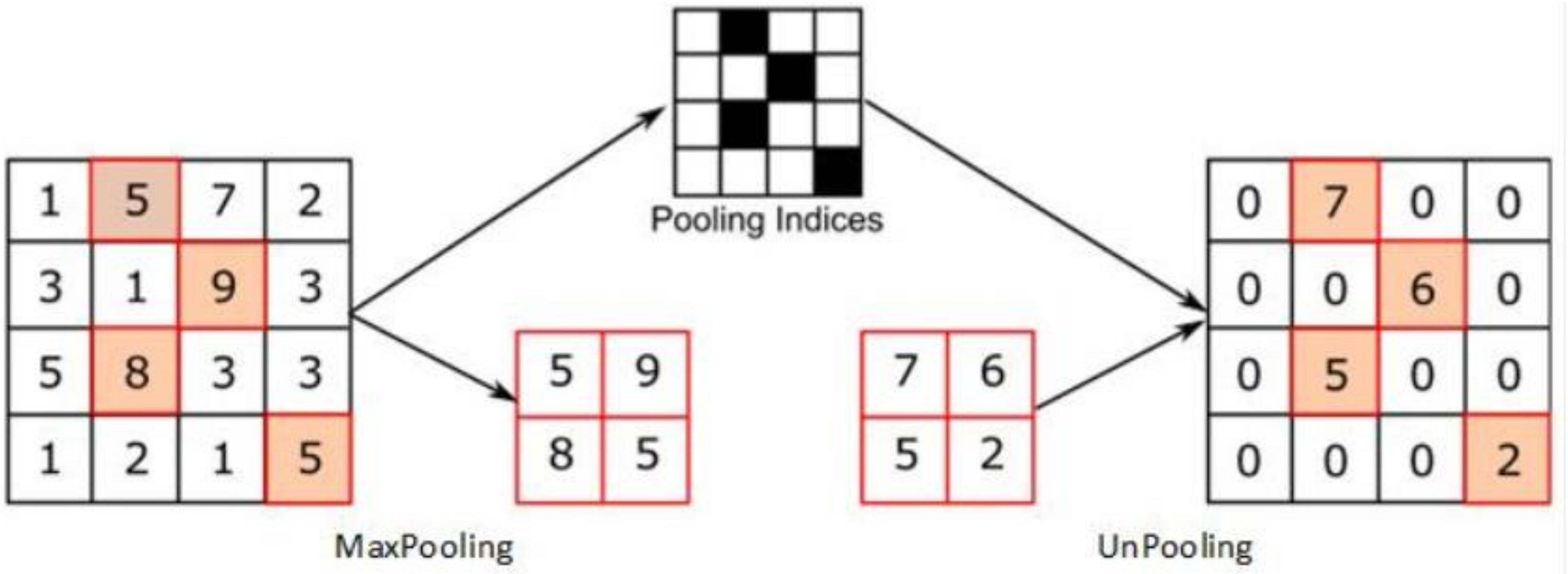
Deep Learning Methods
SegNet



SegNet, 2017

Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation." IEEE transactions on pattern analysis and machine intelligence 39.12 (2017): 2481-2495.

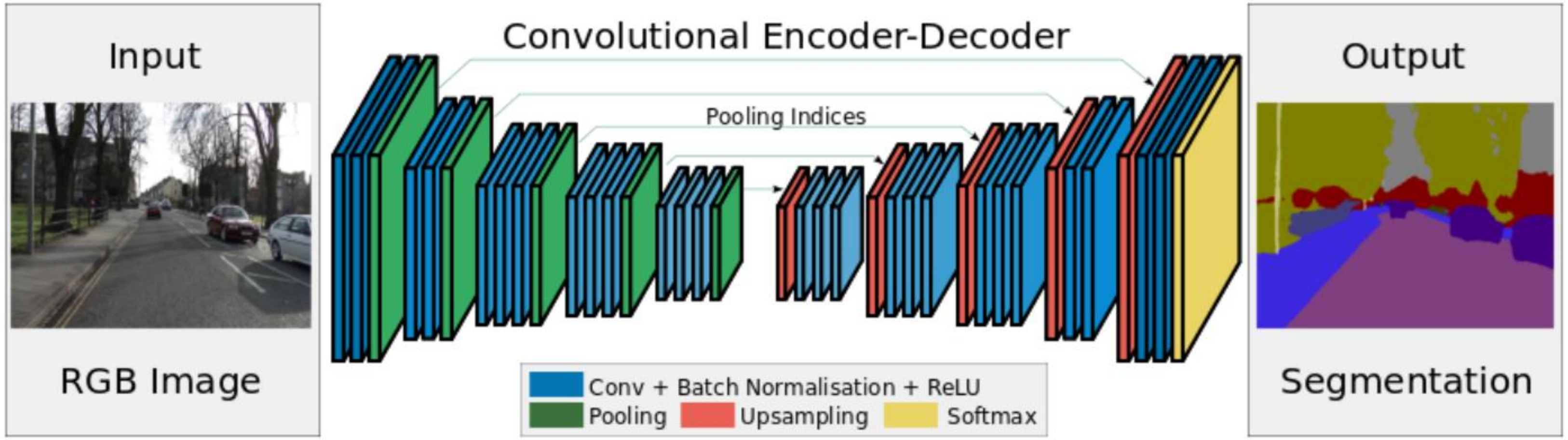
Unpooling Layer



Max pooling reduces the size of the matrix, while the unpooling operation restores it

SegNet

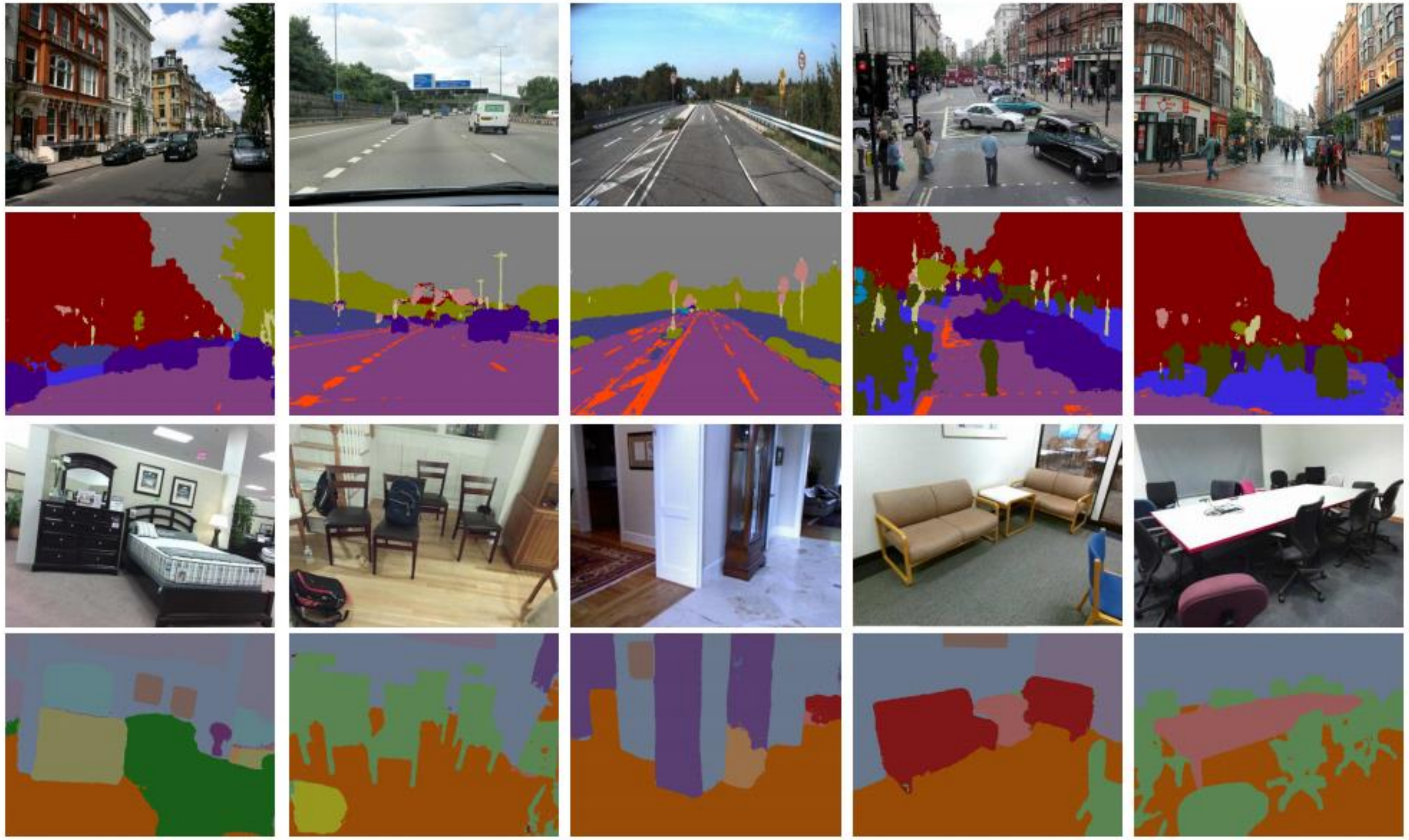
Abstract—We present a novel and practical deep fully convolutional neural network architecture for semantic pixel-wise segmentation termed SegNet. This core trainable segmentation engine consists of an encoder network, a corresponding decoder network followed by a pixel-wise classification layer. The architecture of the encoder network is topologically identical to the 13 convolutional layers in the VGG16 network [1]. The role of the decoder network is to map the low resolution encoder feature maps to full input resolution feature maps for pixel-wise classification. The novelty of SegNet lies is in the manner in which the decoder upsamples its lower resolution input feature map(s). Specifically, the decoder uses pooling indices computed in the max-pooling step of the corresponding encoder to perform non-linear upsampling. This eliminates the need for learning to upsample. The upsampled maps are sparse and are then convolved with trainable filters to produce dense feature maps. We compare our proposed architecture with the widely adopted FCN [2] and also with the well known DeepLab-LargeFOV [3], DeconvNet [4] architectures. This comparison reveals the memory versus accuracy trade-off involved in achieving good segmentation performance.



