BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

Learning from Images

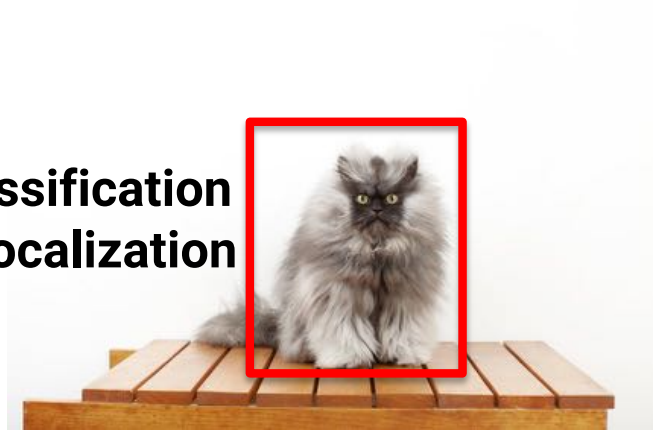
Image Segmentation and Object recognition

Master DataScience
Winter term 2019/20

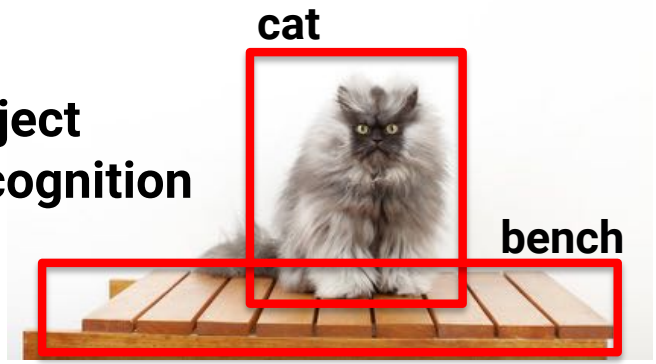
Prof. Dr. Kristian Hildebrand
khildebrand@beuth-hochschule.de

