UNIVERSIDAD POLITÉCNICA DE MADRID

Departamento Inteligencia Artificial

ETS DE INGENIEROS INFORMÁTICOS

Unit 2

Object Detection

Deep Learning

Professor

- Office: 2101
- Email: rvalle@fi.upm.es
- Personal web: https://bobetocalo.github.io/
- Department: https://dia.fi.upm.es/
- Research group: https://pcr-upm.github.io/

Copyright © Roberto Valle These slides are licensed under CC BY-SA 4.0,
Creative Commons Attribution-ShareAlike 4.0 International.

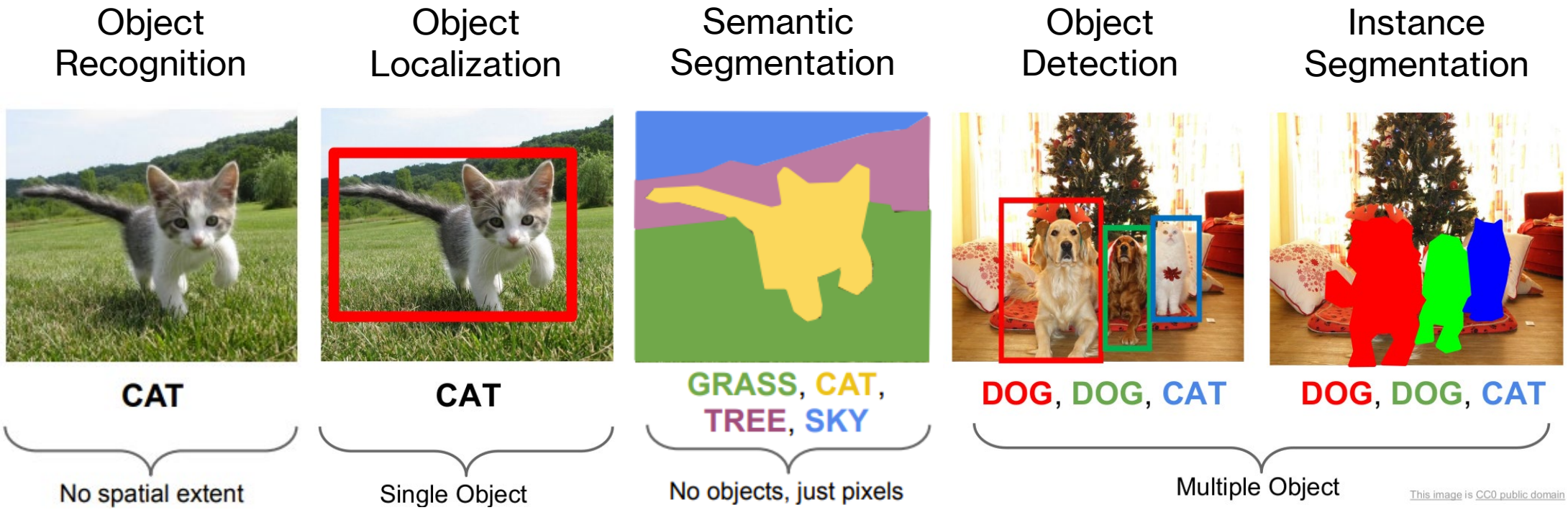
Departamento de Inteligencia Artificial

Grupo de Percepción Computacional y Robótica

Contents

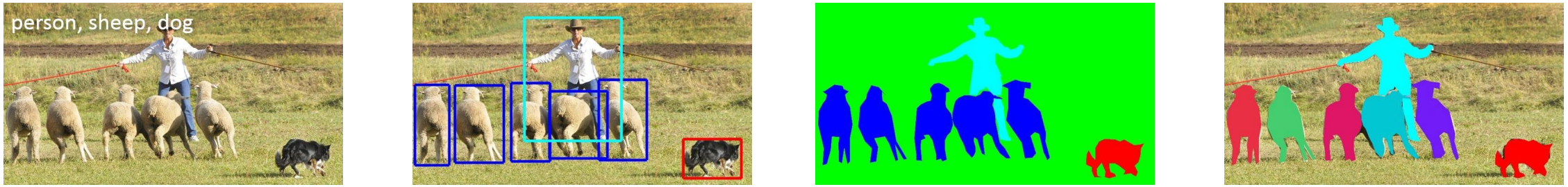
- *Traditional machine learning approaches*
  - *Sliding window*
  - *Non-maximum suppression*
  - *Detection metrics*
- Deep learning approaches
  - Two-stage detection
  - Single-stage detection

Beyond image classification



© http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf

Object detection datasets



- **PASCAL VOC** 2007-2012: 20 classes; images 5-11K train/val, 5-11K test.
- **ImageNet ILSVRC** 2010-2017: 200 classes (subset or merged from classification task); images 400-450K train (partially annotated), 20K val, 40K test.
- **COCO** 2015-now: 80 classes; images 80K train, 40K val, 40K test, 120K unlabelled; smaller objects.
- **Open Images** 2018-now: 600 classes; images 1.74M train, 41K val, 125K test.

© Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper R. R. Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, Vittorio Ferrari: The Open Images Dataset V4 IJCV 2020

Traditional sliding window

- It's natural to think of building an object detection on the top of an image classification model.
- Once we have a good image classifier, a simple way to detect objects is to slide a "window" across the image and classify the cropped image in that window.

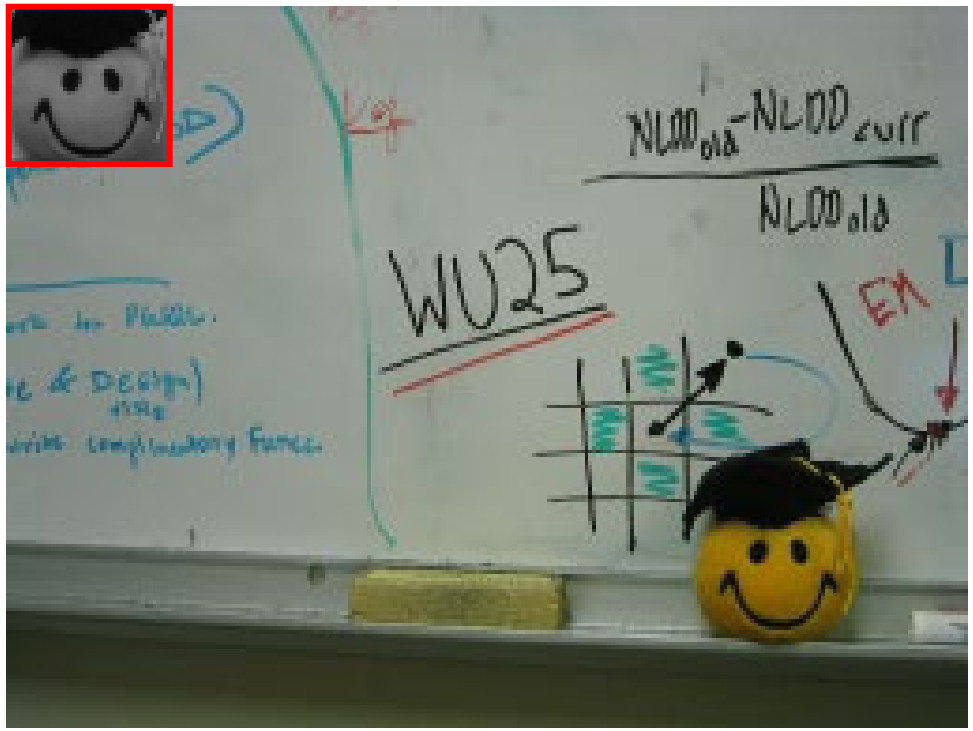
28x28x3 image + 14x14 kernel + stride 7 = 16 windows

Problems:

- Computationally infeasible
  - The size of the sliding window determines the scale at which objects can be detected (objects can be of varying sizes)
  - A fixed rectangular or square sliding window may not always align well with objects of irregular shapes (aspect ratio)
  - The step size (stride) of the sliding window determines how finely the image is scanned.
- Very imbalanced

Template matching

- A predefined template of the object is compared to each sliding window using a similarity measure.

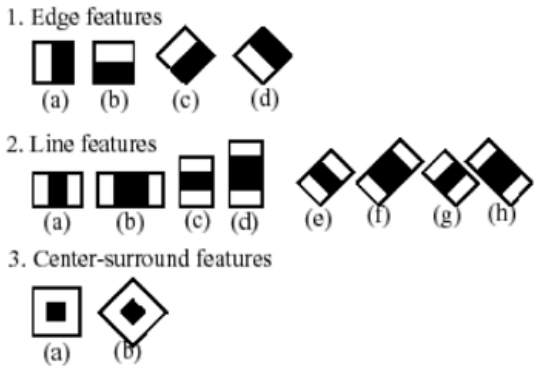


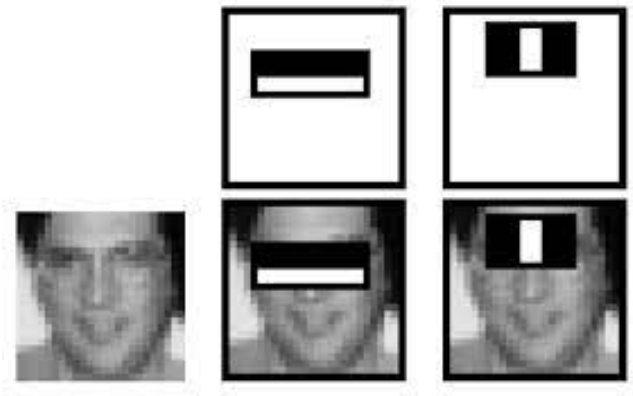
Seek maximum similarity score

© https://sif-dlv.github.io/slides/det.pdf

Haar-like features

- To detect a given instance (template), a similarity score may be enough; but to detect an object of a given class, we need strong **features** and a good **classifier**.
- The Haar features used in the popular Viola-Jones detector makes it easy to find out the edges or the lines in the image, or to pick areas where there is a sudden change in the intensities of the pixels.
- The “AdaBoost” classifier selects the most important features to form a strong classifier.