SegNet

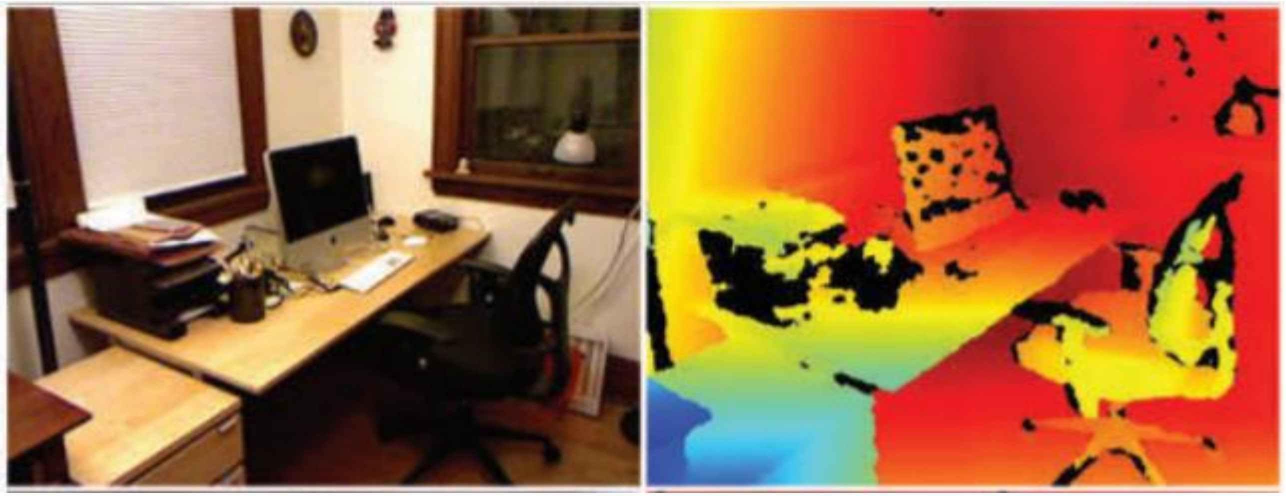
Road scenes
dataset

RGB-D
RGB+depth



RGB-D

- RGB images + depth information
- Depth information allows one have an estimate of the depth of objects or at least a relative comparison of the location of different objects.



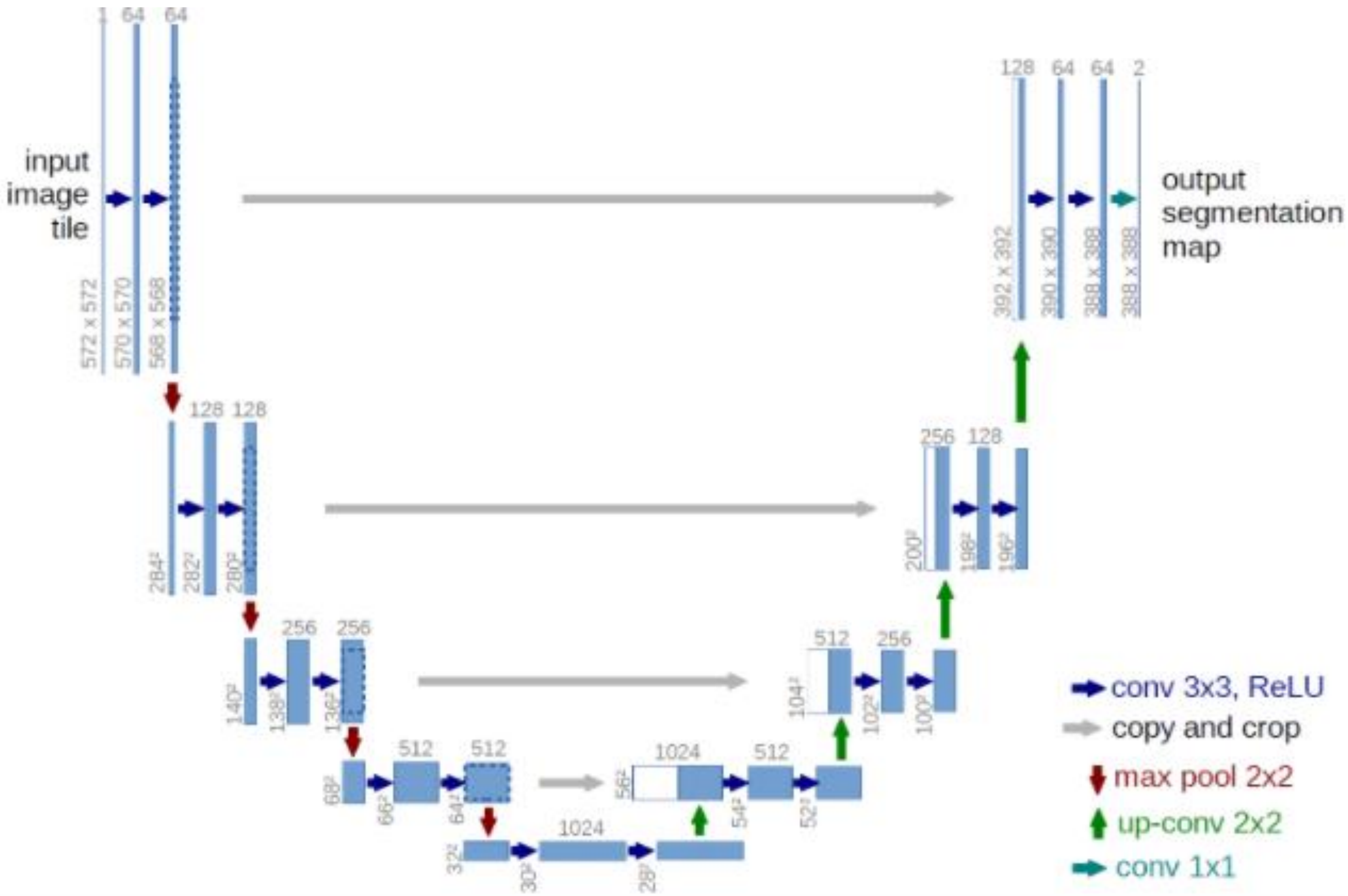
Examples: https://www.youtube.com/watch?v=8CTJL5lUjHg

Datasets

KITTI

Cityscapes

U-Net Architectures

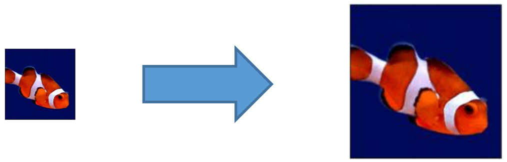


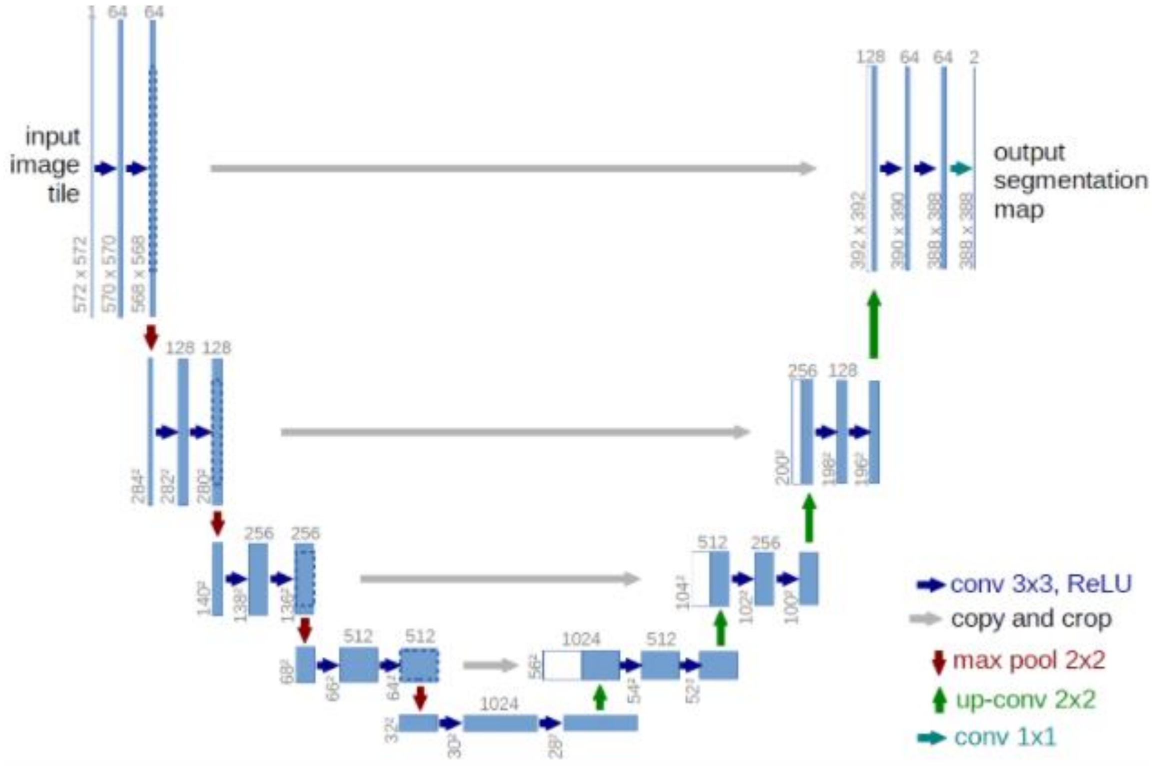
https://link.springer.com/content/pdf/10.1007%2F978-3-319-24574-4_28.pdf