From classification to segmentation and recognition

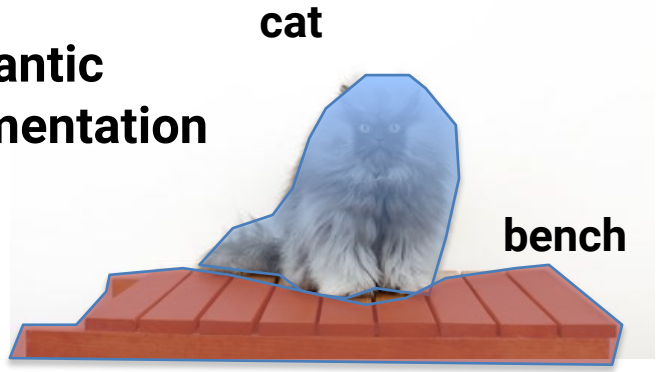
Classification
+ Localization



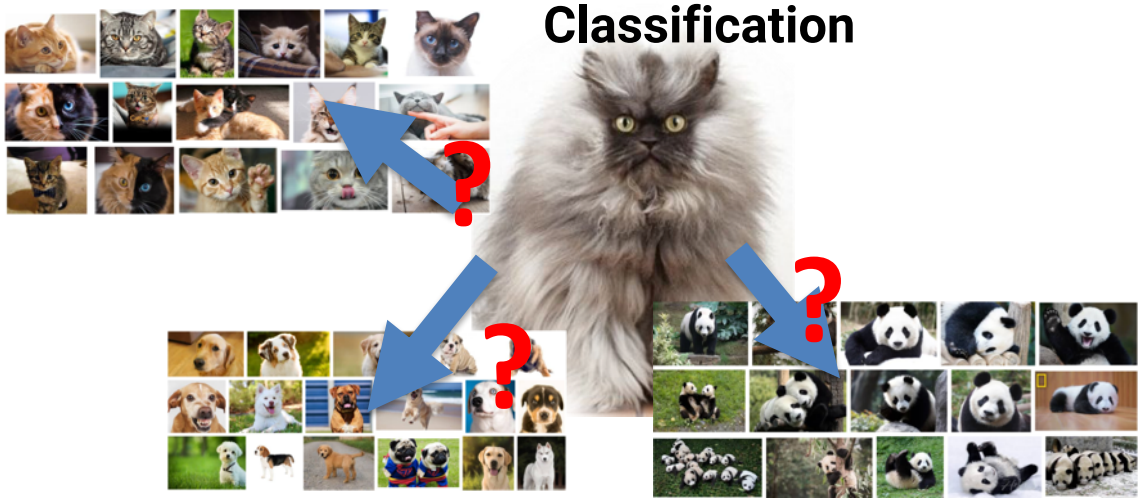
Object
recognition



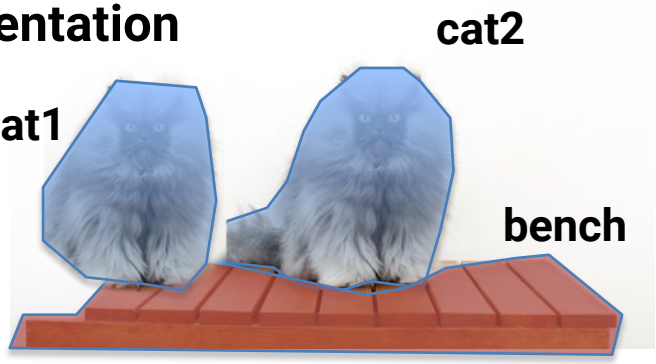
Semantic
Segmentation



Classification



Instance
Segmentation

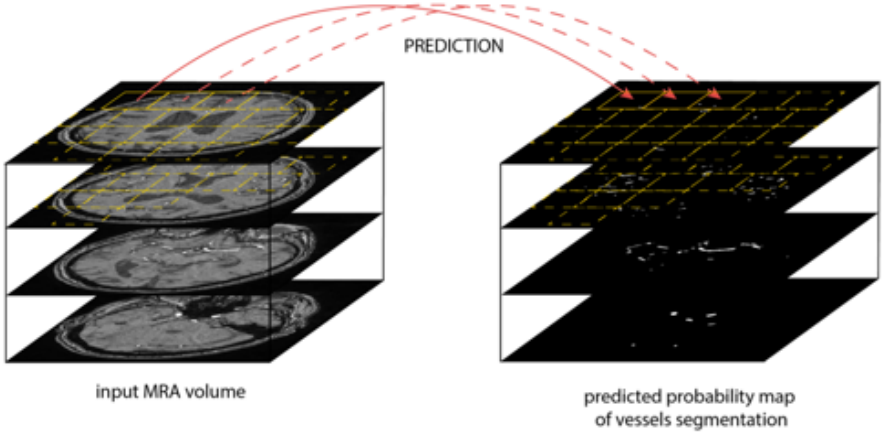


Applications

Autonomous Driving

Source: https://www.youtube.com/watch?v=ATlcEDSPWXY

Medical Imaging



Geo Sensing

Source: https://blog.playment.io/semantic-segmentation/

Entertainment / Gaming

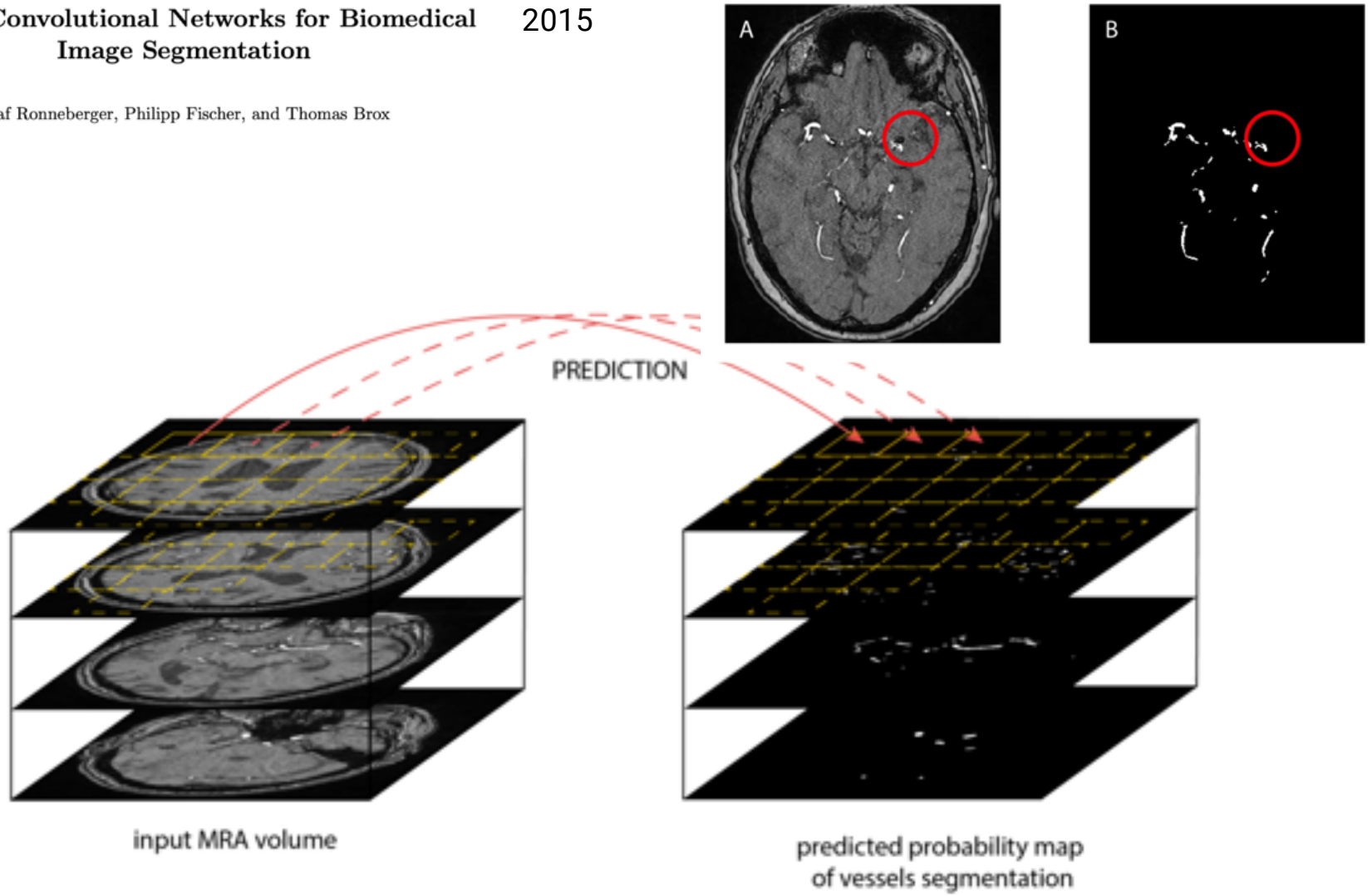
http://www.virtualairguitar.com/expertise/

Semantic Segmentation

U-Net for image segmentation

U-Net: Convolutional Networks for Biomedical Image Segmentation 2015

Olaf Ronneberger, Philipp Fischer, and Thomas Brox

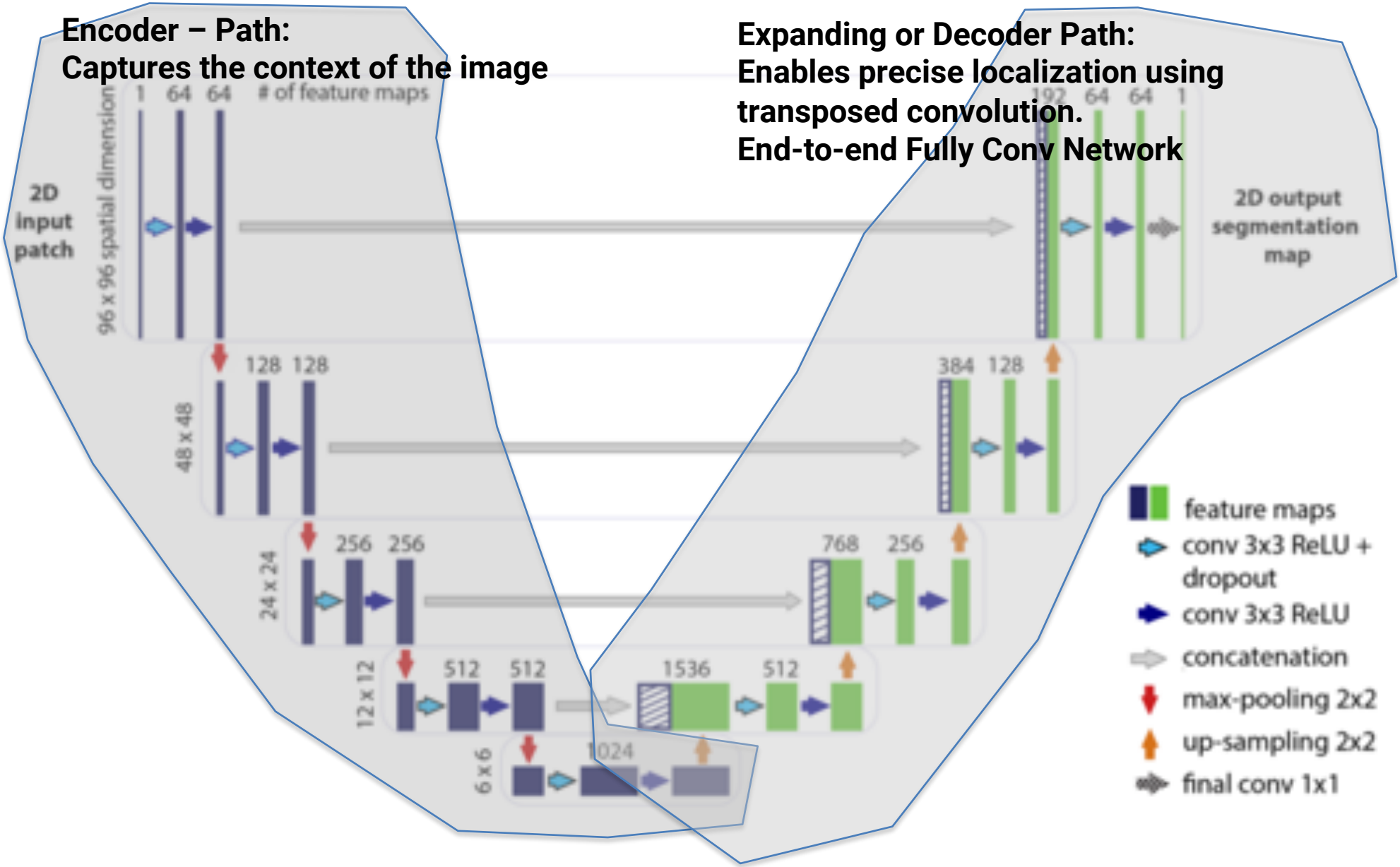


U-net Deep Learning Framework for High Performance Vessel Segmentation in Patients with Cerebrovascular Disease Frontiers in Neuroscience. 2019. Livne, Michelle and Rieger, Jana and Aydin, Orhun and Taha, Abdel and Akay, Ela and Kossen, Tabea and Sobesky, Jan and Kelleher, John and Hildebrand, Kristian and Frey, Dietmar and Madai, Vince

U-Net for image segmentation

Encoder – Path:
Captures the context of the image

Expanding or Decoder Path:
Enables precise localization using transposed convolution.
End-to-end Fully Conv Network