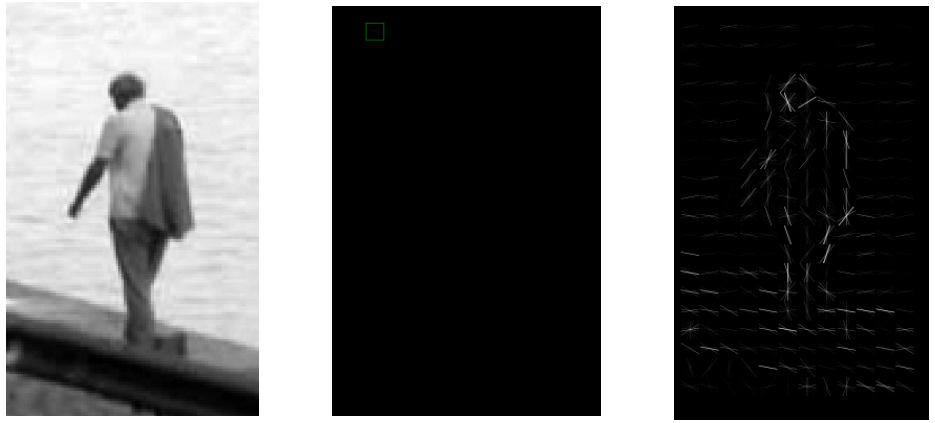




© Paul A. Viola, Michael J. Jones: Rapid Object Detection using a Boosted Cascade of Simple Features CVPR 2001

Histogram of oriented gradients (HOG)

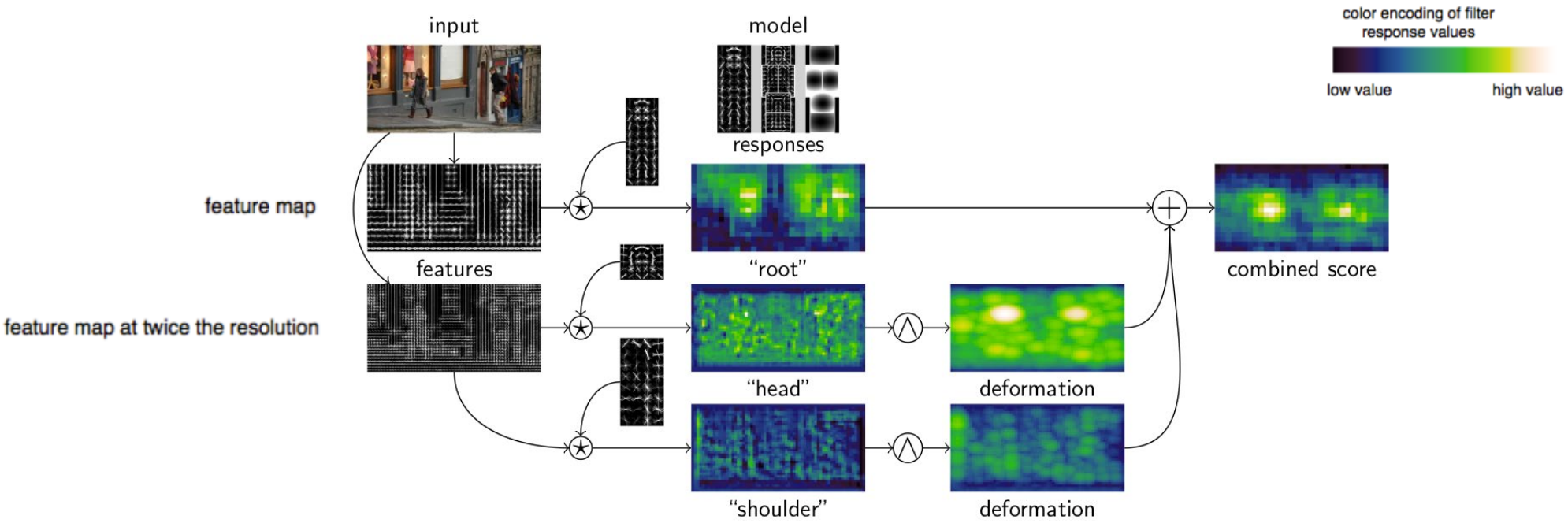
- Sliding window at all positions and scales but using HOG features instead of previous Haar-like.
- After obtaining this feature vector for a given image, a SVM is trained to detect whether this feature vector is of an object of interest or not.
- HoG features are good at describing object shape hence good for pedestrian detection.



© Navneet Dalal, Bill Triggs: Histograms of Oriented Gradients for Human Detection CVPR 2005

Deformable part model (DPM)

- Captures HOG features at two-resolutions. Body parts (head, shoulder, ...) are represented at twice the resolution of the root filter (body).



© Pedro F. Felzenszwalb, David A. McAllester, Deva Ramanan: A discriminatively trained, multiscale, deformable part model CVPR 2008

What is an object?

- Deal with the computationally infeasible problem by reducing the number of windows to evaluate.
- Seek a generic, class-agnostic objectness measure, quantifying how likely it is for an image region to contain an object. If the measure is simple and fast to compute, it can yield several candidate object proposals or **regions of interest (RoI)** where to apply a more expensive classifier.

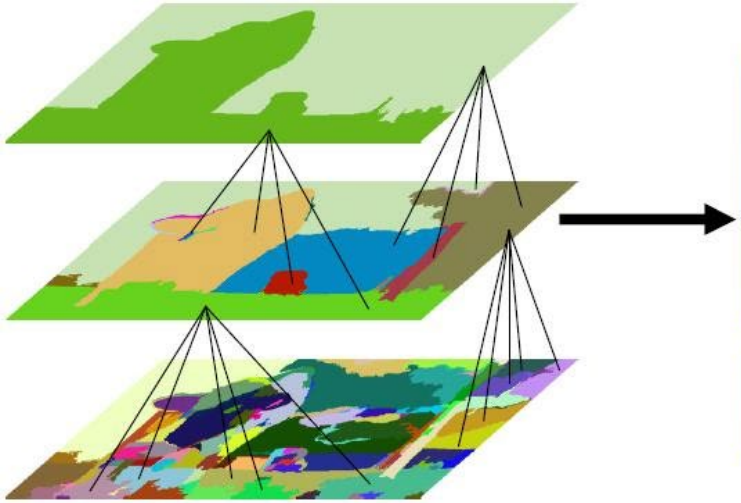
© Bodgan Alexe, Thomas Deselaers, Vittorio Ferrari: What is an object? CVPR 2010

Selective search (SS)

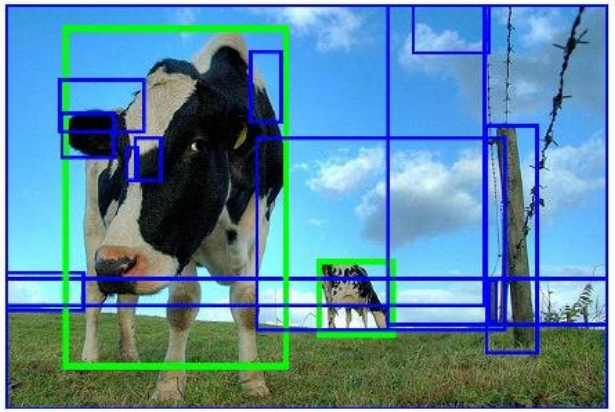
- Hierarchical segmentation at all scales. High recall: ~97% of ground truth objects found with ~2000 proposals per image at ~5s per image on average.

Input image

Hierarchical grouping



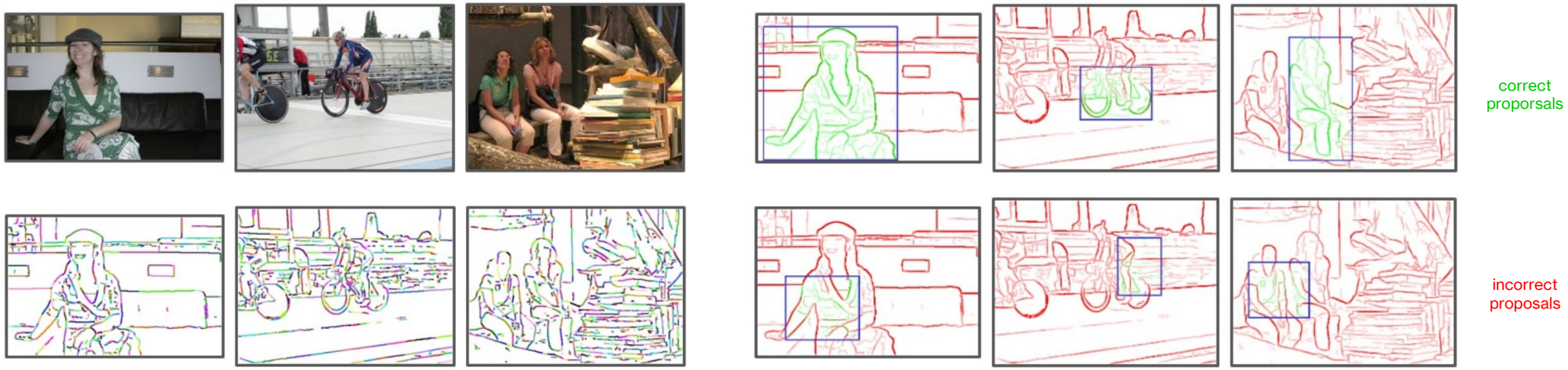
Object proposals



© Koen E. A. van de Sande, Jasper R. R. Uijlings, Theo Gevers, Arnold W. M. Smeulders: Segmentation as selective search for object recognition ICCV 2011

Edge boxes (EB)

- Scoring a candidate box based solely on the number of edges it wholly encloses, creates a surprisingly effective object proposal measure.
- Performance like SS, but at ~0.25s per image on average.



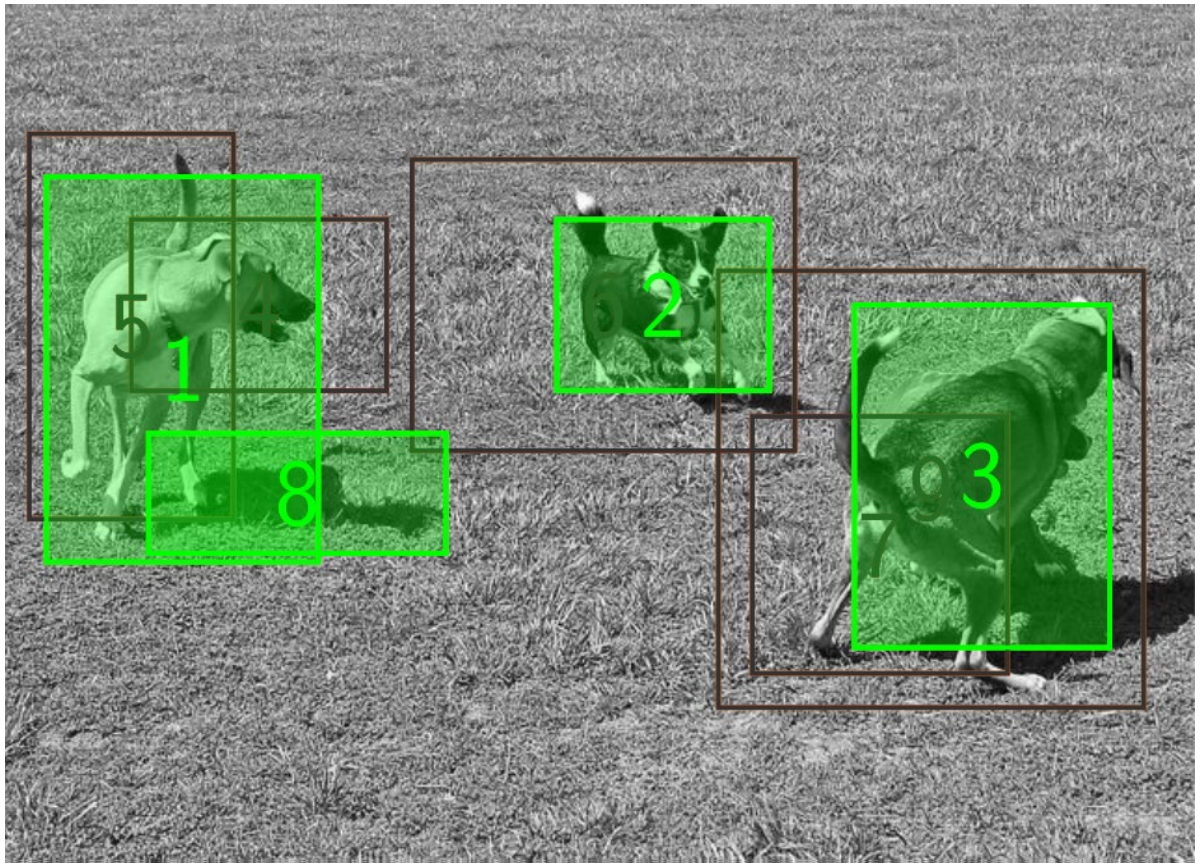
© C. Lawrence Zitnick, Piotr Dollár: Edge Boxes: Locating Object Proposals from Edges ECCV 2014

Non-maximum suppression (NMS)

- Given predictions $r_1, r_2, \ldots$ of each class independently, ranked by decreasing order of confidence score.
- For $i=2,3,\ldots$ reject prediction r_i if it has **intersection-over-union (IoU)** overlap higher than a threshold

$$J(r_i, r_j) > \tau$$

with some higher scoring region r_j with $j < i$ that has not been rejected.