Up-conv layer

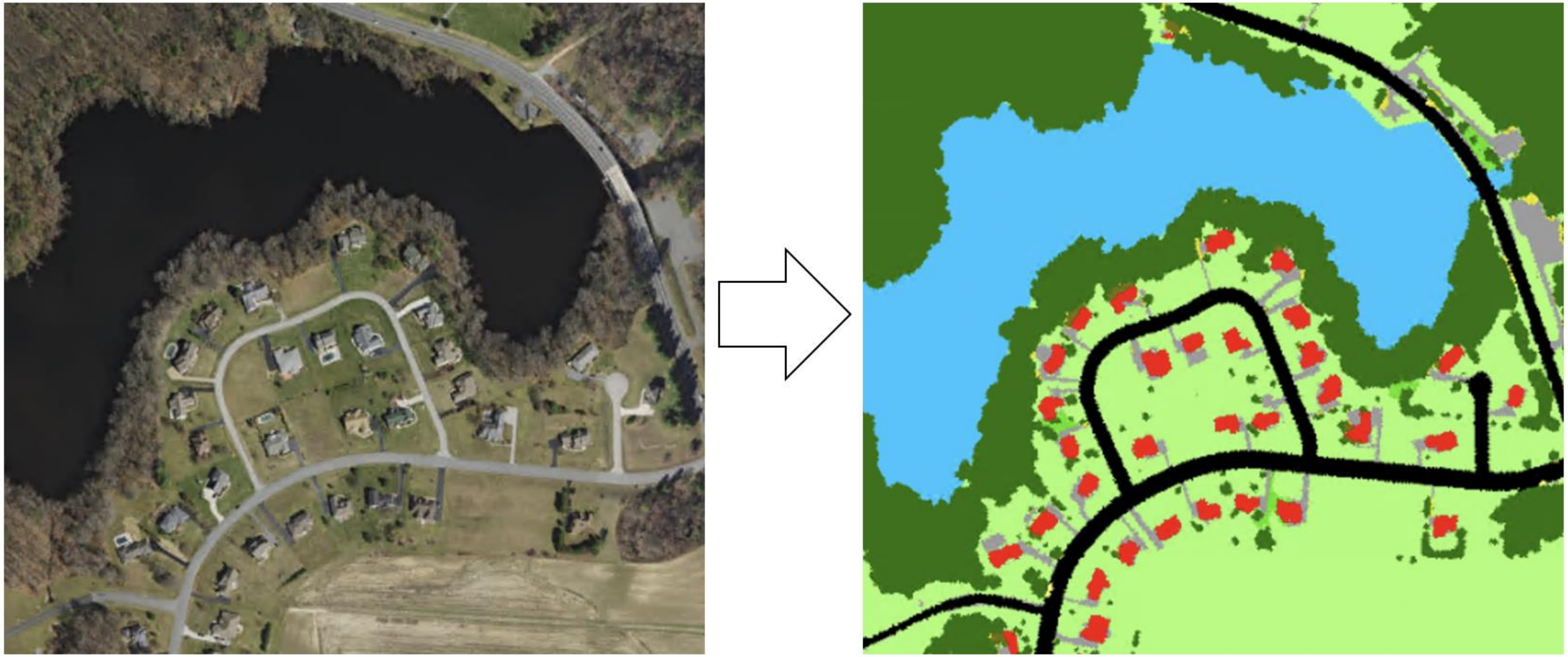
- U-Net: Uses up-conv layers.
- An up-conv layer: upsample the input and then applies a regular convolution
- Upsampling: increase image size

This can be achieved by replication or interpolation methods (averaging nearby values)





Examples



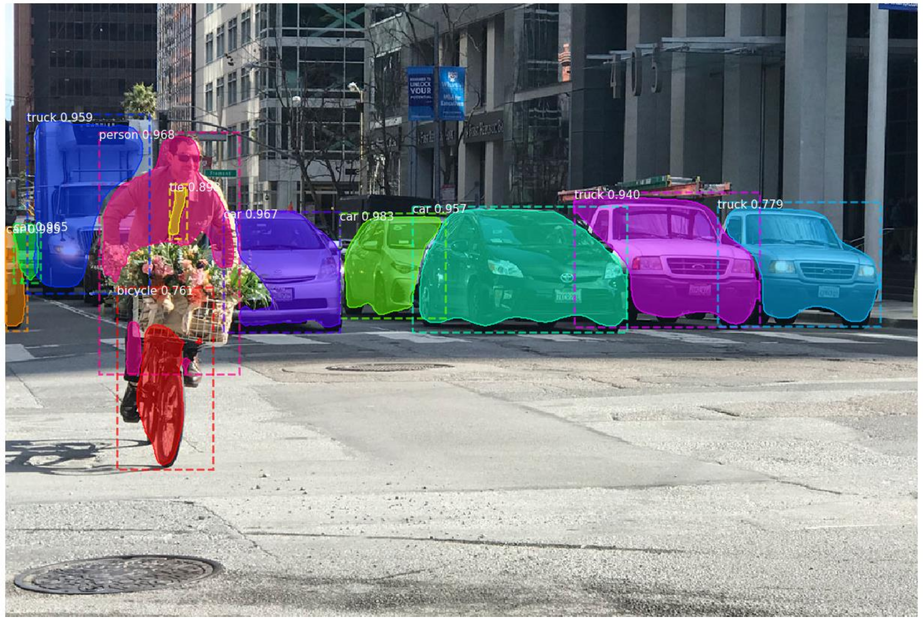
Types of Image Segmentation

Semantic segmentation

Instance segmentation

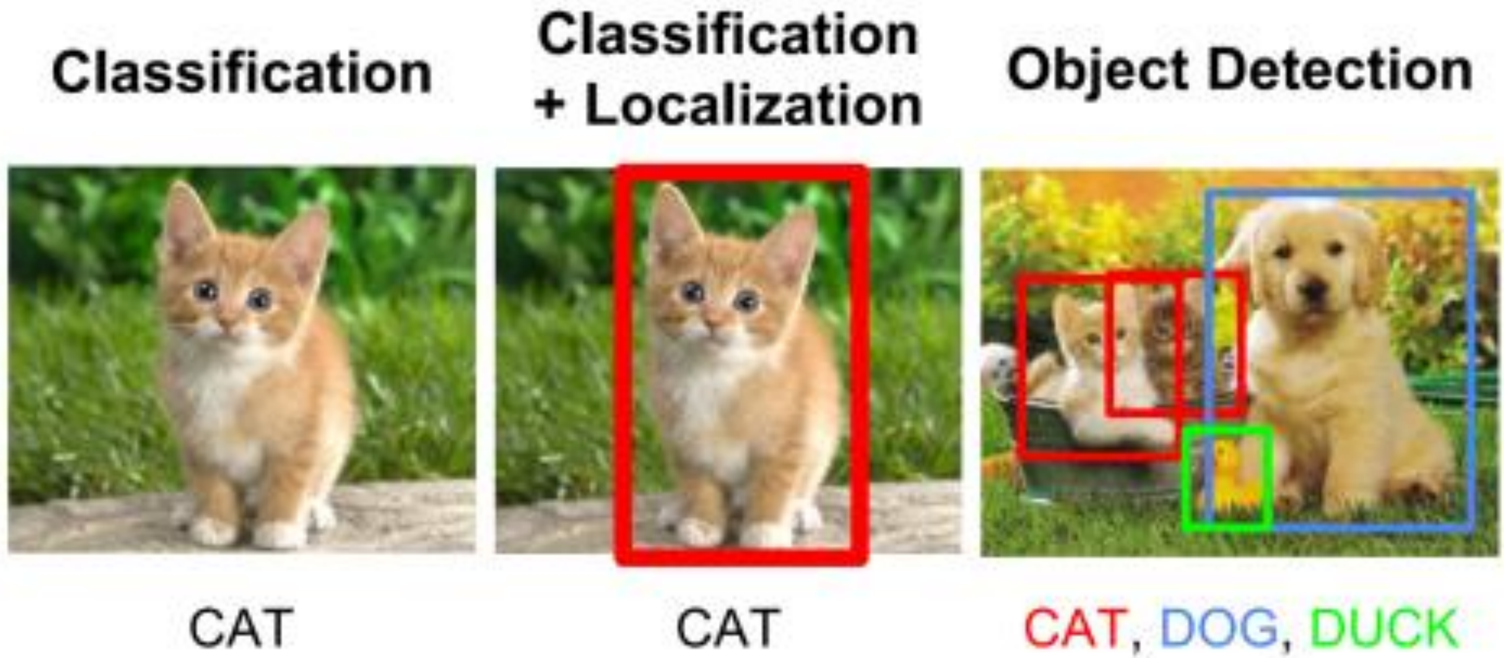
Segmentation using deep learning

- Very high accuracy compared to traditional methods
- Large cost of labelling data