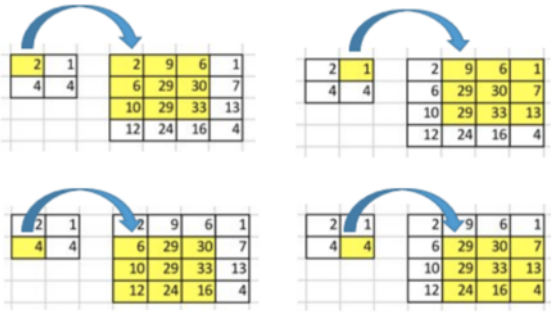


Up Sampling

Upscaling

Without global interpolation method
e.g. bi-linear or cubic interpolatin

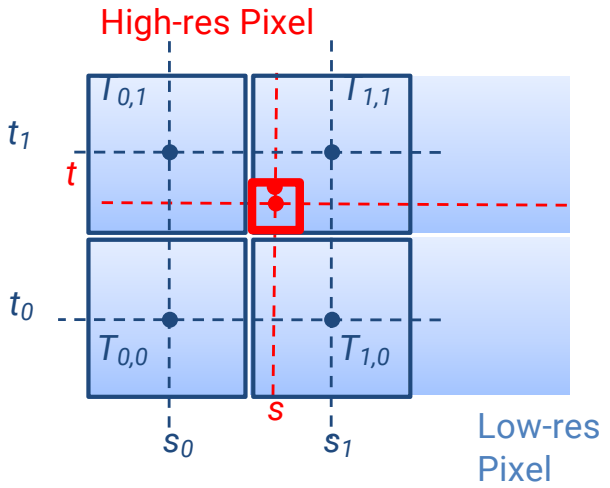
Transposed Convolution



Upsampling: Bilineare Interpolation

Nearest Neighbour

Bilinear



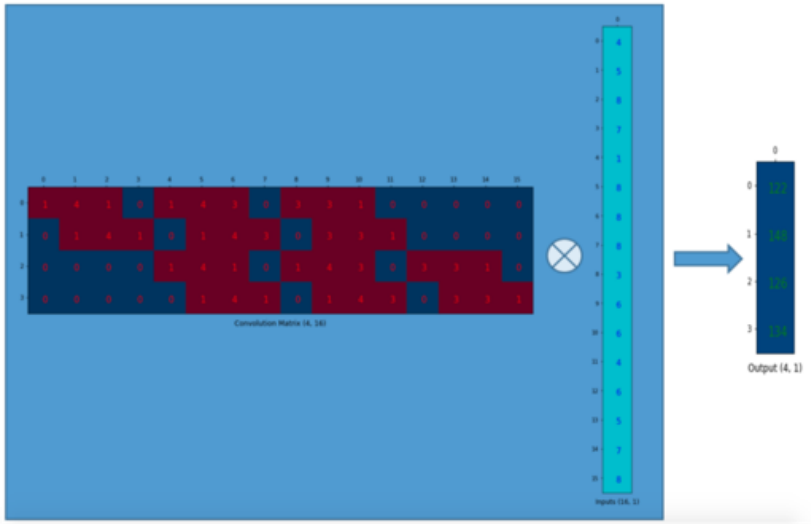
- Bilinear Interpolation uses a weighted mean from values $T_{0,0}$, $T_{0,1}$, $T_{1,0}$, $T_{1,1}$.

$$T(s,t) = (1-s')(1-t')T_{0,0} + (s')(1-t')T_{1,0} + (s')(t')T_{1,1} + (1-s')(t')T_{0,1}$$

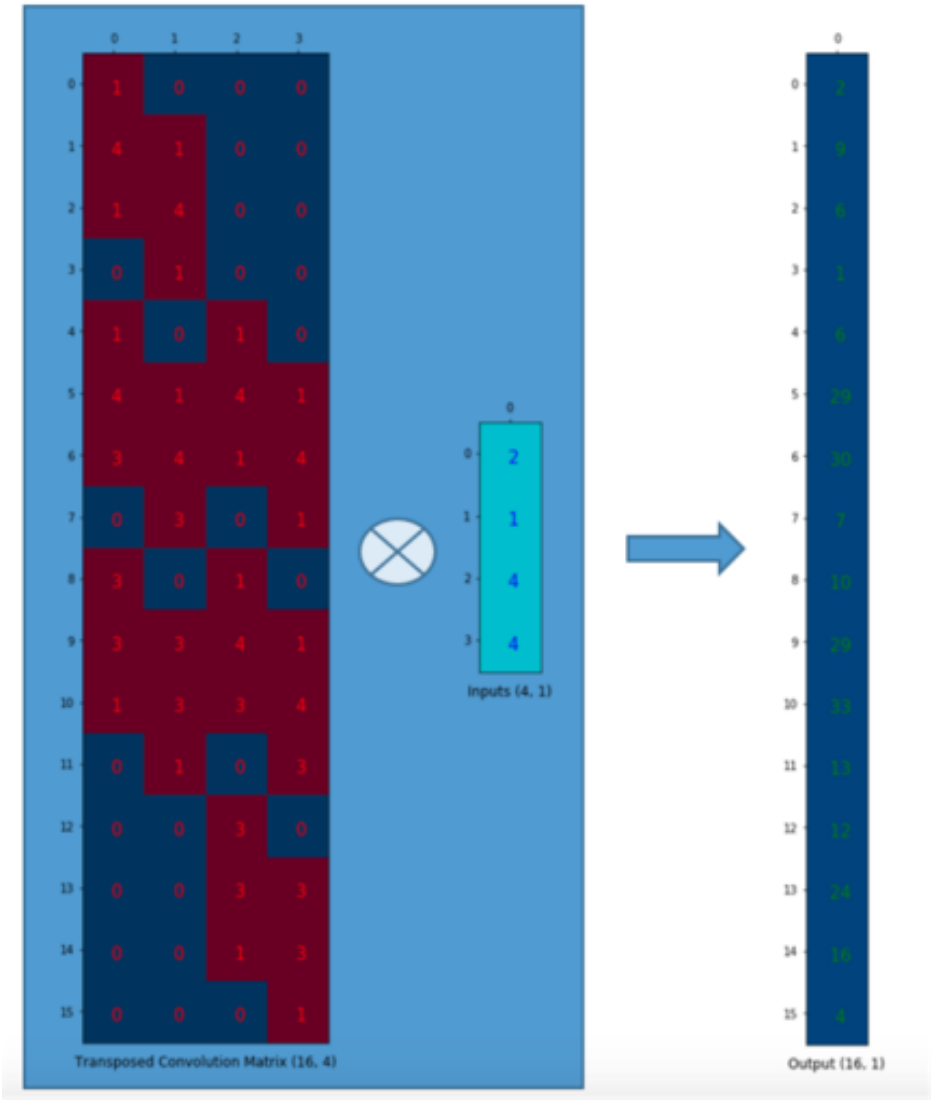
$$\text{mit} \quad s' = \frac{s-s_0}{s_1-s_0} \quad \text{und} \quad t' = \frac{t-t_0}{t_1-t_0}$$

Akenine-Möller et al., *Real-Time Rendering (3rd Edition)*, A K Peters

Transposed Convolution



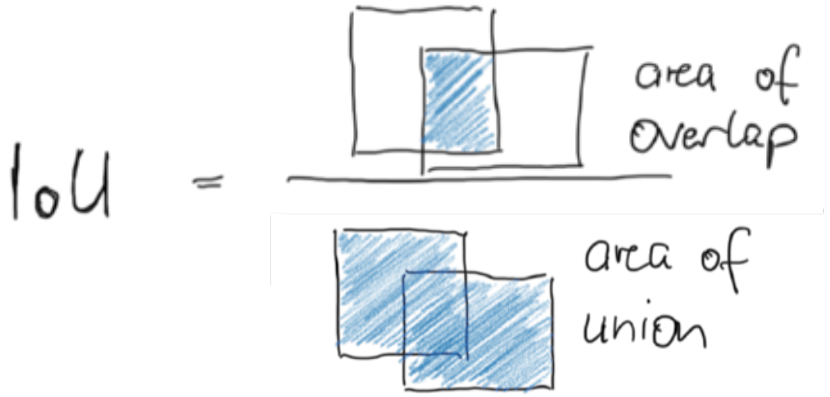
Convolution setting

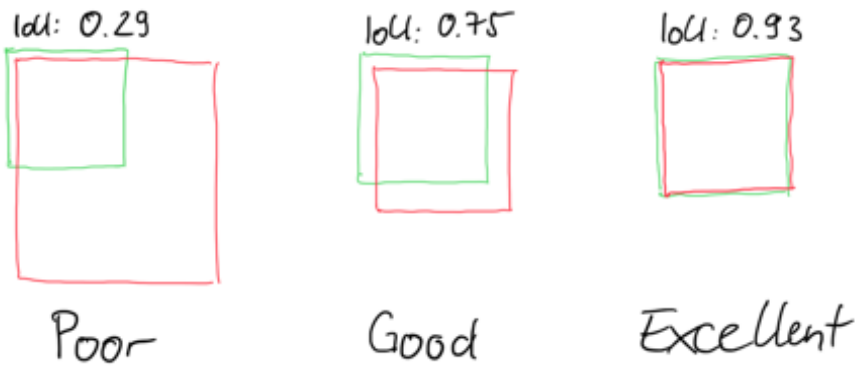


Transposed convolution setting

Intersection of Union (IoU)