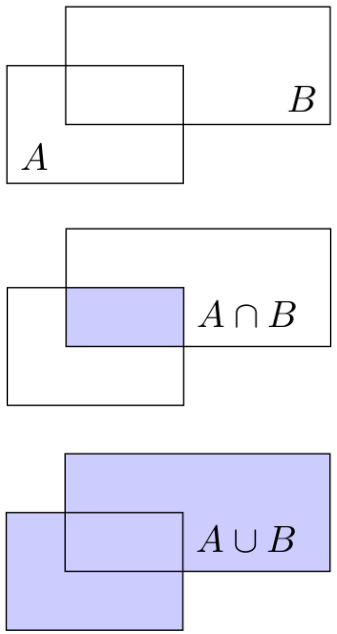


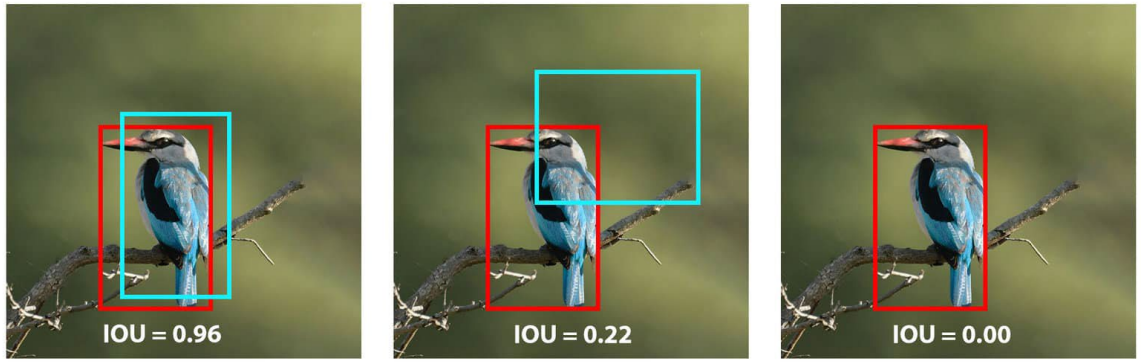
© https://sif-dlv.github.io/slides/det.pdf

Region overlap

- Given predictions A, B represented as planar point sets (including interior).
- Their **intersection-over-union (IoU)** or **Jaccard index** is:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

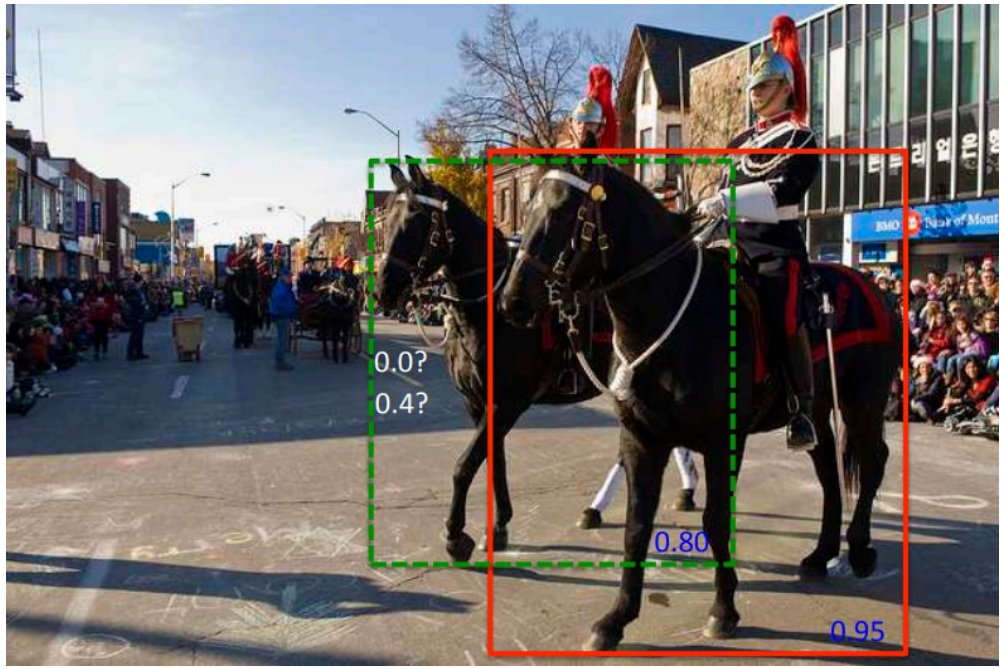




True Positive | False Positive | False Negative

The problems of NMS

- Overlap threshold τ balances two conflicting needs.
  - When the overlapping threshold is larger, proposals are less likely to be suppressed. This leads to larger number of false-positives and hence drop precision.
  - When the overlapping threshold is smaller, proposals get suppressed too aggressively and hence reduce recall.
- There are several recent attempts to improve NMS.
  - Merging or down-weighting instead of rejecting.
  - Integrate a differentiable version so that the entire pipeline is end-to-end trainable.
- From now on, we assume there is always NMS as the last post-processing stage after each detector.



© Navaneeth Bodla, Bharat Singh, Rama Chellappa, Larry S. Davis: Soft-NMS Improving Object Detection with One Line of Code ICCV 2017

Detection evaluation

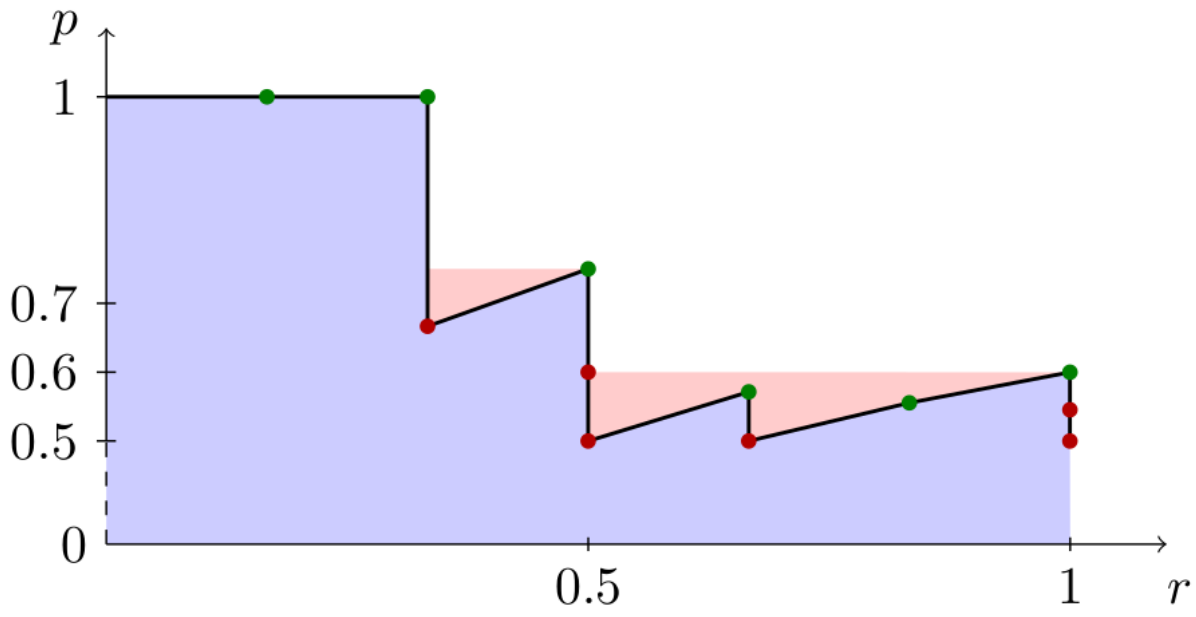
- For each image and for each class independently, rank predicted regions by descending order of confidence and assign each region r to the ground truth region $g^* = argmax_g\ J(r,g)$ of maximum overlap if $J(r,g^*) > \tau$ and mark it as **true positive**, else **false positive**.
- Each ground truth region can be assigned up to one predicted region.
- For each class independently, rank predicted regions of all images by descending order of confidence and compute **average precision (AP)** according to true/false labels.
- The **mean average precision (mAP)** is the mean of the AP values for all classes.
- Object detection models are also evaluated with different τ thresholds.
  - Compute mAP for multiple thresholds (e.g., mAP@0.5:0.95)

Average precision (AP)

The AP is a way to summarize the precision-recall curve into a single value representing the average of all precisions.

- Ranked list of items with **true**/**false** labels.
- # total ground truth n, current rank k, # true positive t
- **precision** $p = \frac{t}{k}$, **recall** $r = \frac{t}{n}$
- **average precision** = area under curve (filled-in curve)

1	2	3	4	5	6	7	8	9	10	11	12
T	T	F	T	F	F	T	F	T	T	F	F



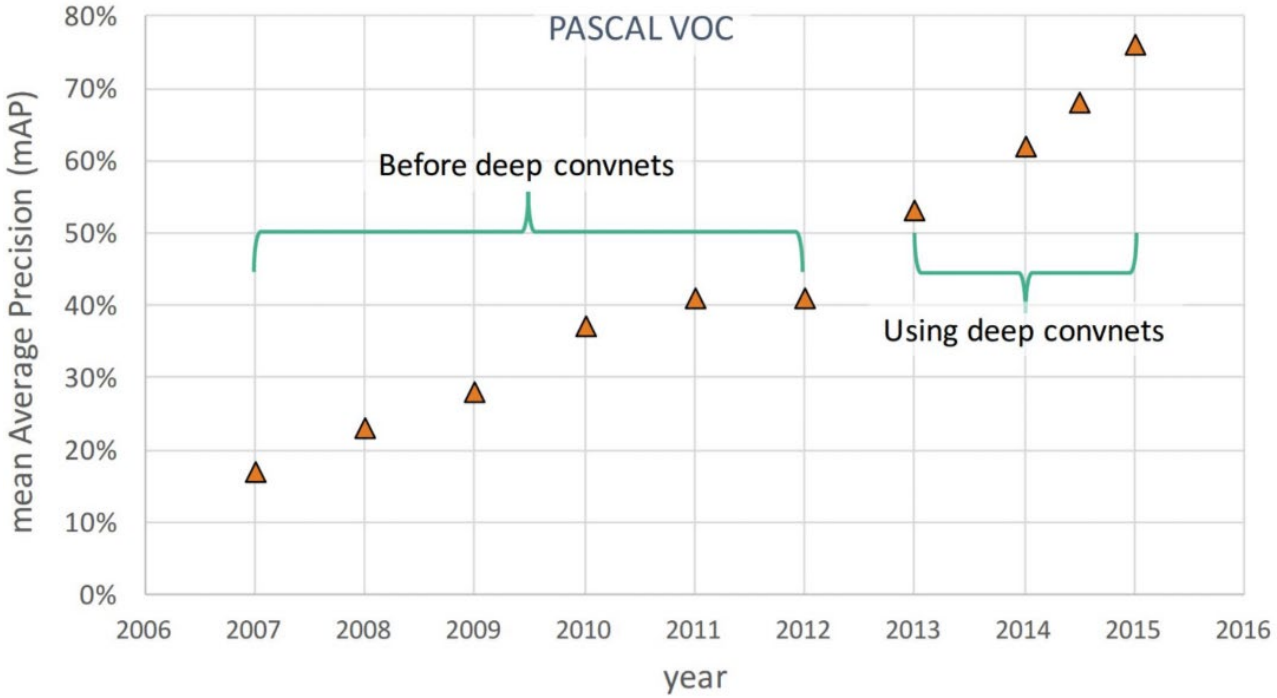
© Paul Henderson, Vittorio Ferrari: End-to-End Training of Object Class Detectors for Mean Average Precision ACCV 2016