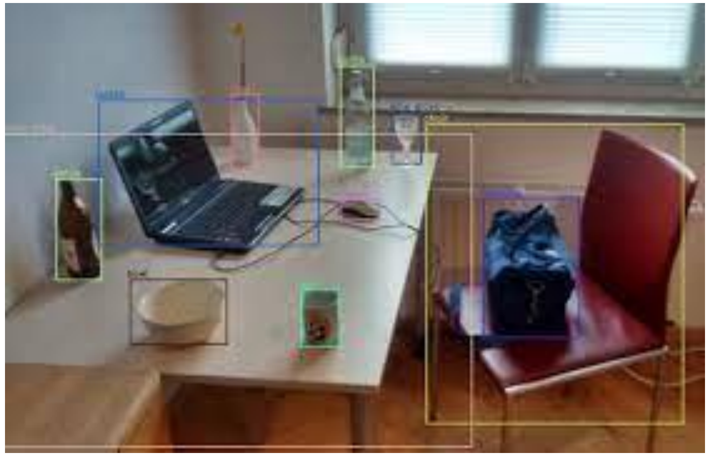


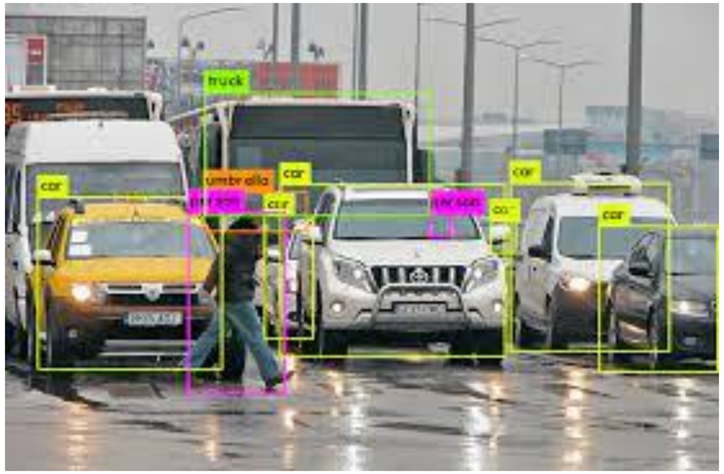
Object Detection

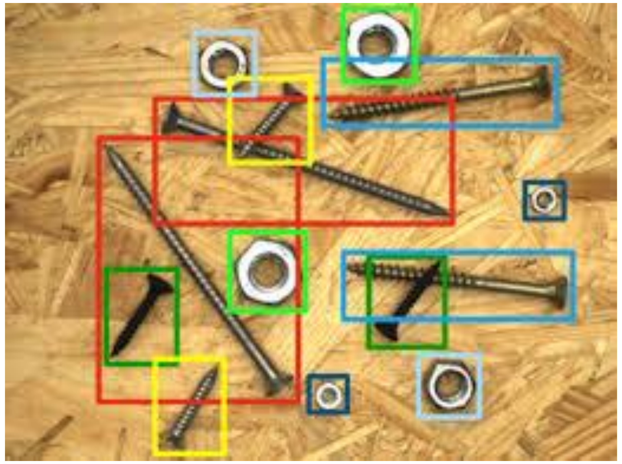
Object detection



Object detection



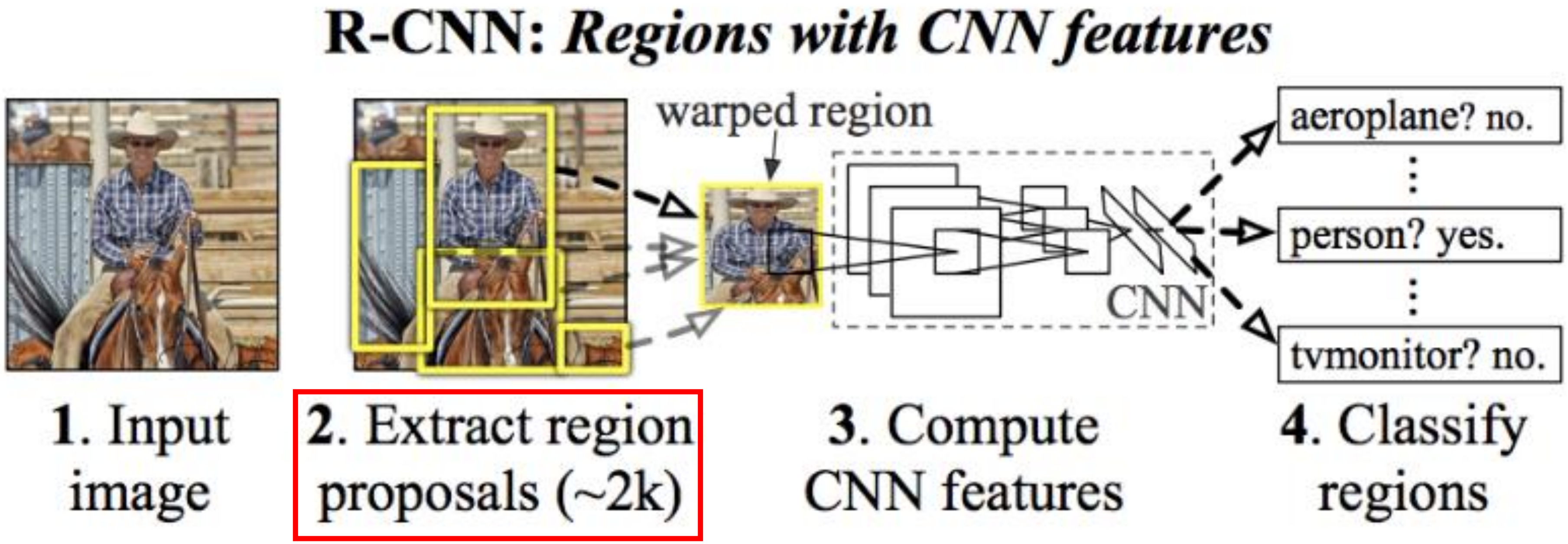




Loss functions

- Labels to be predicted:
  - Class label of each object
  - Bounding box coordinates, for example:
    $\{b_x, b_y, h_x, h_y\}$: center and dimensions of the box
- Mean squared error (MSE) is used as the loss function for the bounding box
- Like image classification problems, cross entropy is used as the loss for class labels

R-CNN



Softmax over two classes
Or
SVM

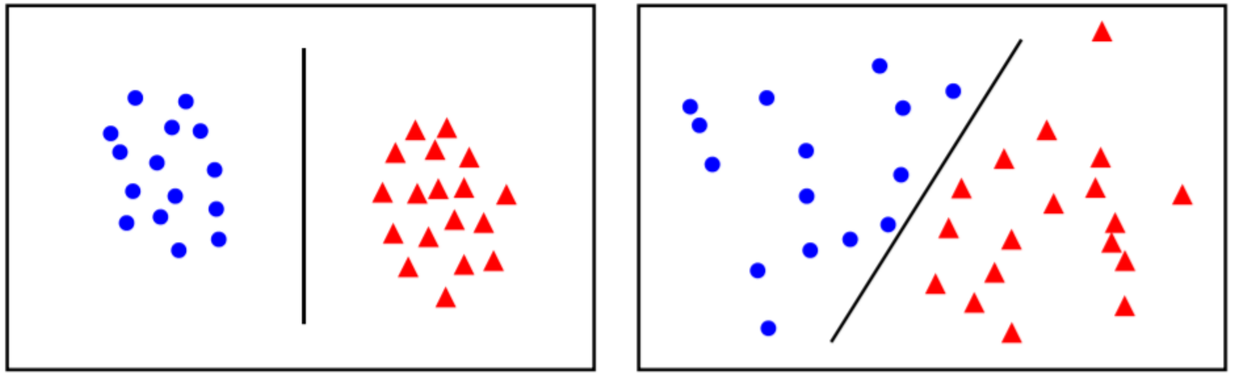
Support Vector Machines (SVM)

Given training data $(\mathbf{x}_i, y_i)$ for $i = 1 \ldots N$, with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$, learn a classifier $f(\mathbf{x})$ such that

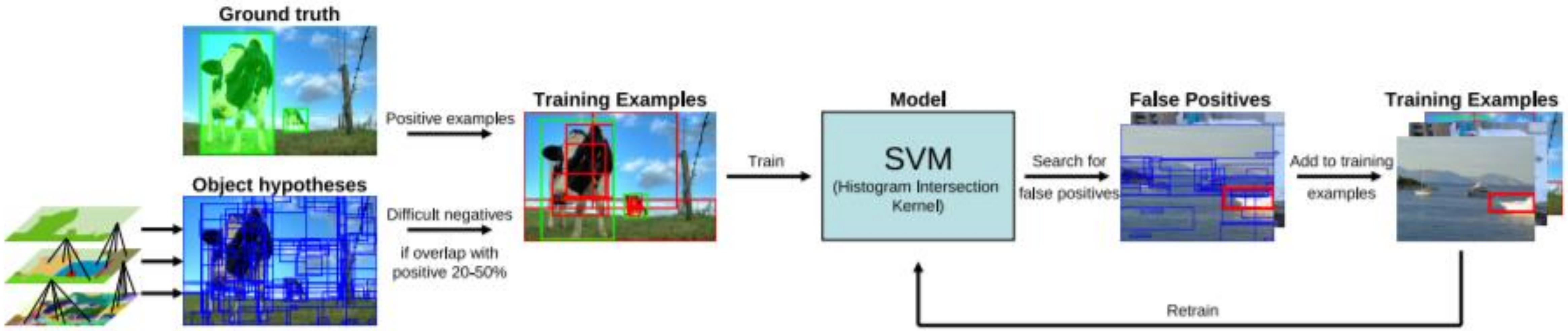
$$f(\mathbf{x}_i) \begin{cases} \geq 0 & y_i = +1 \\ < 0 & y_i = -1 \end{cases}$$

i.e. $y_i f(\mathbf{x}_i) > 0$ for a correct classification.