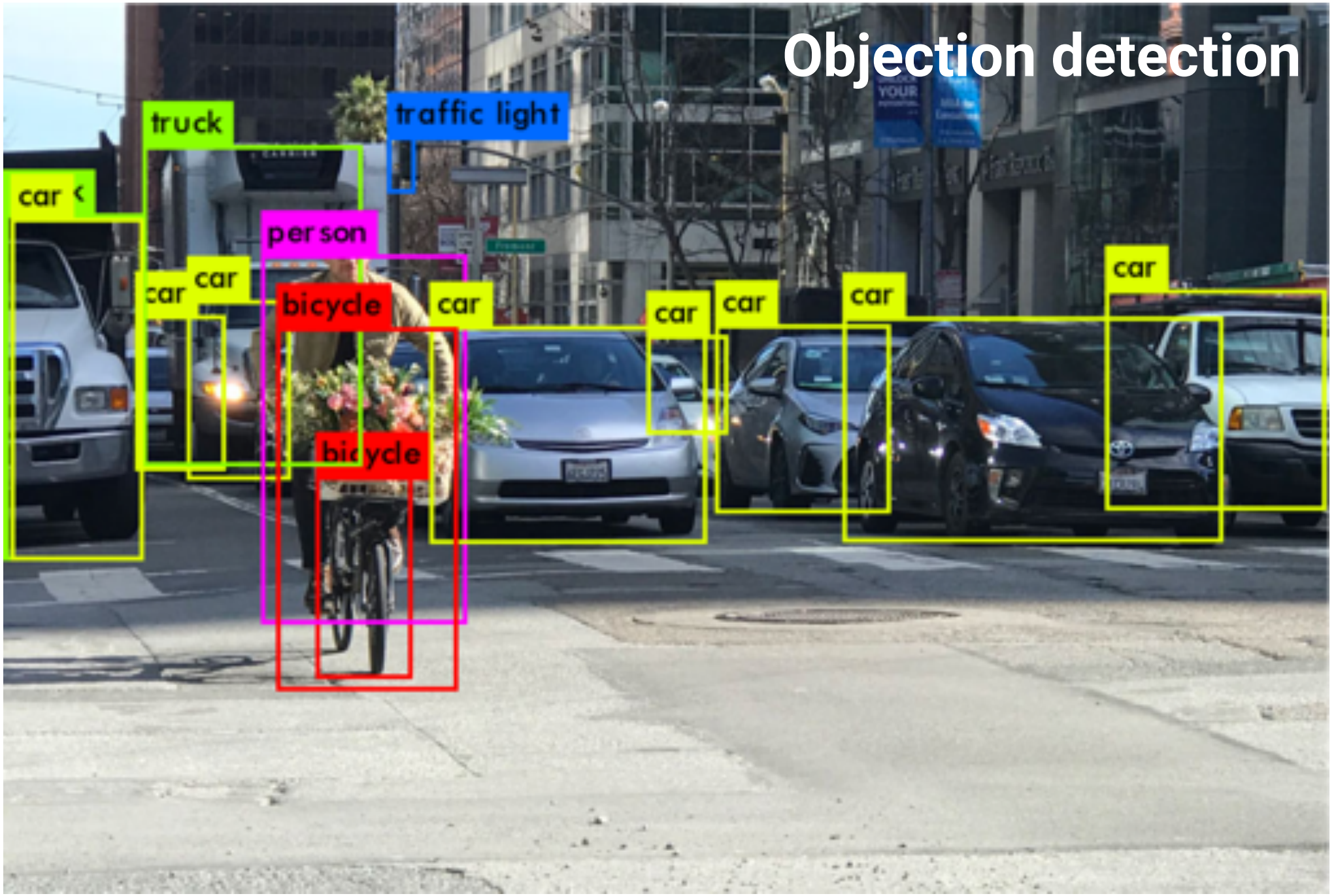
- Evaluation metric used to measure the accuracy of an object detector
  1. The *ground-truth bounding boxes* (x,y, w,h).
  2. The *predicted bounding boxes* from our model.





Dataset PASCAL VOC

- 20 classes
- train/val data has 11,530 images
- containing 27,450 ROI annotated object
- 6,929 segmentations

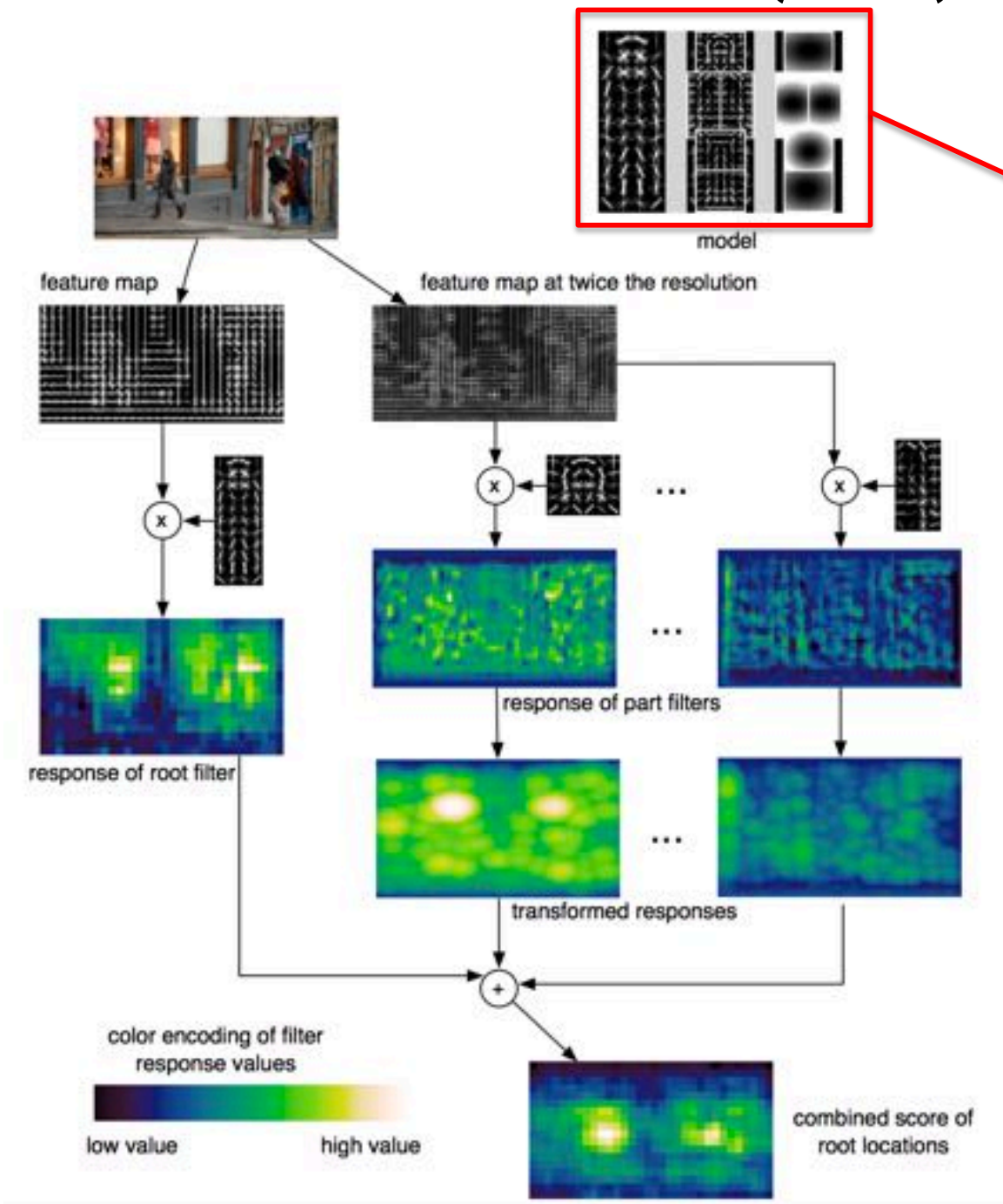

Objection detection
truck
traffic light
car
person
car
car
bicycle
car
car
car
car
car
bicycle

Deformable Model Parts (DMP)

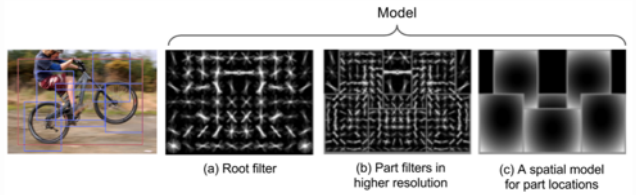
Object Detection with Discriminatively Trained Part Based Models

Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester and Deva Ramanan

2010



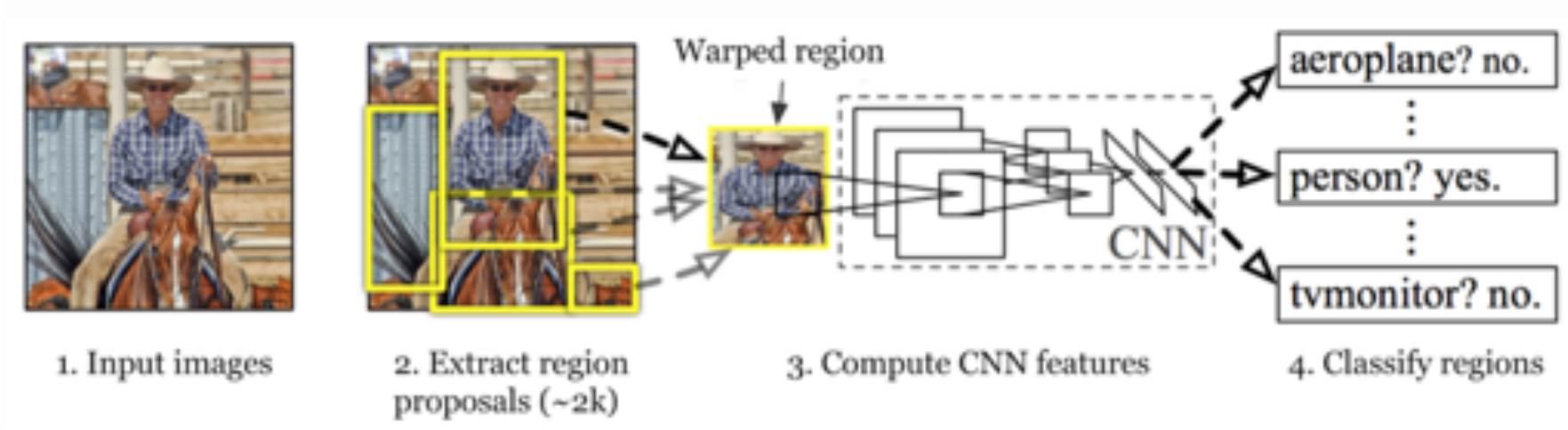
This is what we learn using **HOG** and a **Support Vector Machine**



Overview – State of the art methods

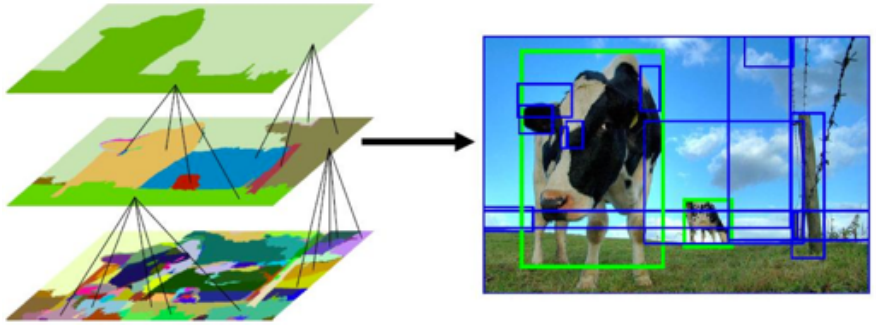
Model	Goal	Resources
R-CNN	Object recognition	[paper][code] https://arxiv.org/abs/1311.2524 https://github.com/rbgirshick/rcnn
Fast R-CNN	Object recognition	[paper][code] https://arxiv.org/abs/1504.08083 https://github.com/rbgirshick/fast-rcnn
Faster R-CNN	Object recognition	[paper][code] https://arxiv.org/abs/1506.01497 https://github.com/rbgirshick/py-faster-rcnn
Mask R-CNN	Image segmentation	[paper][code] https://arxiv.org/abs/1703.06870 https://github.com/CharlesShang/FastMaskRCNN
YOLO	Fast object recognition	[paper][code] https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf https://pjreddie.com/darknet/yolo/

R-CNN



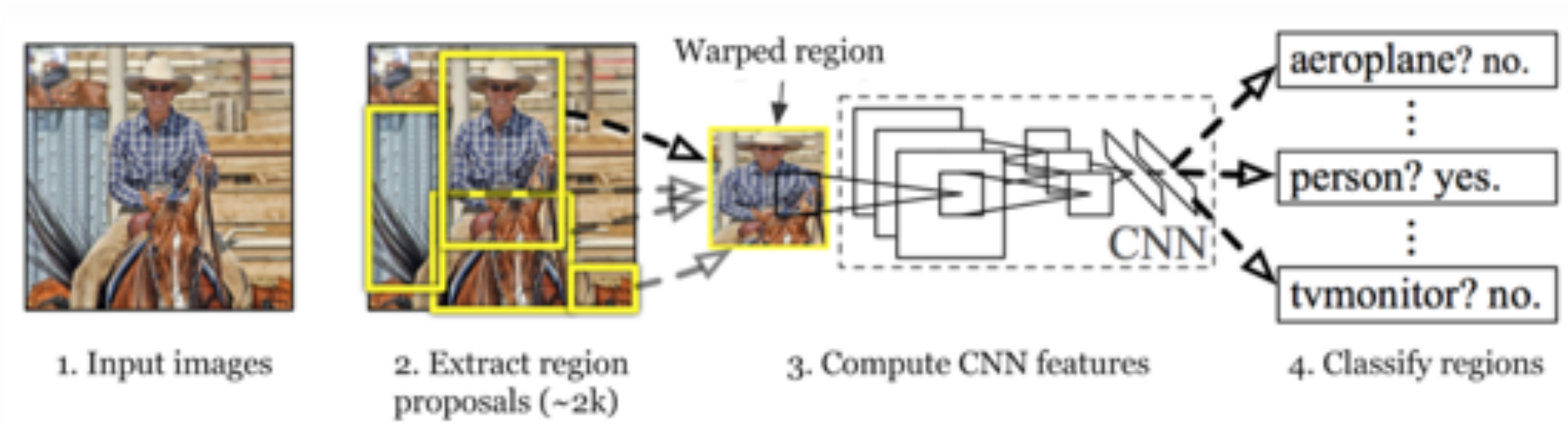
- **Input:** Pre-trained CNN network on image classification tasks (VGG or ResNet trained on ImageNet)

1. Propose category-independent regions of interest by **selective search** (~2k candidates per image). Those regions may contain target objects and they are of different sizes.
2. Region candidates are warped to have a fixed size as required by CNN.



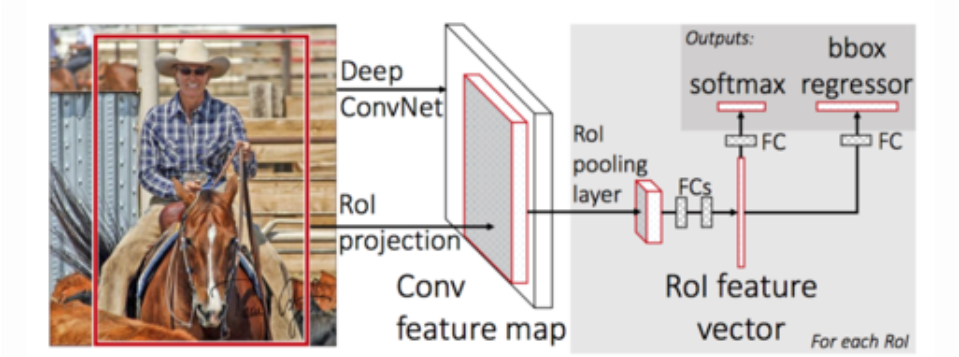
https://www.koen.me/research/selectivesearch/

R-CNN

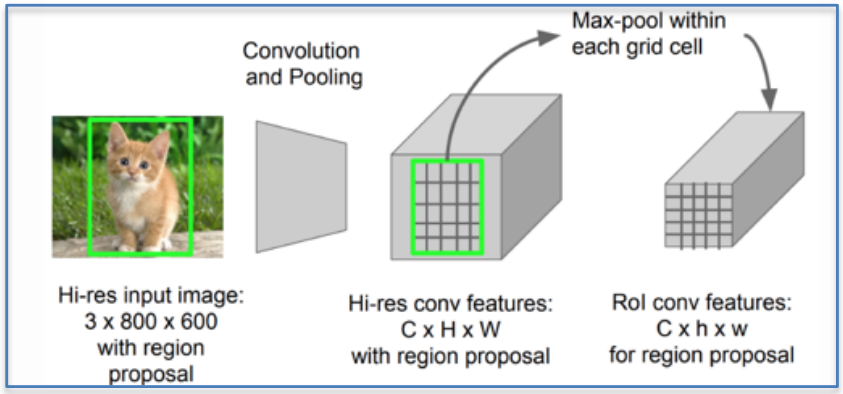


3. Continue fine-tuning CNN on warped proposal regions for K + 1 classes (additional one class refers to the background (no object of interest)).
4. Given every image region, one forward propagation through the CNN generates a feature vector. Feature vector is consumed by a binary SVM trained for each class independently.
   Positive samples are proposed regions with IoU (intersection over union) overlap threshold >= 0.3, negative samples are irrelevant others.