Contents

- Traditional machine learning approaches
  - Sliding window
  - Non-maximum suppression
  - Detection metrics
- *Deep learning approaches*
  - *Two-stage detection*
  - *Single-stage detection*

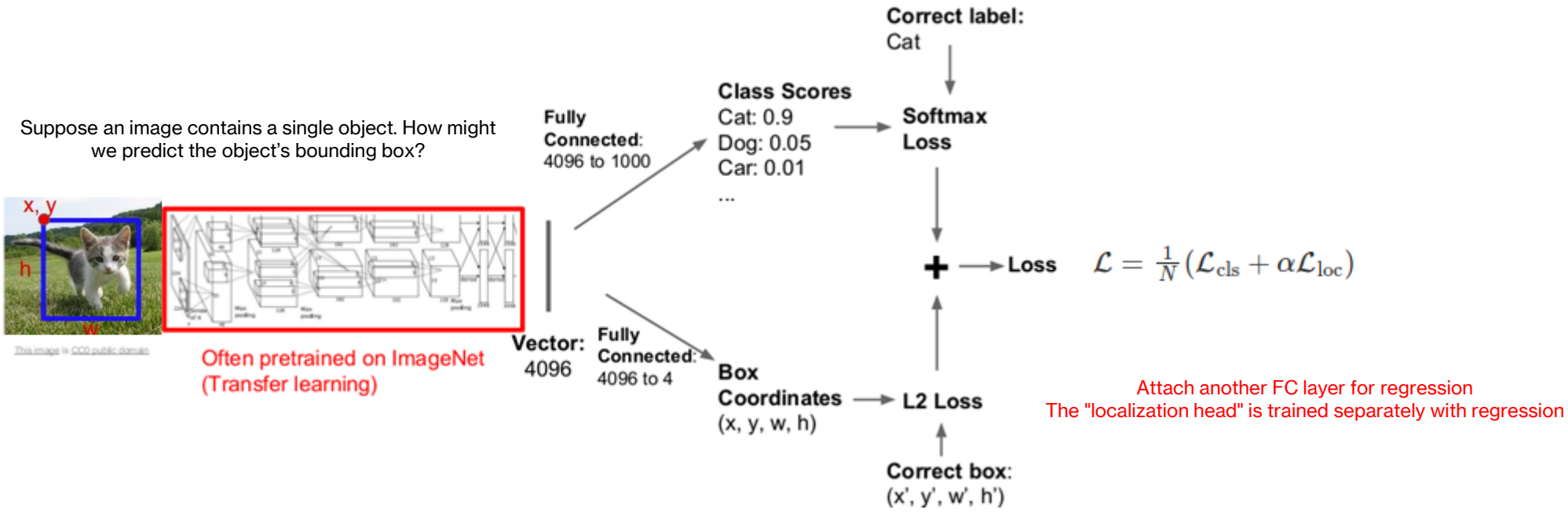
Impact of deep learning

- The timeline below depicts visually the increase in mAP score of object detection techniques from 2006 to 2016 when evaluated on the Pascal VOC dataset.



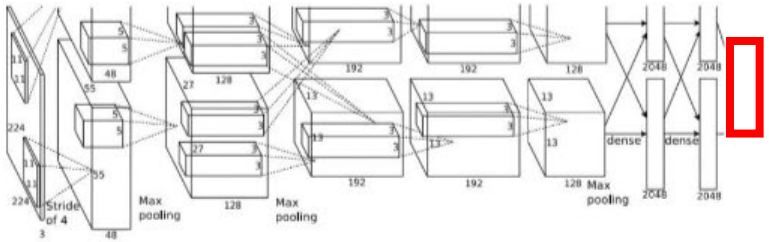
Localization as regression

- Single object per image.
- Compared to object recognition, in localization we predict coordinates of a bounding box (x, y, w, h).



Detection as regression

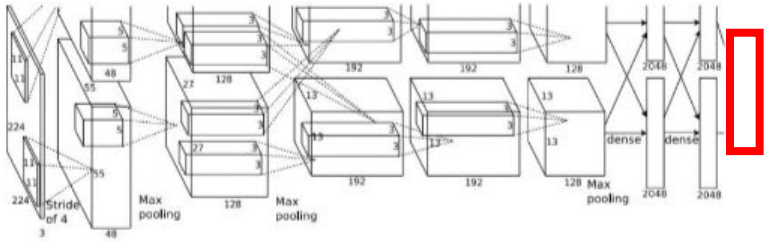
Each image need a different number of outputs!



CAT: (x, y, w, h)

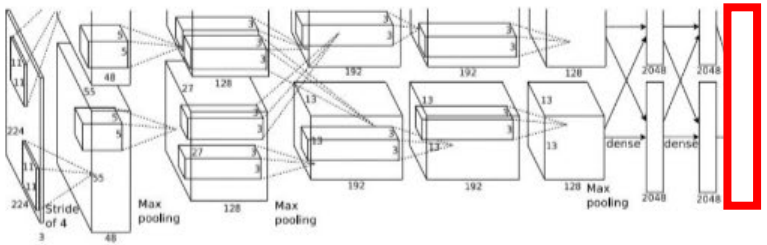
4 numbers

We don't know in advance the number of objects in the image



DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)

12 numbers

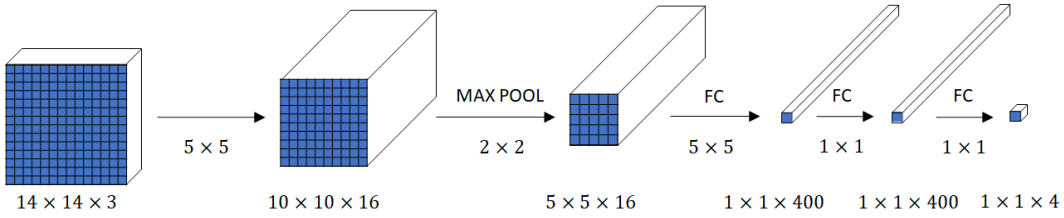


DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
....

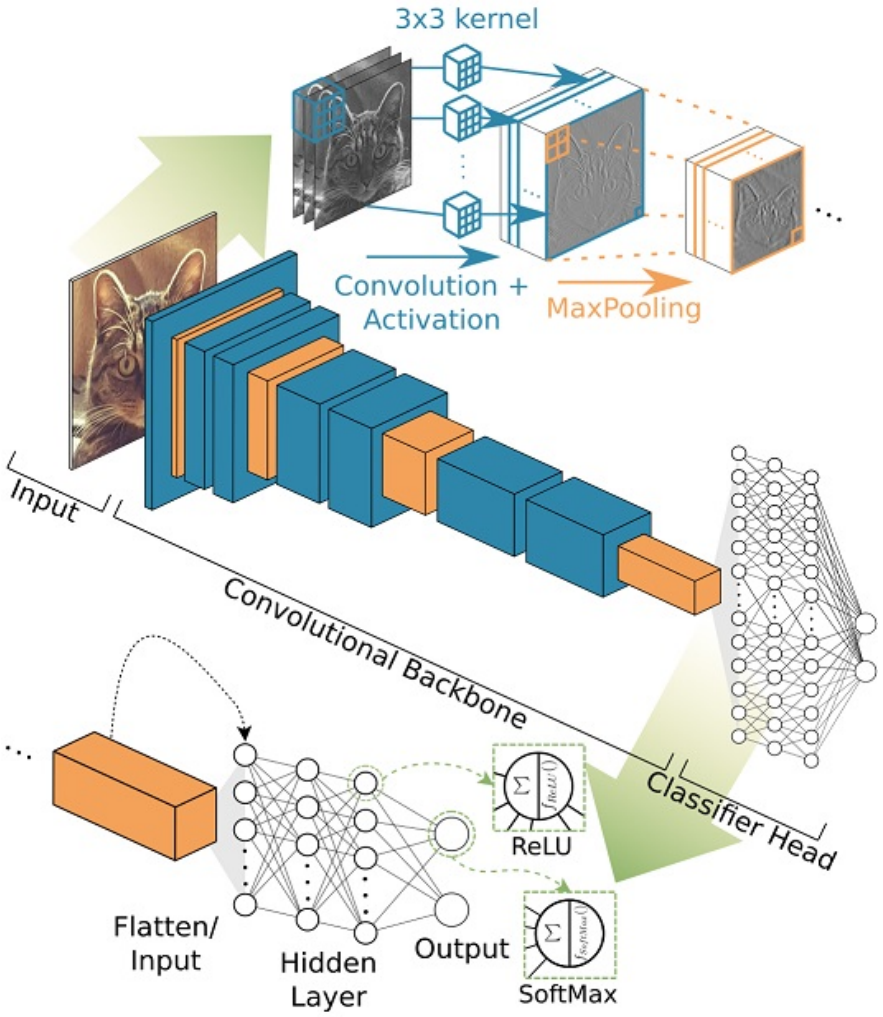
Many numbers

Detection as classification

- Sliding window idea to classify each crop as object or background.
- The following CNN produces only a single spatial output. It learns to identify whether a 14×14 image contains an object of interest or not.