linearly separable

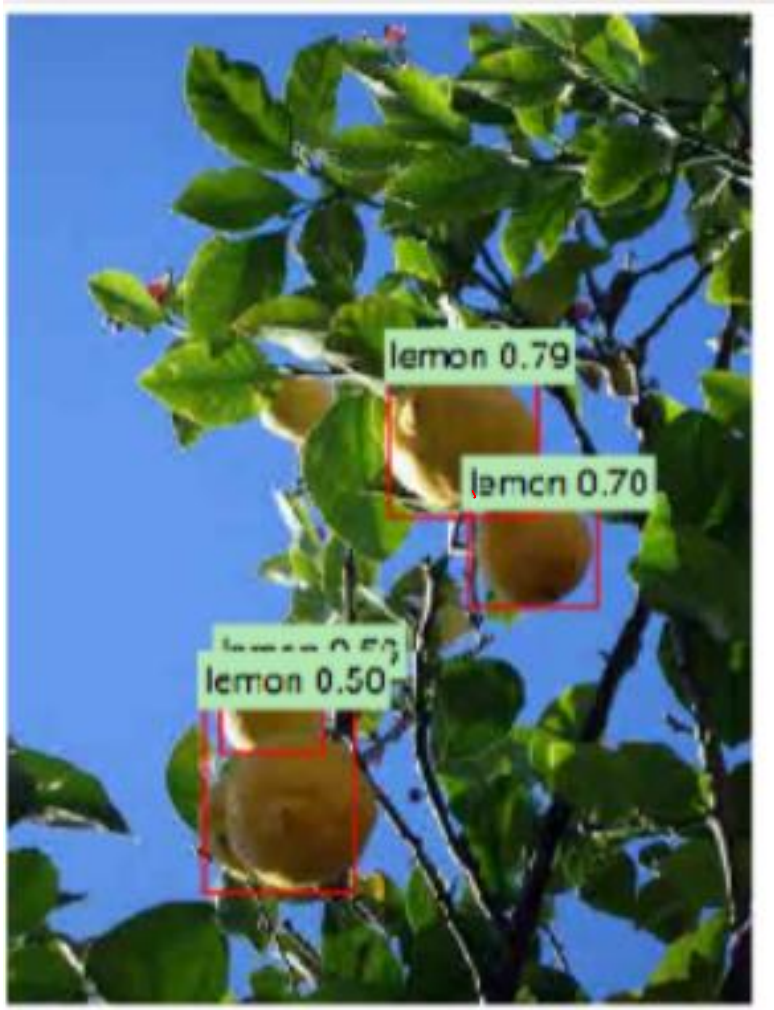


R-CNN: Extracting ROIs

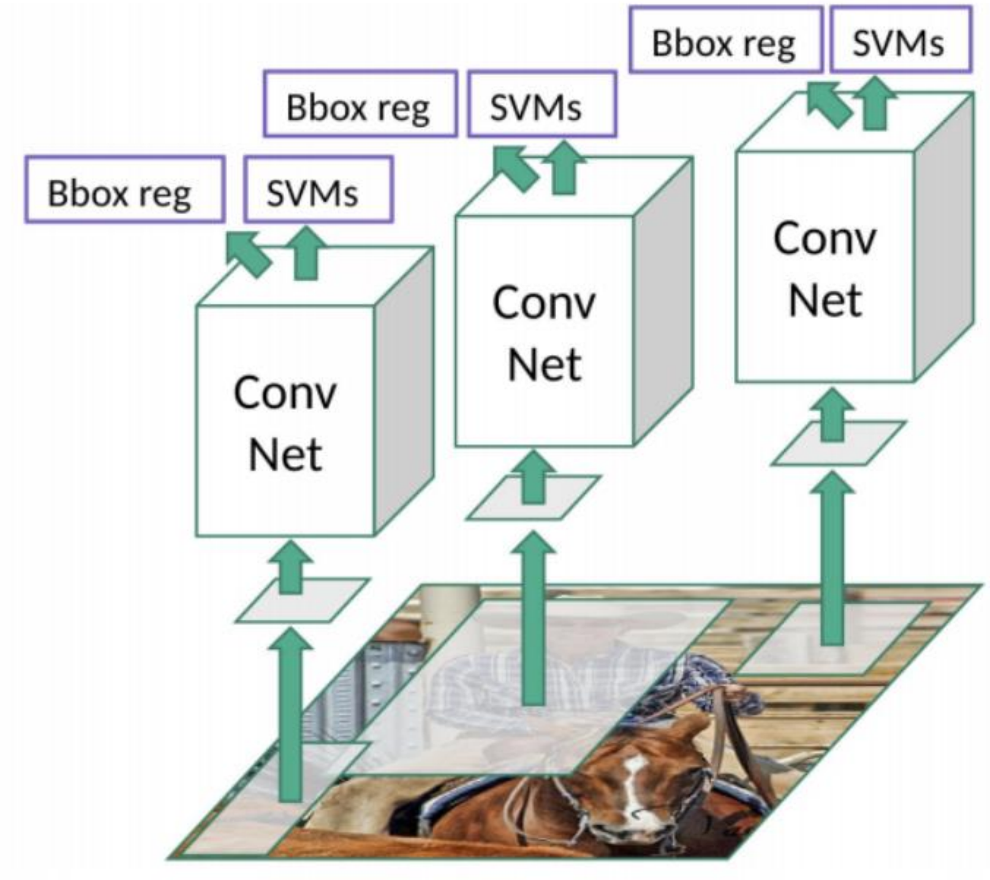


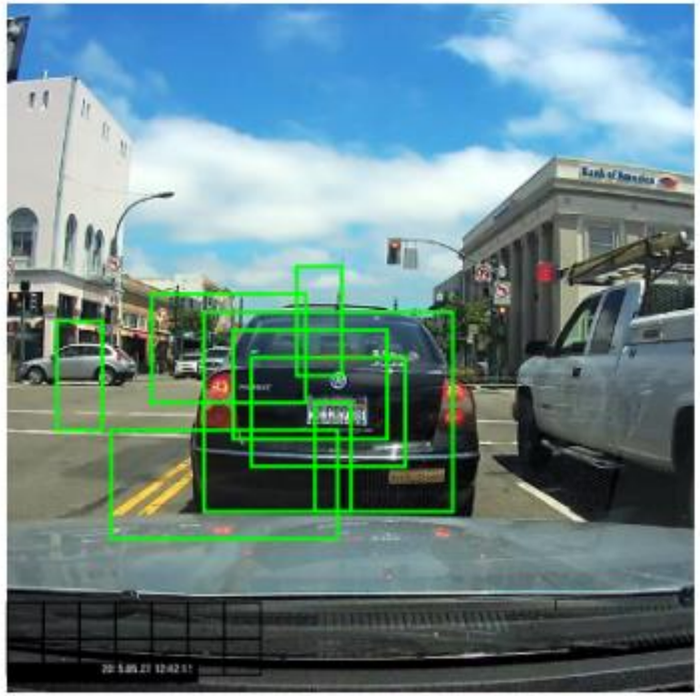
R-CNN

- Network also outputs confidence scores (using SoftMax probability or in the case of SVM computed using distance the margin line)



R-CNN: Improving initial ROIs

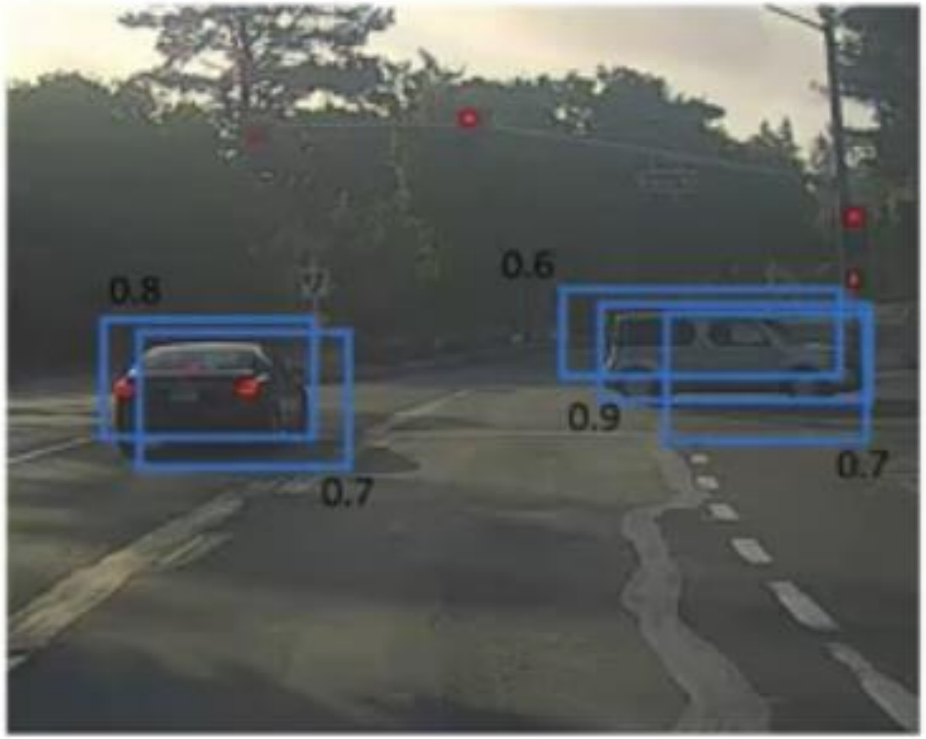




R-CNN: Improving initial ROIs

- Non-maximum Suppression: Reject regions that have a high intersection-over-union (IoU) overlap with a higher scoring selected region

R-CNN: Non-maximum Suppression



Select the highest scoring rectangles, and reject other rectangles with high overlap (IOU) with the selected rectangles. In this example the 0.9 rectangle on the right car and the 0.8 rectangle on the black car will remain after non-max suppression

Improvements over R-CNN

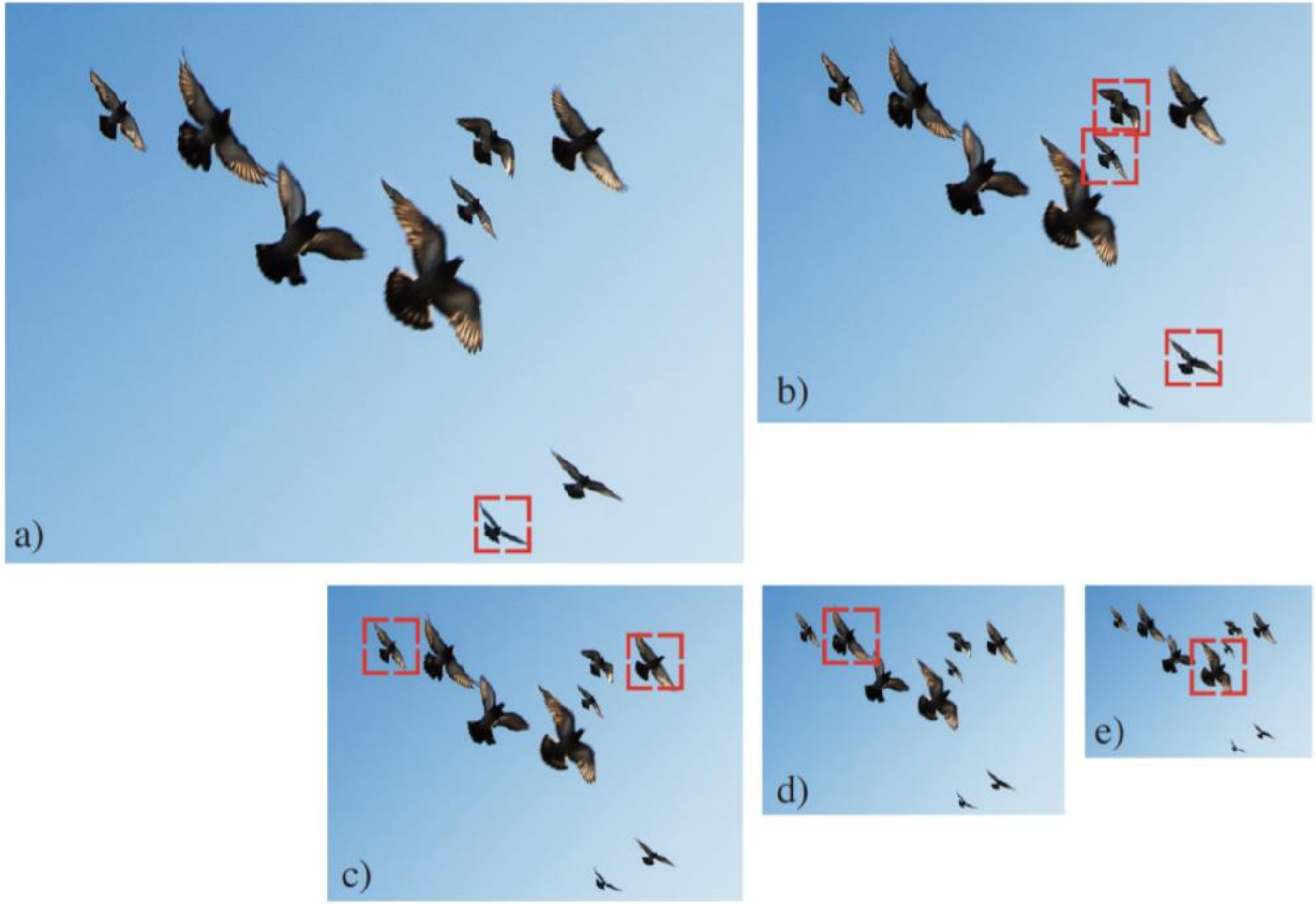
R-CNN takes a long time and require two separate operations (regions proposals, and bounding box regression/classification)