Fast R-CNN



RoI Pooling



3. Replace the last max pooling layer of the pre-trained CNN with RoI Pooling layer.
   - The RoI pooling layer outputs fixed-length feature vectors of region proposals.
   - Replace last fully connected layer and last softmax layer (K classes) with fully connected layer and softmax over K + 1 classes.

4. Model branches into two output layers:
   - softmax estimator of K + 1 classes outputting a discrete probability distribution per RoI.
   - Bounding-box regression model which predicts offsets relative to the original RoI for each of K classes.

Object recognition
YOLO
(You only look once)

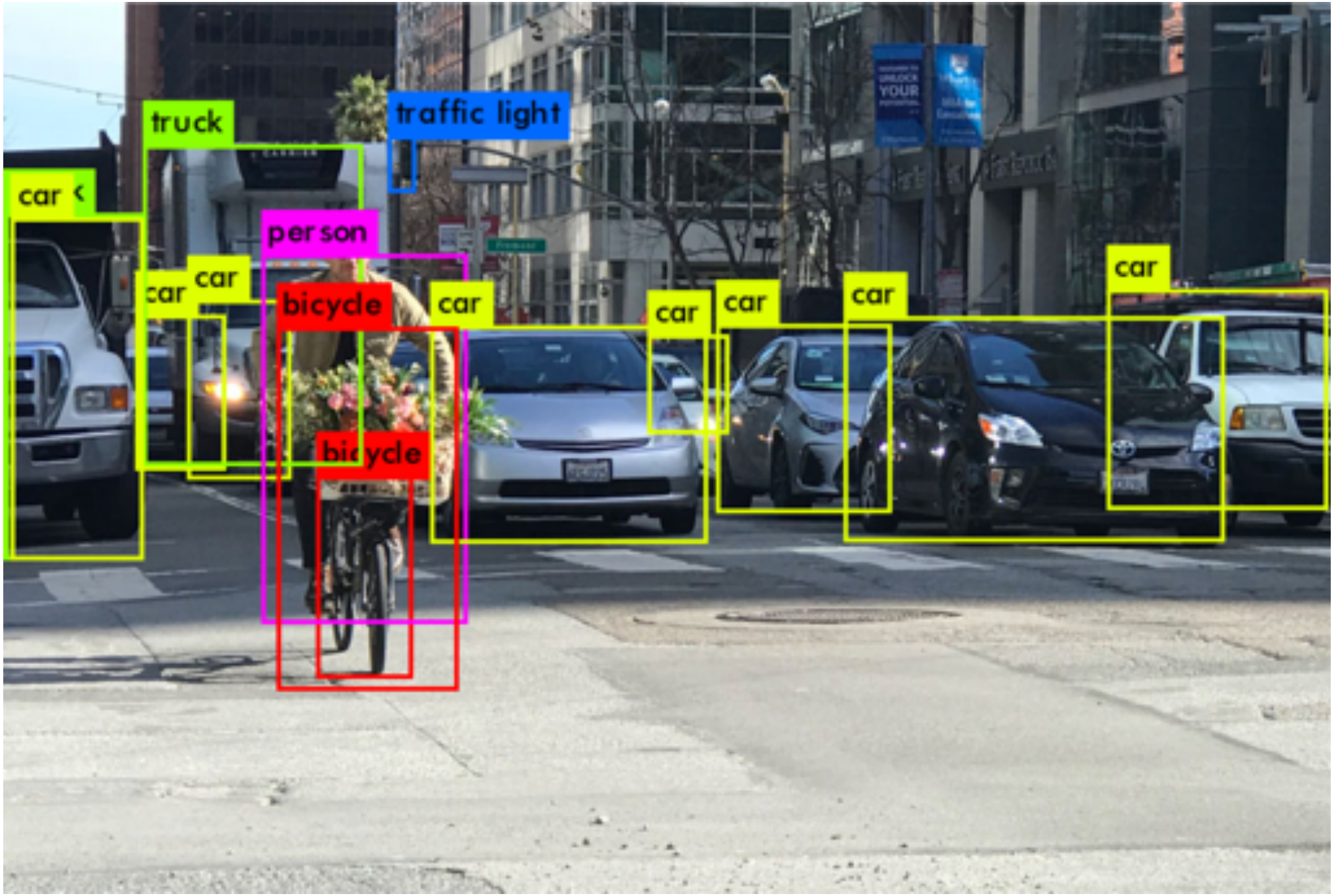
https://www.youtube.com/watch?t=&v=VOC3huqHrss

Pros: Very fast.
Cons: Accuracy tradeoff; not good at recognizing irregularly shaped objects or a group of small objects (i.e. a flock of birds?)