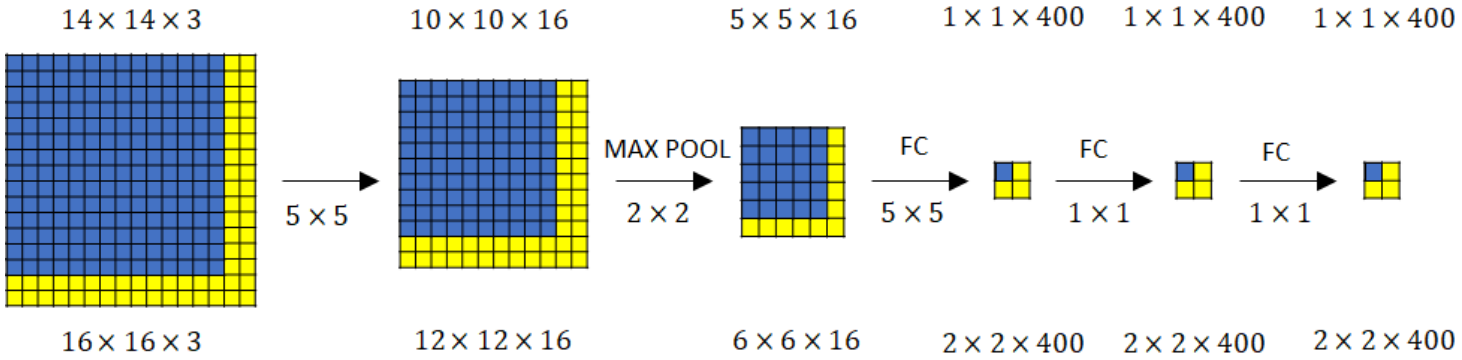


Example CNN (5×5 conv, no padding), (pool, /2), (fc 400), (fc 400), (fc 4)



Convolutional sliding window

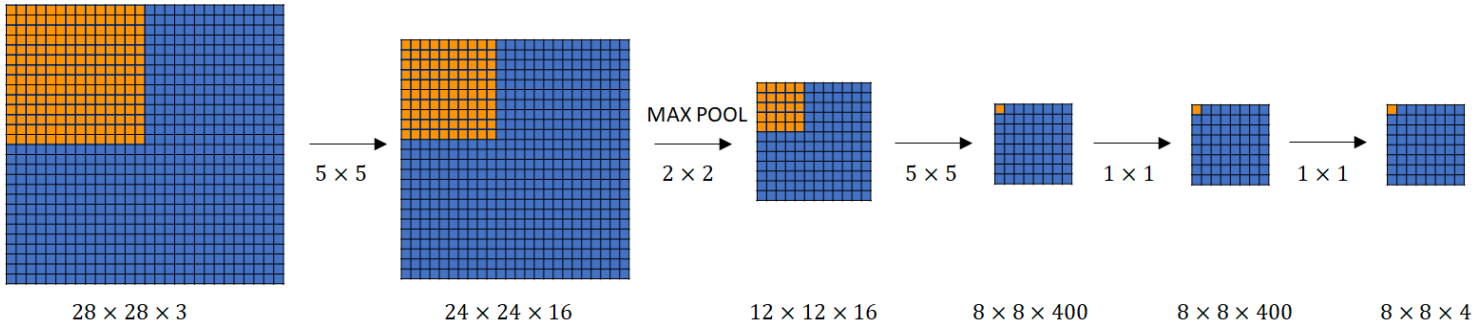
- CNNs are inherently efficient when applied in a sliding fashion because they naturally share computations common to overlapping regions.
- Previous FC layers are effectively replaced by convolutional operations, since all layers are applied convolutionally. When applying our network to larger images, we simply apply each convolution over the extent of the full image.
- In the example, the CNN output was a 2×2 matrix which decides the class of each window.



Convolutional sliding window approach (16×16×3 image + 14x14 kernel + stride 2 => 2×2 output = 4 windows)

Convolutional sliding window

- In the example, the CNN output was an 8x8 matrix which decides the class of each window.



Convolutional sliding window approach (28×28×3 image image + 14x14 kernel + stride 2 => 8×8 output = 64 windows)

- Using this convolutional implementation, we make all the predictions at the same time through this CNN instead of doing sliding window sequentially.
- This algorithm for object detection is computationally efficient but is not the most accurate one.

© Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, Yann LeCun: OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks ICLR 2014

Object's size

- The convolutional sliding window does not solve scale problems. Traditionally in computer vision, a popular technique for scale-invariant object detection is to utilize an **image pyramid**.
- This process is computationally intensive. With the advent of deep learning, the image pyramid was enhanced using the inherent multi-scale **pyramidal feature hierarchy** of the CNNs architecture, where the output size of feature maps decreases after each successive block of convolutional operations.

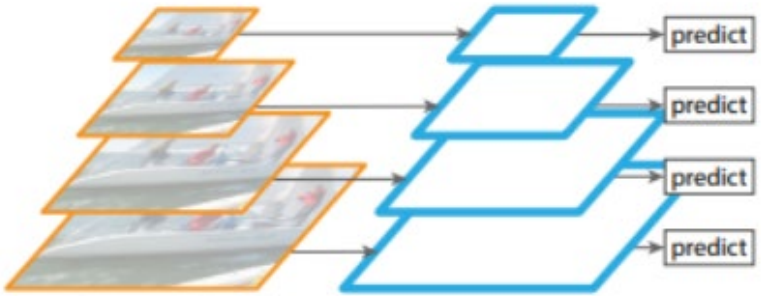
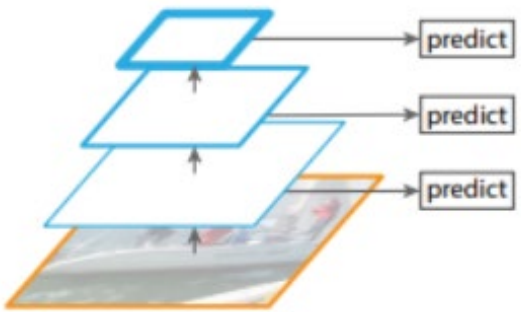


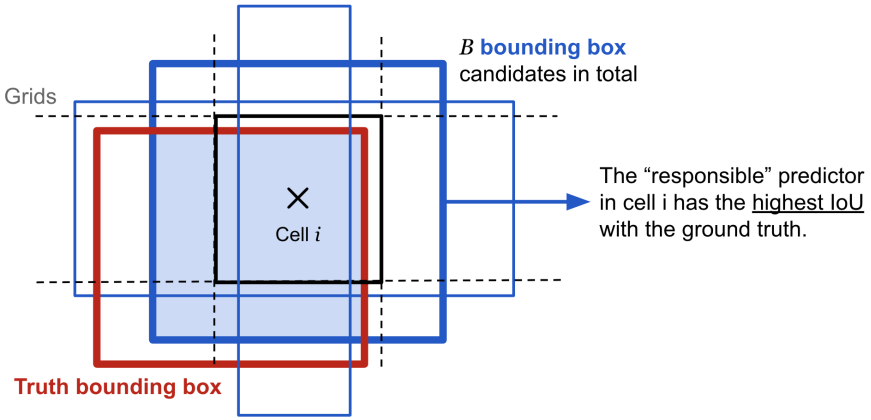
Image pyramid



Pyramidal feature hierarchy

Object's aspect ratio

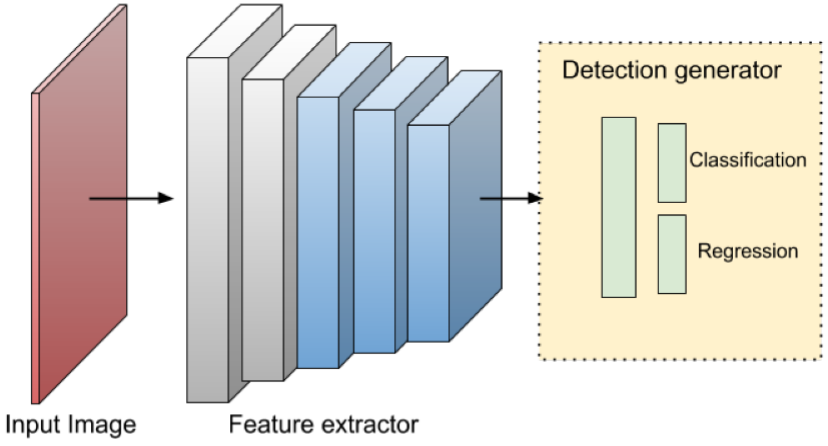
- Detection methods also use **anchors** to handle the aspect ratio problem. These anchors are basically predefined bounding boxes in different proportions to facilitate various kinds of objects and their size. However, using several anchors involves a lot of parameters.
- The network must predict the location, category and IoU for every tiled anchor box.



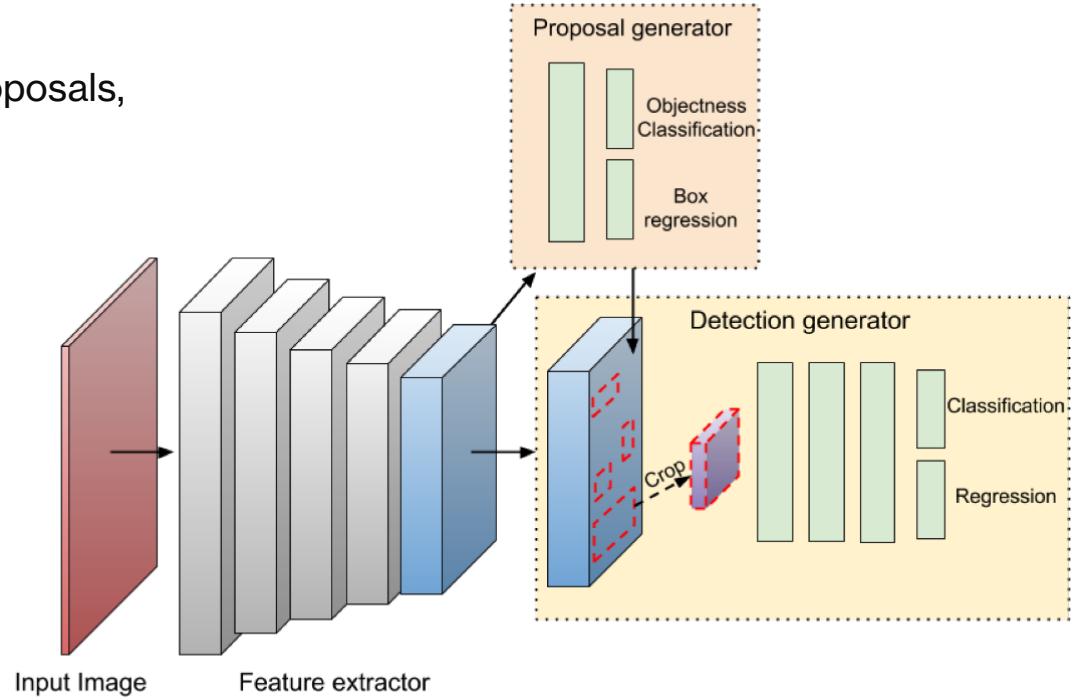
Anchors are defined to capture the scale and aspect ratio of specific object classes you want to detect and are typically chosen based on object sizes in the data set.

Detection using deep learning

- There are two types of mainstream models:
  - **Single-stage:** One stage models directly predict objects from an image in a single step.
  - **Two-stage:** Two-stage detectors first generate region proposals, then refine them for final detection.