- Fast R-CNN
- Faster R-CNN
- Yolo (you only look once)

Image Pyramids

Image Pyramids

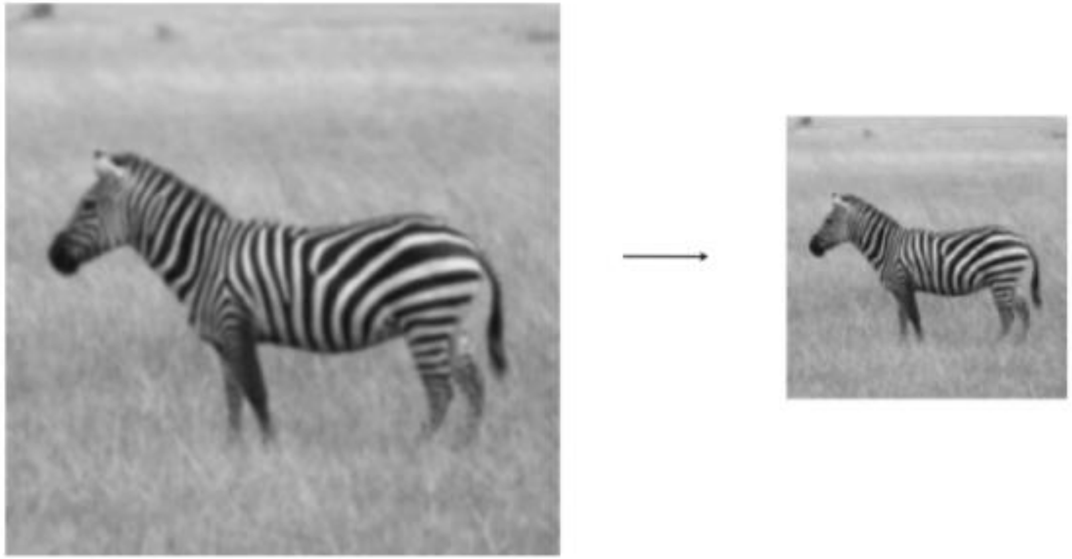
Object detection across multiple scales



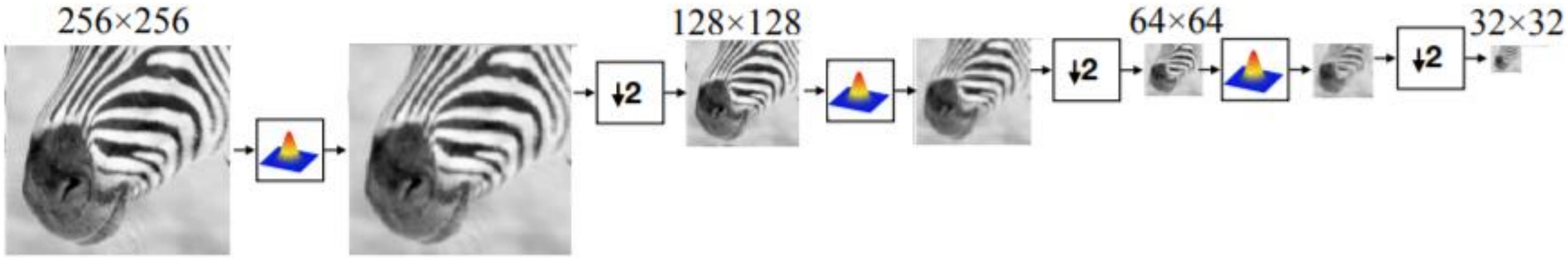
The Gaussian Pyramid

For each level

1. Blur input image with a Gaussian filter
2. Downsample image



The Gaussian Pyramid



Downsampling artifacts

Original image

Downsampled
½ each dimension

Smoothed (blurred) then
downsampled

Downsampling artifacts

Original image

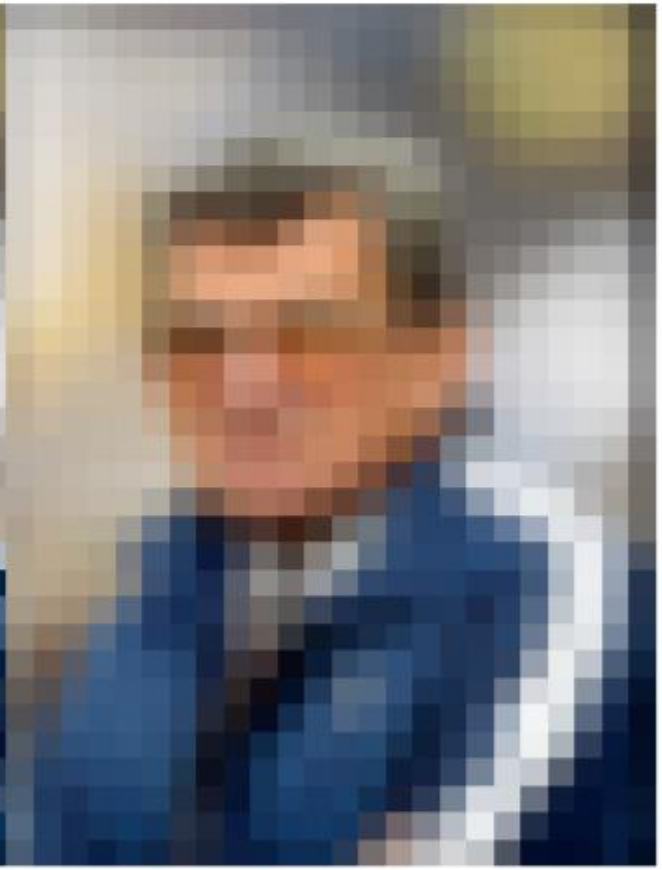
Downsampled
1/4th each dimension

Smoothed (blurred) then