YOLO v1

Images taken from: https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2

YOLO v1

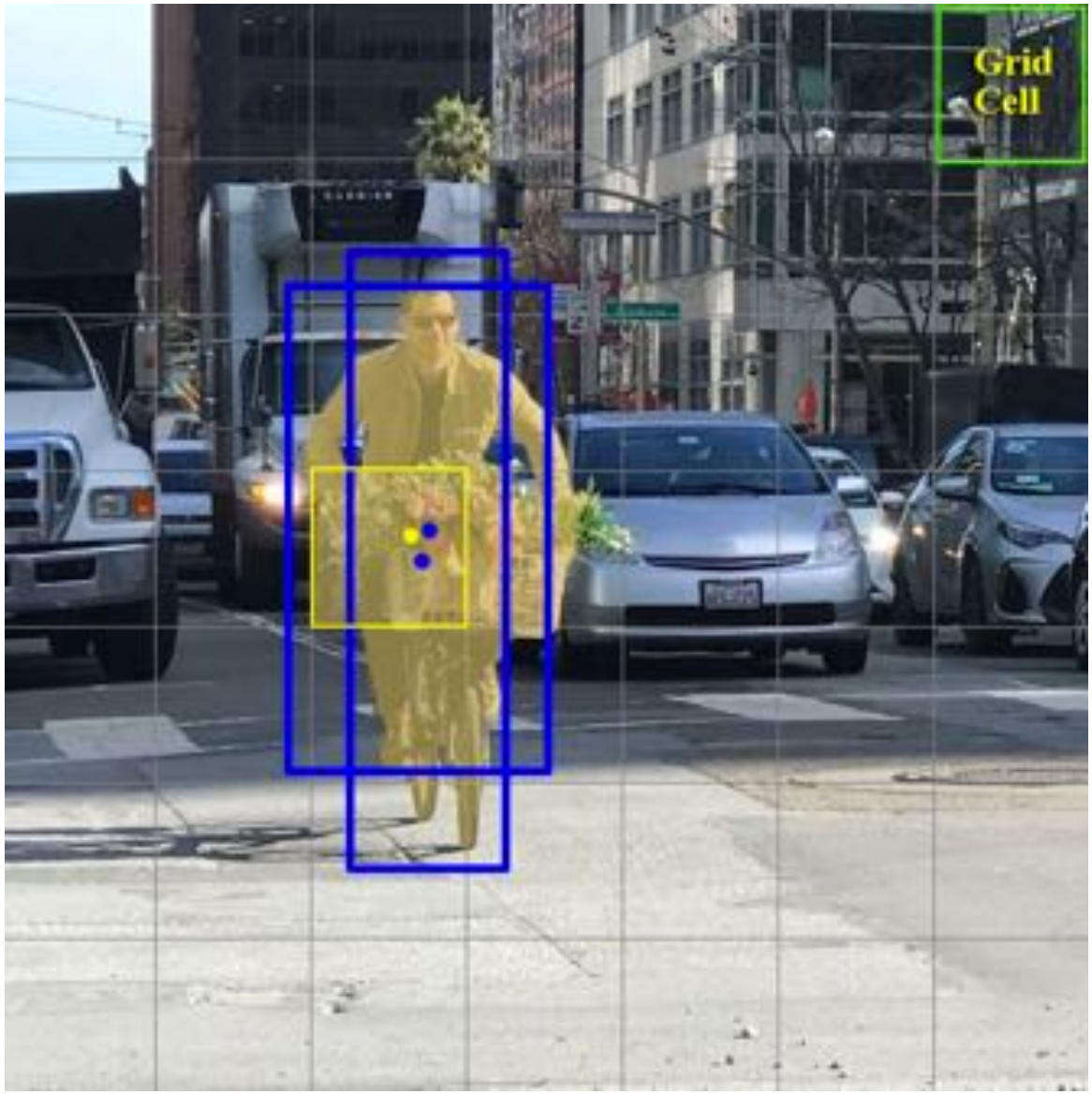

truck
traffic light
car
person
car
car
bicycle
car
car
car
car
car
bicycle

YOLO v1

1. Pre-train a CNN network on image classification tasks.
2. Split an image into S x S cells.
    1. Each cell is responsible for identifying the object (if any) with its center located in this cell.
    2. Each cell predicts
        1. location of B bounding boxes
        2. confidence score
        3. and a probability of object class conditioned on the existence of an object in the bounding box.

YOLO v1

YOLO v1



Setup:

- 7x7 grid cells (SxS)
- boundary boxes (B) with position x, y, width, height and confidence score
- 20 classes (C)

For each grid cell YOLO predicts:

- 2 boundary boxes
- one object for all boundary boxes
- 1 of 20 classes

YOLO v1

YOLO prediction tensors have a shape of
(S, S, B×5 + C) = (7, 7, 2×5 + 20) = (7, 7, 30)

YOLO

(x, y, w, h) + confidence score (objectness+box)

(x, y, w, h) + confidence score (objectness+box)

car: 0.05
person: 0.10
...
truck : 0.02

...

cell(1, 1)

...

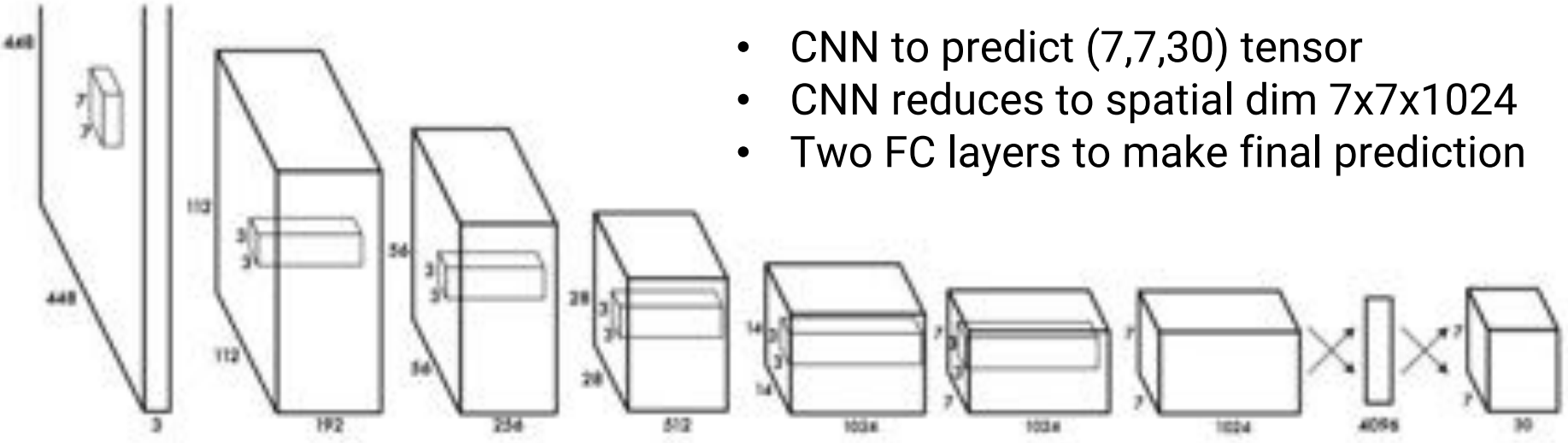
...

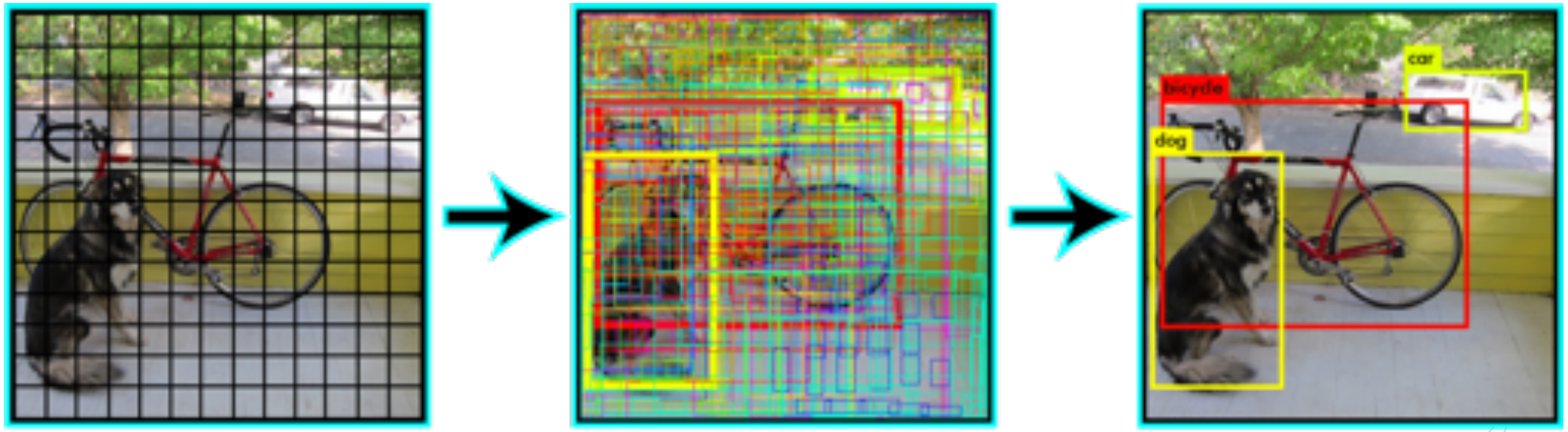
cell(S, S)

YOLO v1

YOLO prediction tensors have a shape of
(S, S, B×5 + C) = (7, 7, 2×5 + 20) = (7, 7, 30)

- CNN to predict (7,7,30) tensor
- CNN reduces to spatial dim 7x7x1024
- Two FC layers to make final prediction