Single-stage detection



Two-stage detection

Comparison: one vs two stage detectors

- There are two types of mainstream models:
  - **Single-stage:** e.g., SSD
  - **Two-stage:** e.g., Faster R-CNN, R-FCN

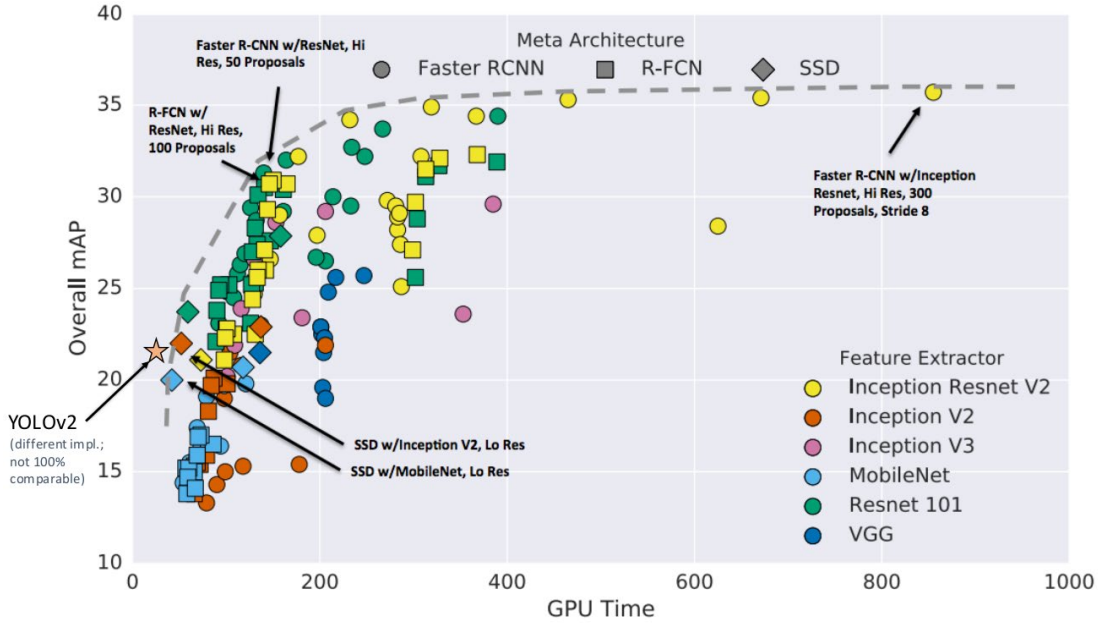
Base Network
VGG16
ResNet-101
Inception V2
Inception V3
Inception
ResNet
MobileNet

Object Detection architecture
Faster R-CNN
R-FCN
SSD

Image Size
 # Region Proposals

Takeaways
Faster R-CNN is slower but more accurate

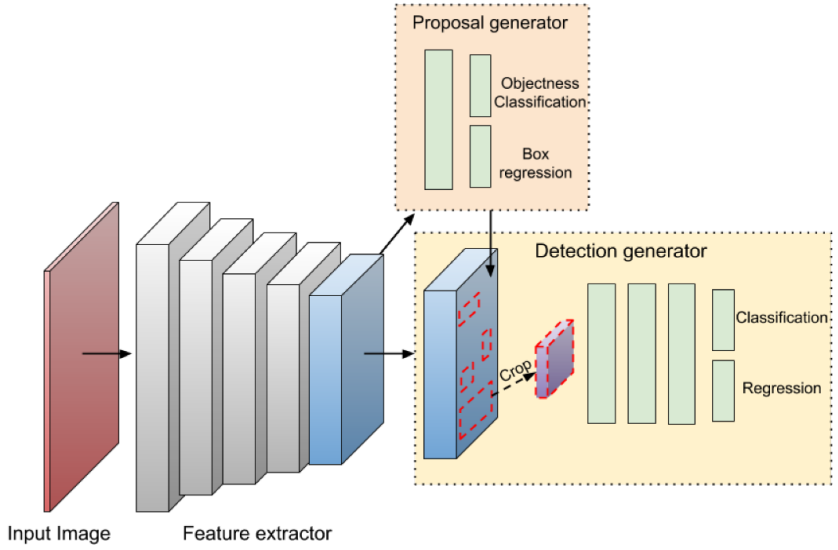
SSD is much faster but not as accurate



© Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, Kevin Murphy: Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors CVPR 2017

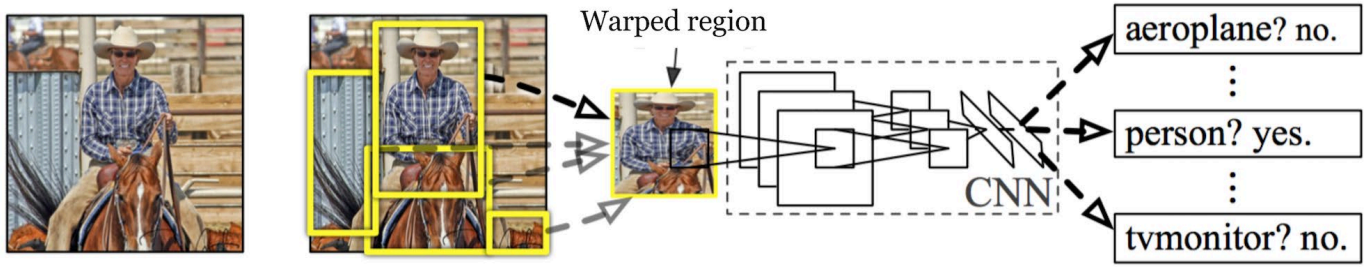
Two stage detection

- In two stage detectors, **dense** region proposals are filtered in the first stage and then **sparse** candidates are further regressed and classified in the second stage.
- These detectors filter out most of the negative proposals.
- Two-stage detectors usually achieve better accuracy but are slower than single-stage detectors.
  - R-CNN
  - Fast R-CNN
  - Faster R-CNN
  - R-FCN



R-CNN

- R-CNN is short for **Region-based Convolutional Neural Networks**.

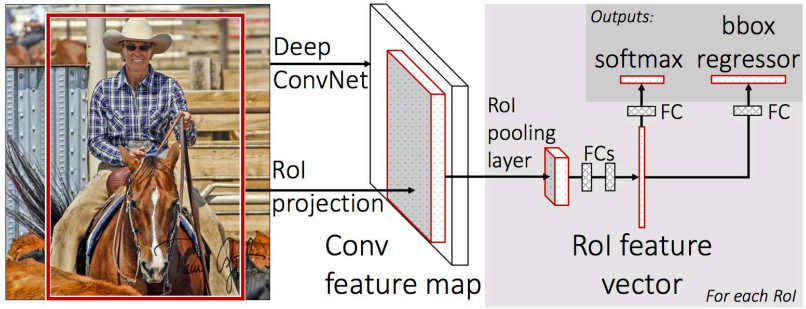


- First, using "Selective Search", it identifies up to ~2000 bounding box object region candidates.
- R-CNN uses AlexNet to extract features from selected and warped regions in 227x227 images. It generates a 4096-dimensional feature vector for each of the ~2000 cropped and resized images.
- The original R-CNN uses a SVM classifier for each category, that detects the presence or absence of an object belonging to a particular class.

© Ross B. Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik: Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation CVPR 2014

Fast R-CNN

- Instead of running the CNN ~2000 times per image, we can run it just once per image and get all the features of interest.
- Whereas the previous R-CNN detector must classify each warped region, Fast R-CNN pools CNN features corresponding to each proposed region. The **RoI Pooling layer** uses max pooling to convert the features in a region of interest into a feature map with fixed size dimension.
- Previous SVM can be replaced by a FC layer.



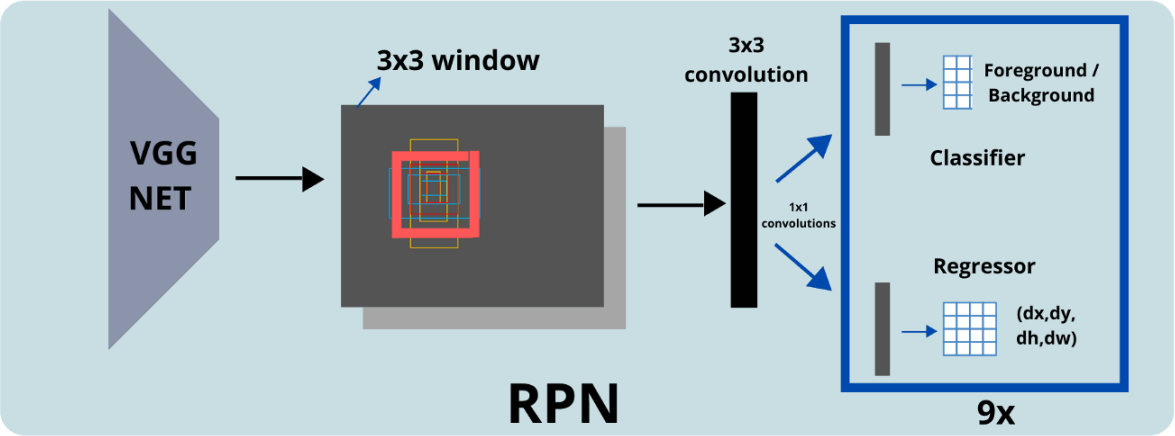
© Ross B. Girshick: Fast R-CNN ICCV 2015

Faster R-CNN

- Faster R-CNN is the enhanced version of Fast R-CNN.
- The key difference between them is that Fast R-CNN uses "Selective Search" for generating regions of interest, while Faster R-CNN uses a **Region Proposal Network**.

The RPN module **shares feature maps** with the CNN feature extractor and generates a set of object proposals, each with an objectness score as output.

The RPN **takes any sized input** to achieve these tasks. It consists of a 3×3 convolution on the feature map given by the backbone network and two 1×1 sibling convolution layers each for objectiveness score (classification) and offset regression.

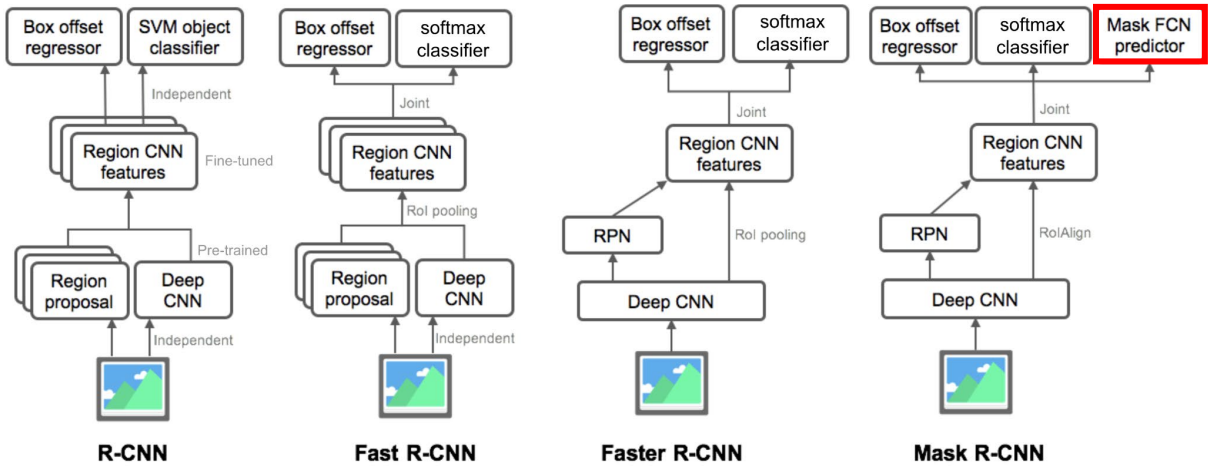


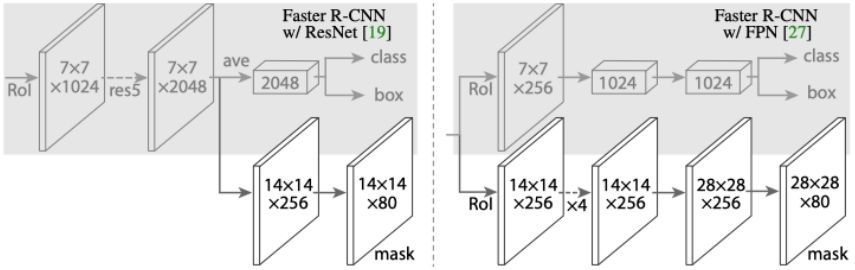
The RPN slides a small window over the output of the last layer of the feature map. For each window a set of B=9 anchors are generated with three scales and aspect ratios respectively.