downsampled

Downsampling artifacts

Original image

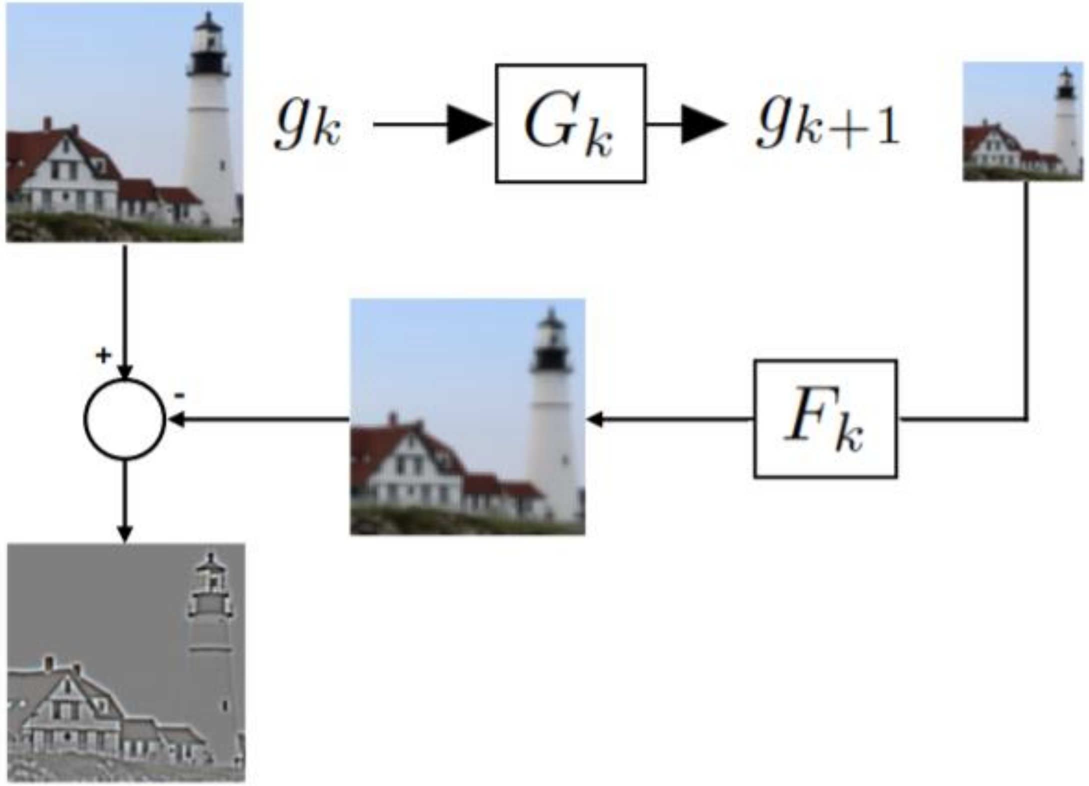
Downsampled
1/8th each dimension



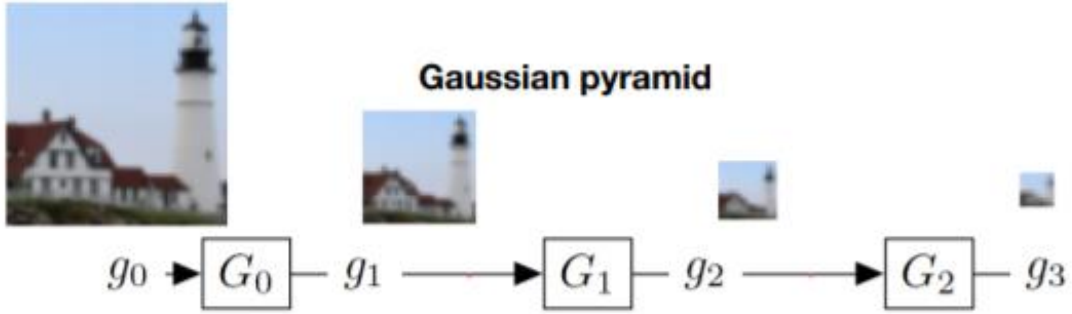
Smoothed (blurred) then
downsampled

Laplacian Pyramid

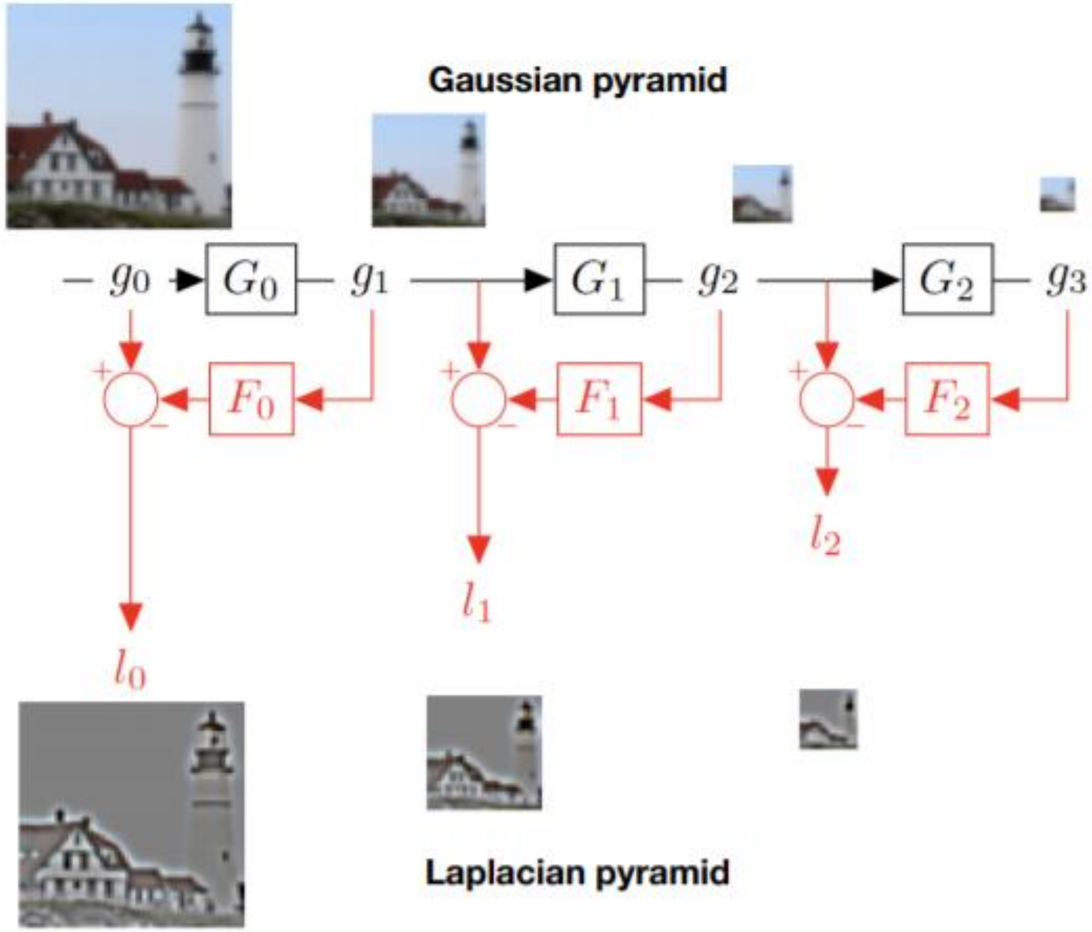
Compute the difference between upsampled Gaussian pyramid level k+1 and Gaussian pyramid level k.



Laplacian Pyramid

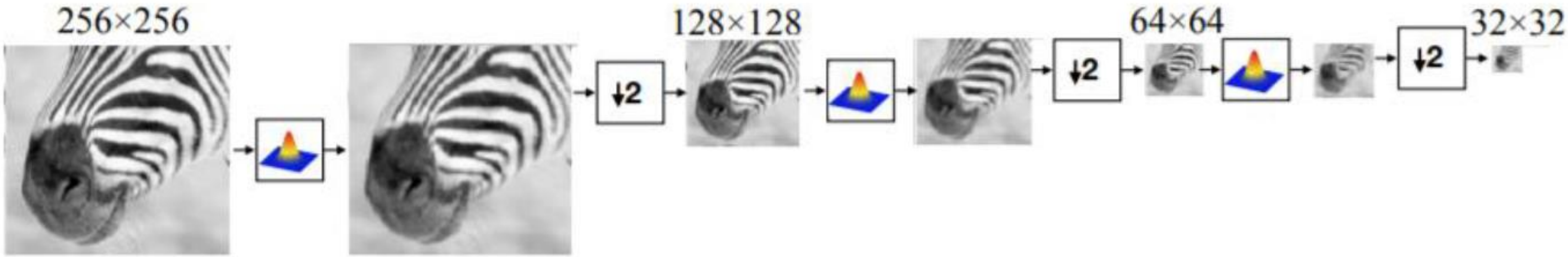


Laplacian Pyramid

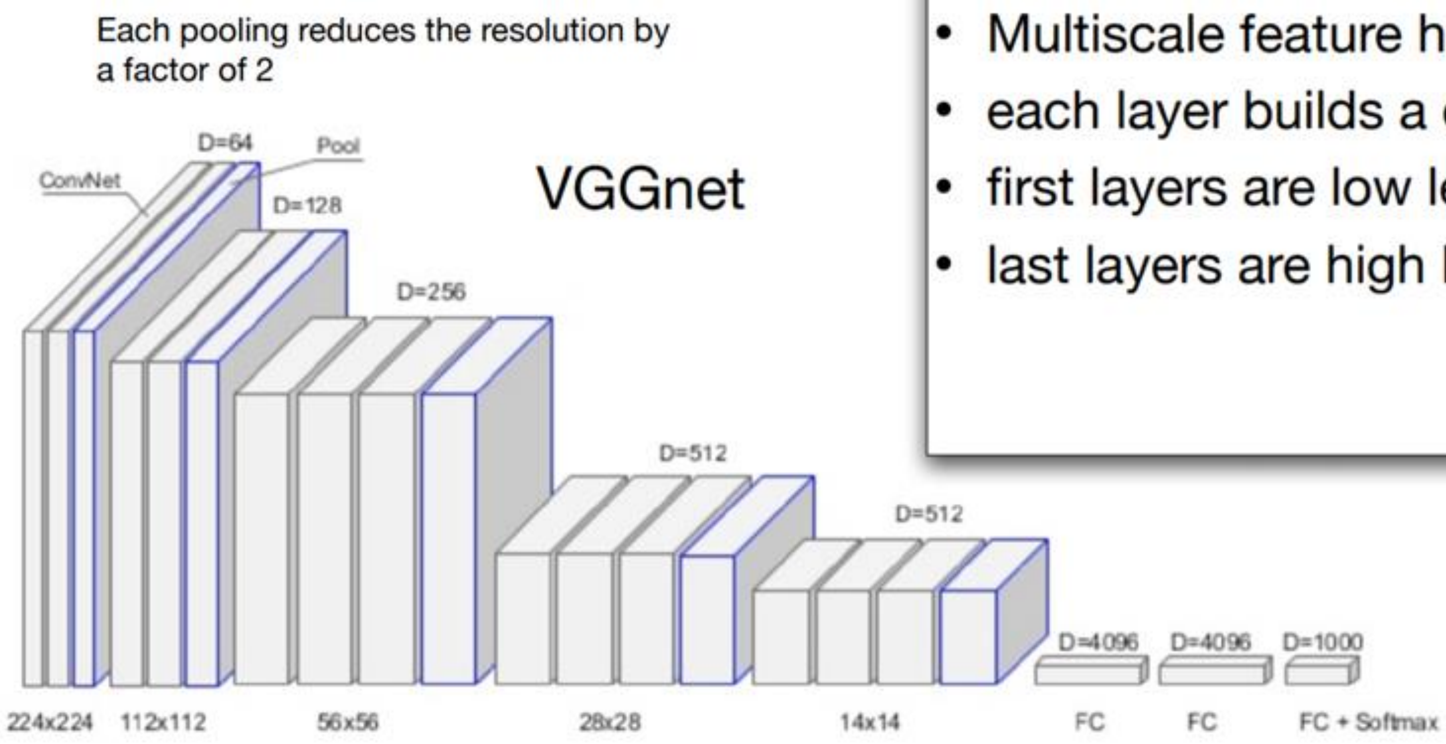


Object Detection

- Can image pyramids used to improve the performance of deep learning units?