YOLO v1 Loss function

- Loss function consists of three terms:
  - Classification lo

$$\sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2$$

+

  - Localization loss

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2\right]$$

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2\right]$$

+

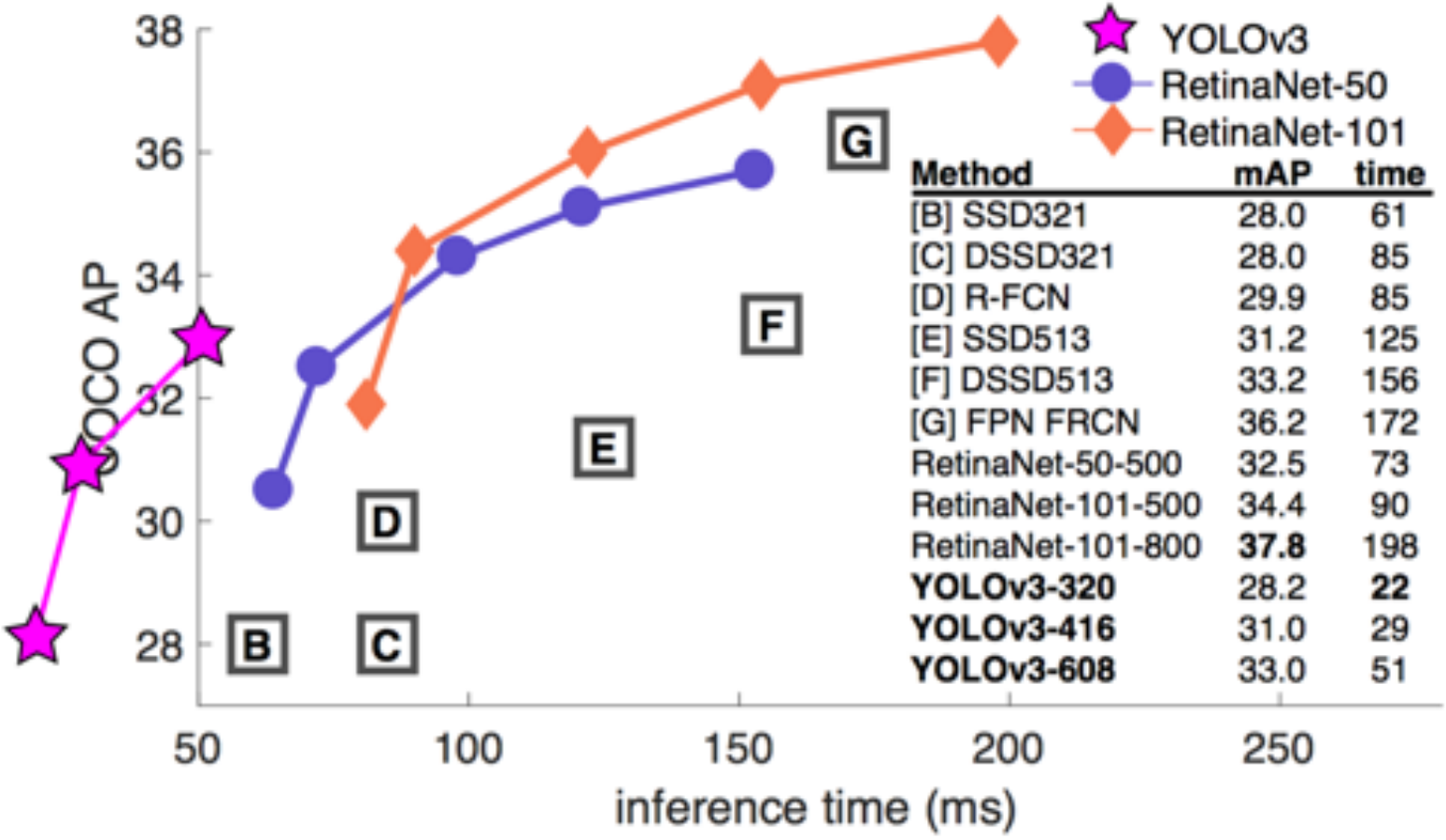
  - Confidence loss

$$\sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i\right)^2$$

$$\lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i\right)^2$$

YOLO v2 --> YOLO v3

https://www.youtube.com/watch?v=MPU2HistivI

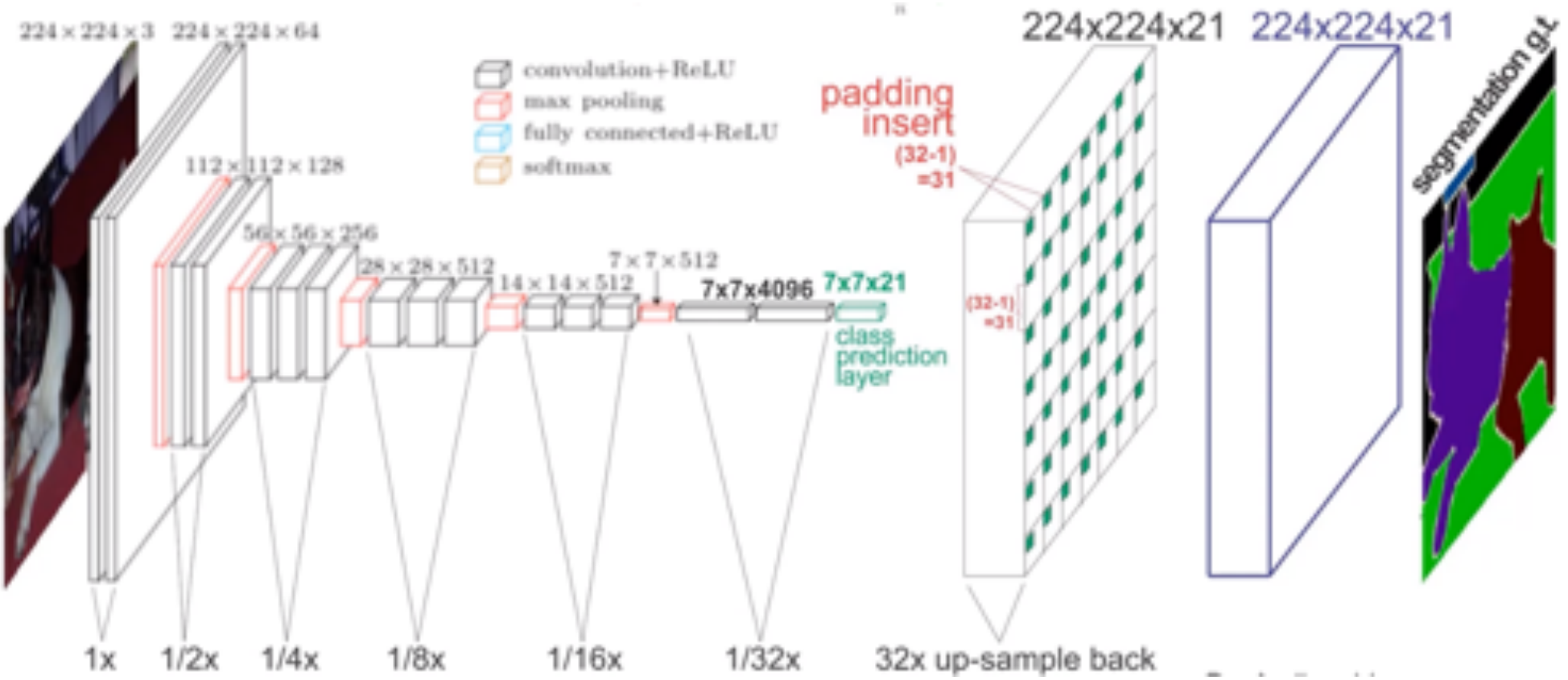


Method	mAP	time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	**37.8**	198
YOLOv3-320	28.2	**22**
YOLOv3-416	31.0	29
YOLOv3-608	33.0	51

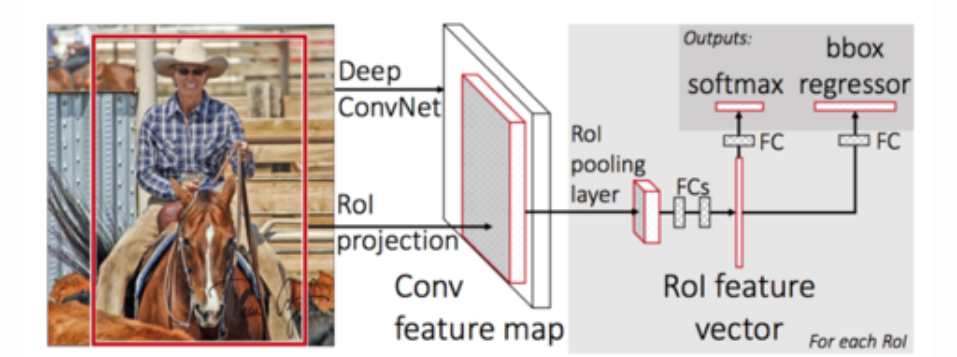
Instance Segmentation

Mask R-CNN

Mask R-CNN


224 × 224 × 3
224 × 224 × 64
112 × 112 × 128
56 × 56 × 256
28 × 28 × 512
14 × 14 × 512
7 × 7 × 512
7x7x4096
7x7x21
convolution+ReLU
max pooling
fully connected+ReLU
softmax
class prediction layer
1x
1/2x
1/4x
1/8x
1/16x
1/32x
padding insert
(32-1) =31
(32-1) =31
224x224x21
224x224x21
segmentation g.t
32x up-sample back

Mask R-CNN



- Alter the pre-trained CNN:
  - Replace the last max pooling layer of the pre-trained CNN with a RoI pooling layer. The RoI pooling layer outputs fixed-length feature vectors of region proposals.
- Finally the model branches into two output layers:
  - A softmax estimator of K + 1 classes (same as in R-CNN, +1 is the “background” class), outputting a discrete probability distribution per RoI.
  - A bounding-box regression model which predicts offsets relative to the original RoI for each of K classes.