© Shaoqing Ren, Kaiming He, Ross B. Girshick, Jian Sun: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks NIPS 2015

Mask R-CNN

- Mask R-CNN extends such techniques to go one step further and locate exact pixels of each object instead of only bounding boxes by adding a branch to the Faster R-CNN that outputs a binary mask that says whether a given pixel is part of an object (**instance segmentation**).
  - Add one branch that predicts a segmentation mask within each RoI using a ResNet-C4/FPN backbone and an average binary cross-entropy loss function



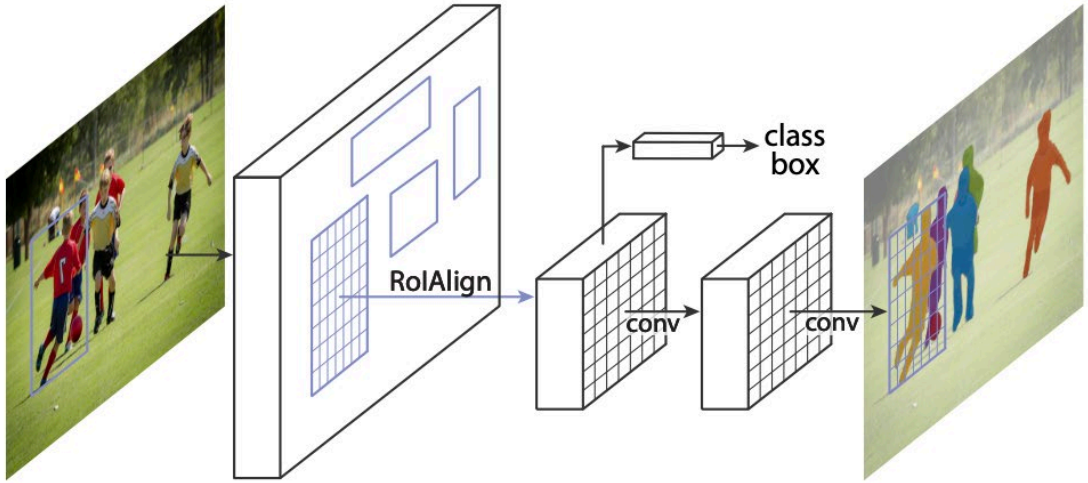


Faster R-CNN can support various backbones but the head with the FPN backbone is more efficient (fewer filters)

© Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross B. Girshick: Mask R-CNN ICCV 2017

Mask R-CNN

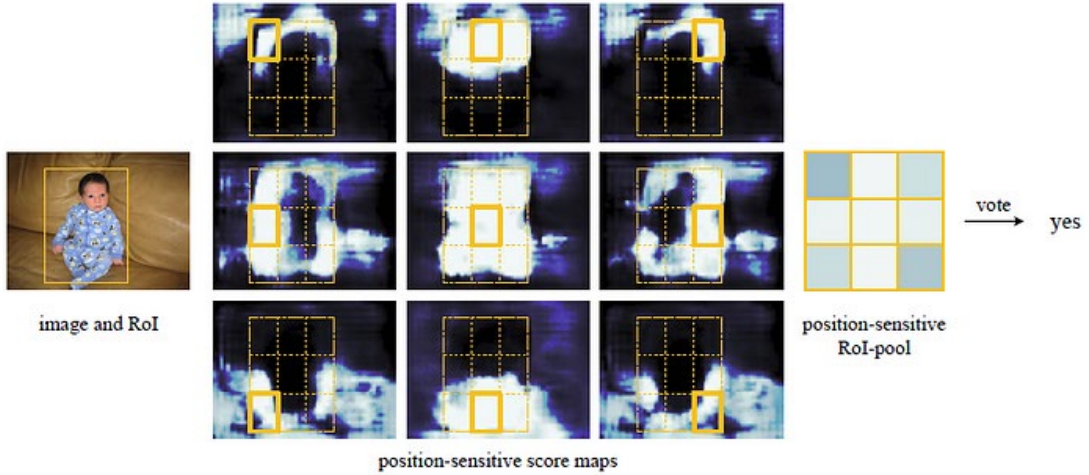
- The **RoI Align layer** was proposed to address misalignment introduced by quantization's (i.e., ceil, floor) operations during pooling.
  - RoI Pooling quantize the original RoI and cause misalignments from the original pixel locations

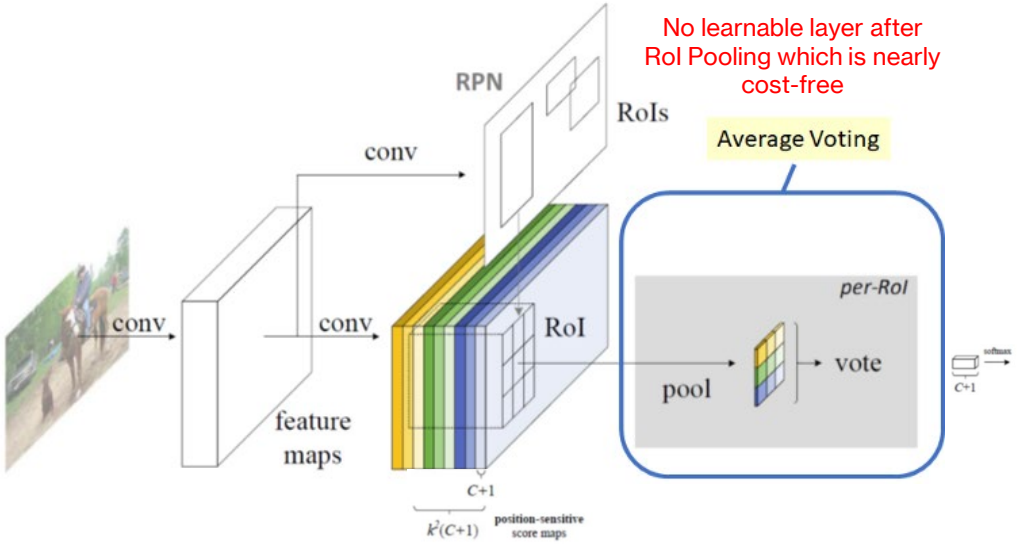


© Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross B. Girshick: Mask R-CNN ICCV 2017

R-FCN

- In previous R-CNN family, each detector applies the final FC layers to make predictions for each RoI.
  - The process after RoI Pooling (FC layers) does not share information among RoIs and takes time
- **Region-based Fully Convolutional Networks** removes FC layers. All major complexity is moved before RoI Pooling to generate score maps. Using k=3, we generate 9 position-sensitive score maps instead of only 1 feature map per category.

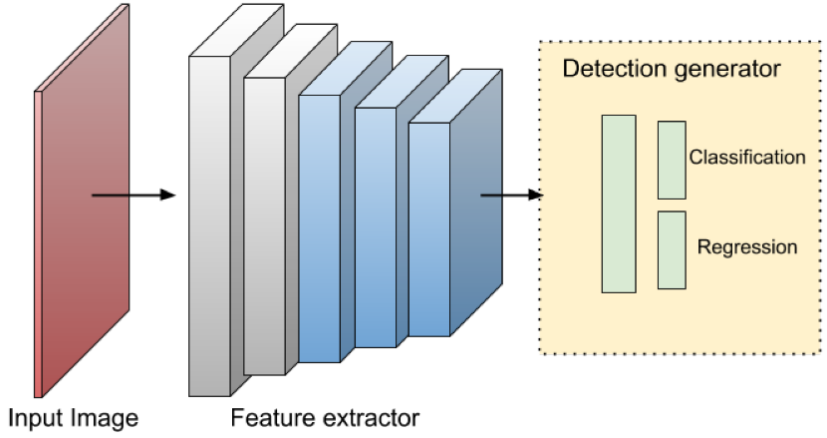




© Jifeng Dai, Yi Li, Kaiming He, Jian Sun: R-FCN: Object Detection via Region-based Fully Convolutional Networks NIPS 2016

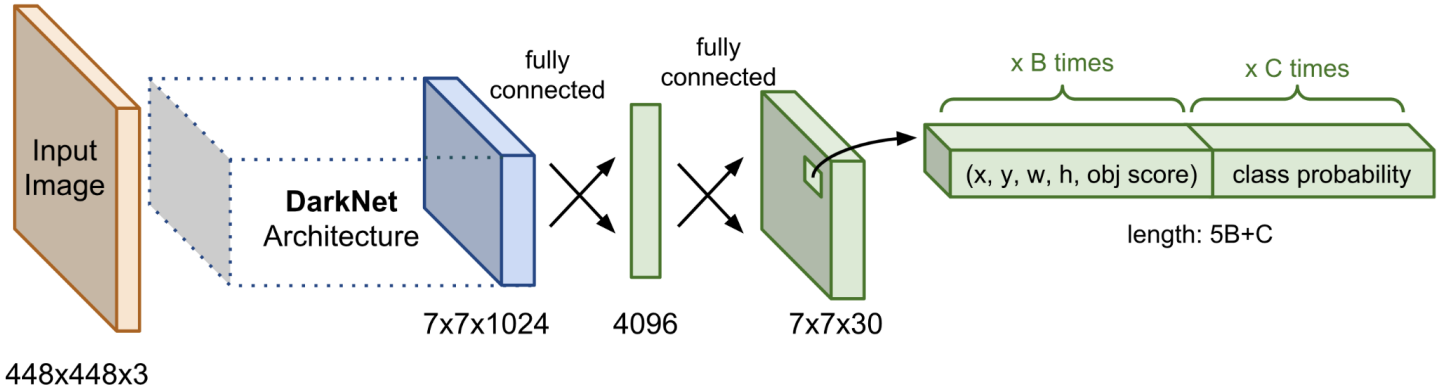
One stage detection

- Both tasks, class classification and bounding box regression, are done directly without using any region proposal typically from **dense** candidates.
- Directly face all the regions on the image and have a problem of class imbalance if no specialized design is introduced.
- Typically, this strategy is highly regarded for its speed, but the downside is its high memory cost and lower detection accuracy.
  - YOLO
  - SSD
  - RetinaNet
  - DETR



YOLO

- The well-known "You Only Look Once" model was the first detection model to combine the problem of drawing bounding boxes and identifying class labels in an end-to-end network.
- The YOLO family (e.g., YOLOv1, YOLOv2, YOLOv3, etc.) has continued to evolve since the initial release in 2016.
  - YOLOv1 feature extractor consists of DarkNet, that consists of 24 convolutional layers followed by 2 FC layers.
  - YOLOv2 uses DarkNet-19, where FC layers are removed to make it fully convolutional. YOLOv3 uses DarkNet-53.

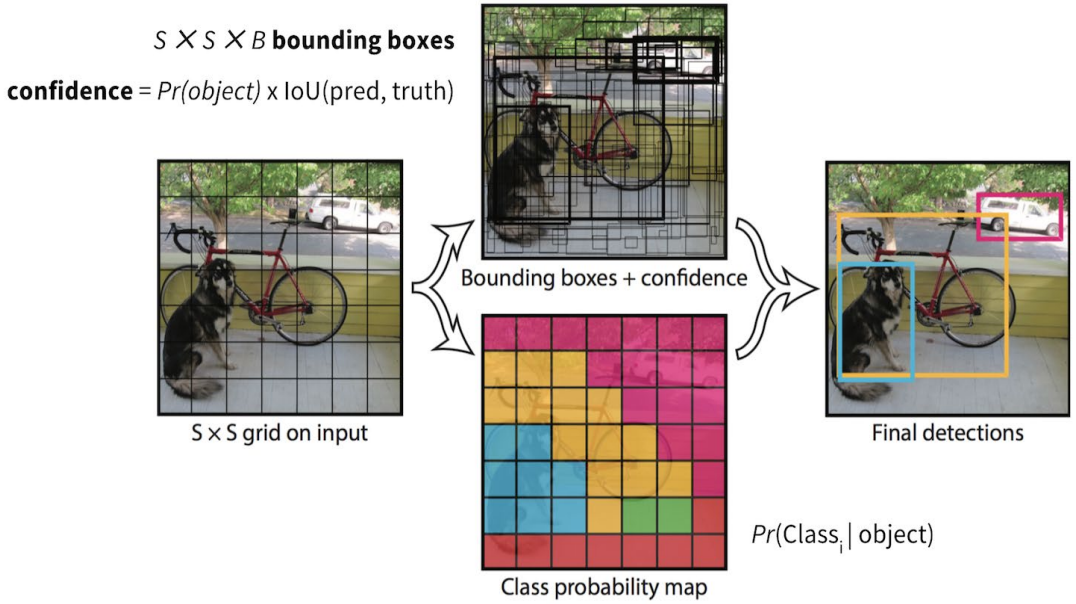


The output feature maps indicate that YOLO has conceptually divided the input image into 7×7 grid cells. In each cell the depth is 30 since B=2 (anchors) and C=20 (categories).