VGGnet: A popular image classification network

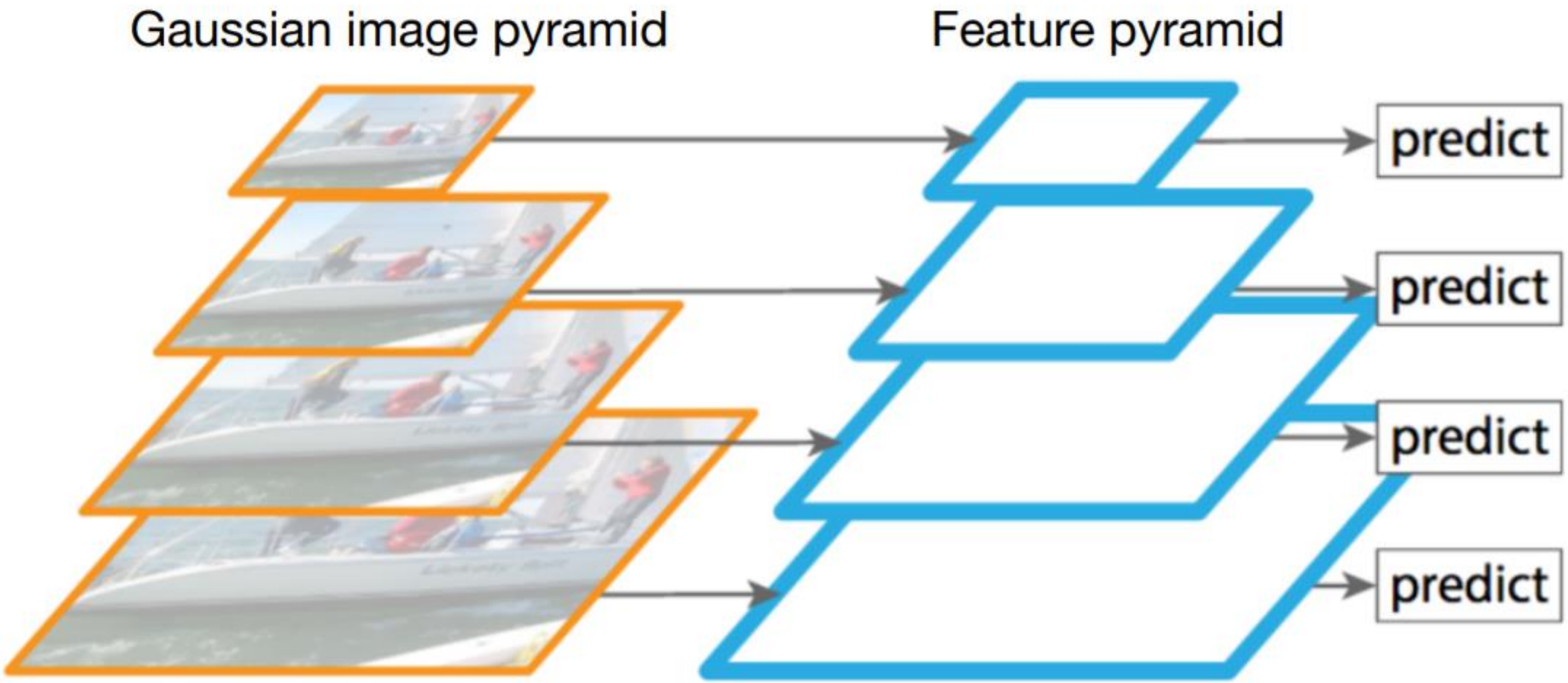


ConvNet architectures build:

- Multiscale feature hierarchies, but
- each layer builds a different representation
- first layers are low level, while
- last layers are high level.

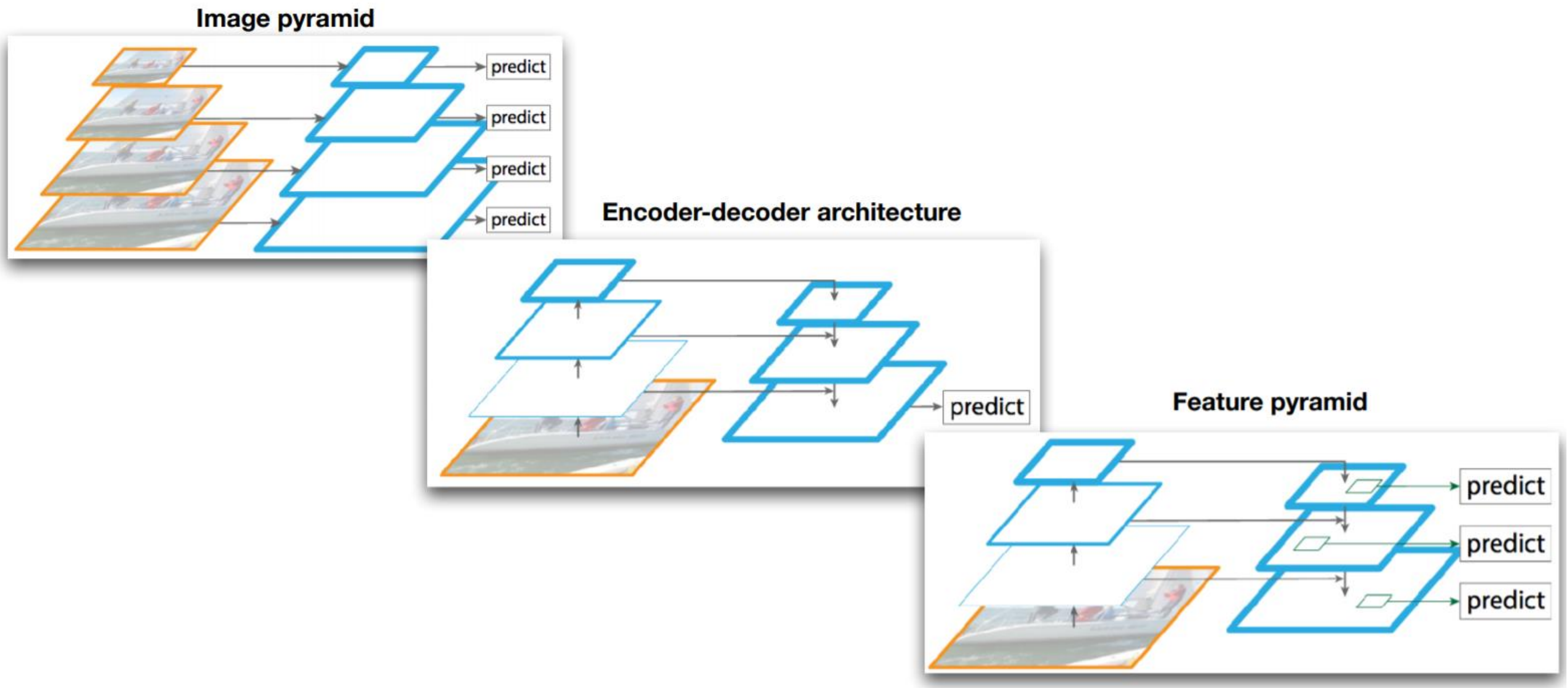
Object Detection: Image Pyramids

When should we do object detection in using pyramids ?



This will be slow and require huge computational resources

Object Detection: Image Pyramids



Lin, Tsung-Yi, et al. "Feature pyramid networks for object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

Object Detection: Image Pyramids

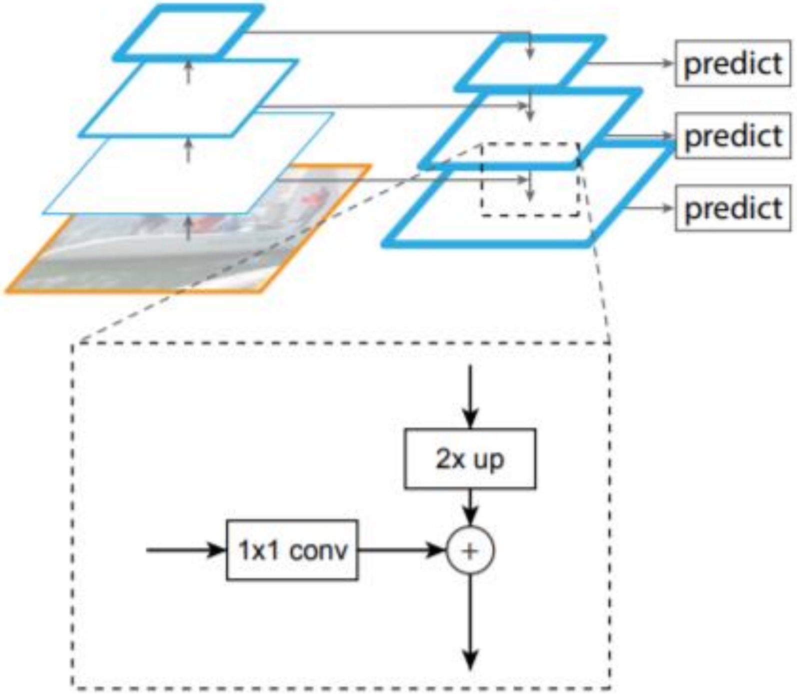
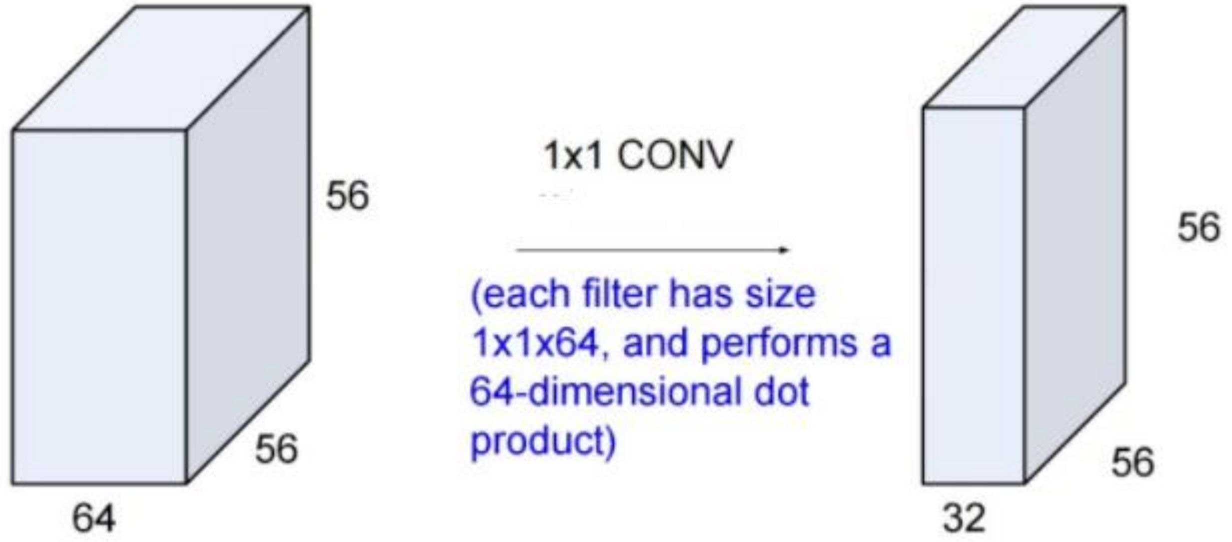


Figure 3. A building block illustrating the lateral connection and the top-down pathway, merged by addition.

1x1 Convolutions ?



- 1x1 convolutions is used to describe a 1 x 1 x Z convolution, where Z is the number of layers
- How many 1x1 convolutions used to generate the output?