YOLO

- The input image is divided into S×S grid cells. Each cell can predict only one object. When the center of a ground truth object falls in a particular cell, it is responsible for predicting that object.
- Each cell regresses B anchors along with the confident scores of those boxes, and a probability of C classes conditioned on the existence of an object in the bounding box.
- For each cell we compute an element wise product between objectness confidence and predicted probabilities of each class.

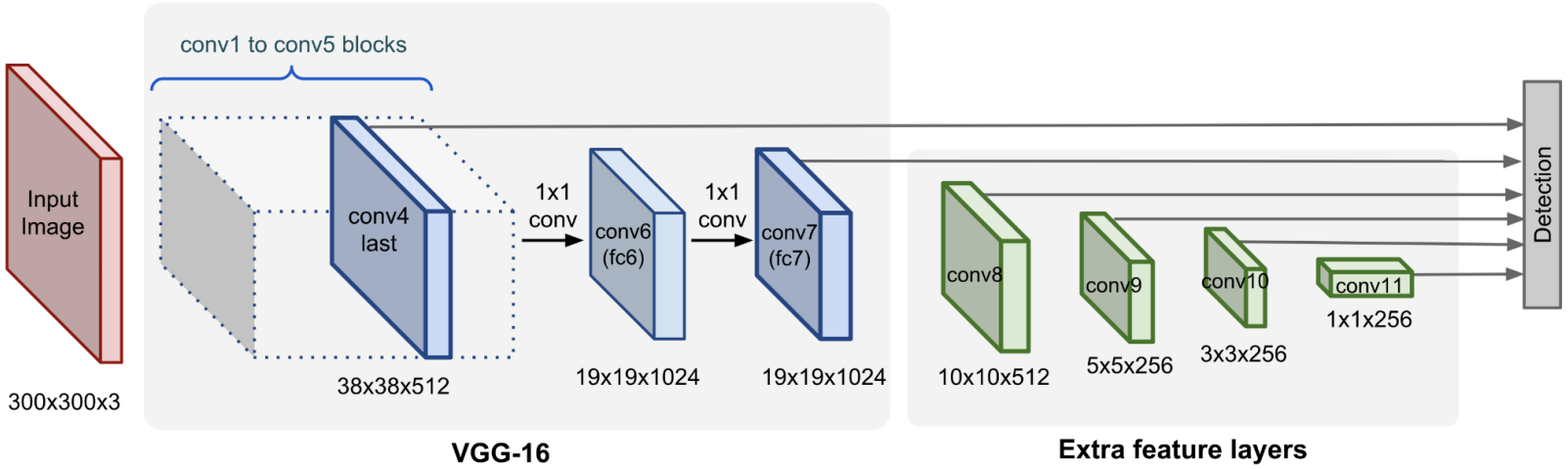
Extremely fast but got troubles localizing small objects



© Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, Ali Farhadi: You Only Look Once: Unified, Real-Time Object Detection CVPR 2016

SSD

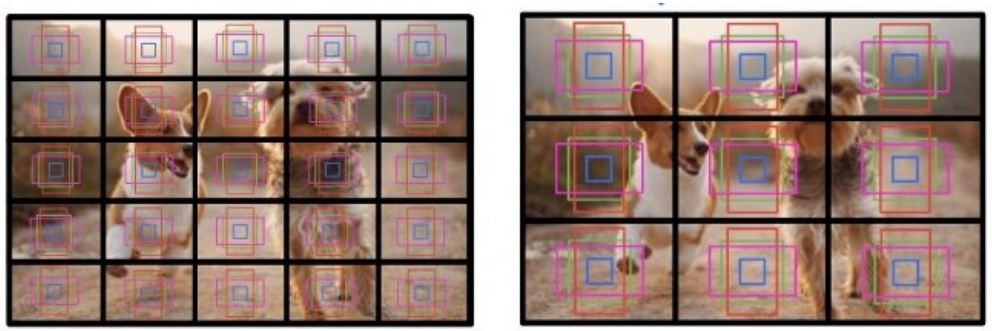
- The Single Shot Detector was one of the first attempts at using pyramidal feature hierarchy for efficient multi-scale object detection. To produce detections of different scales, the SSD network uses feature maps from different layers of a VGG-16 network.



The feature maps at the beginning of the network allows for detections of small objects while those at the end of the network allows for detections of large objects.

SSD

- There are anchor boxes associated with each cell of a feature map and the number of anchors (e.g., B=4, B=6).



An example of how the anchor box size is scaled up at conv9 and conv10 layers.

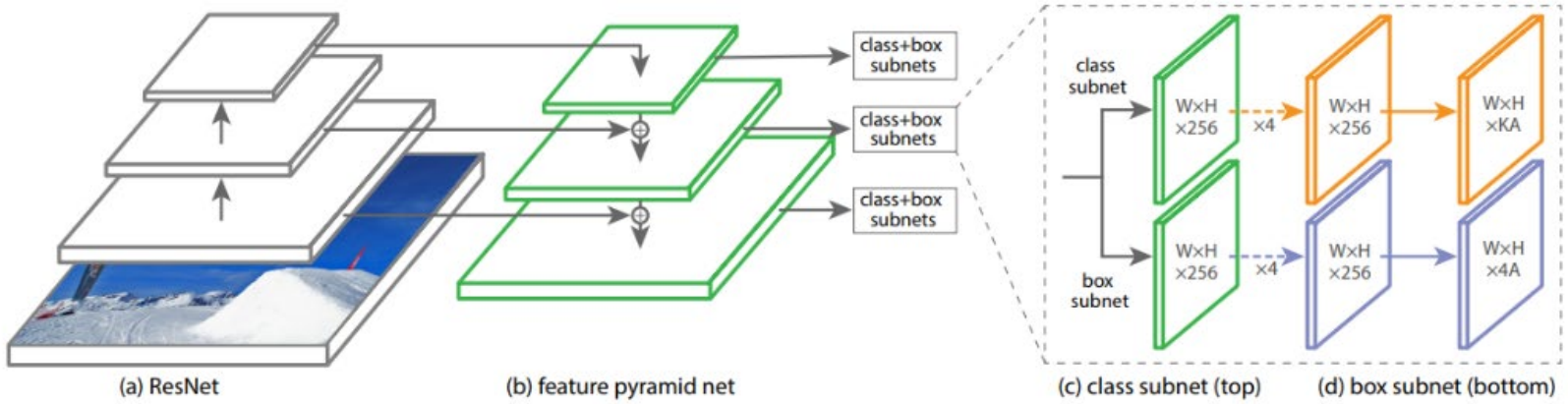
Pyramid starts at low resolution: difficulty with small objects

- If you want better accuracy (especially for small objects) but still fast (~30 FPS), although slower than YOLO v1 (~45 FPS).

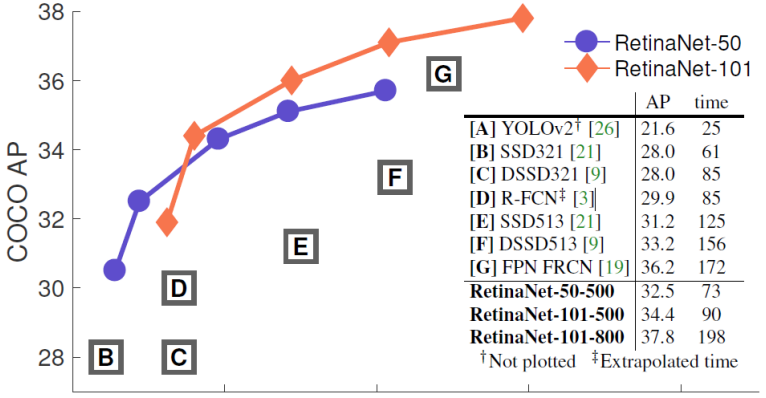
© Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, Alexander C. Berg: SSD: Single Shot MultiBox Detector ECCV 2016

RetinaNet

- RetinaNet is one of the best one-stage object detection models that has proven to work well with dense and small-scale objects.



RetinaNet uses the Feature Pyramid Network (FPN) on top of the ResNet as a backbone network to generate a rich convolutional feature pyramid.



	AP	time
[A] YOLOv2† [26]	21.6	25
[B] SSD321 [21]	28.0	61
[C] DSSD321 [9]	28.0	85
[D] R-FCN‡ [3]	29.9	85
[E] SSD513 [21]	31.2	125
[F] DSSD513 [9]	33.2	156
[G] FPN FRCN [19]	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	37.8	198

†Not plotted ‡Extrapolated time

Outperforms most one-stage and two-stage detectors

RetinaNet

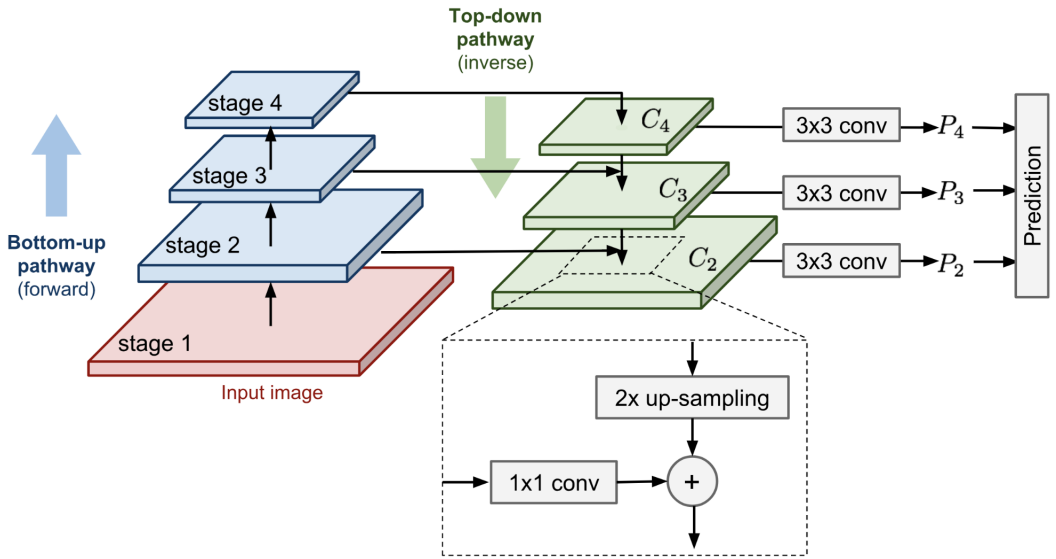
- RetinaNet makes two improvements over previous single stage object detection models:
  - FPN

The **Feature Pyramid Network** consists of a bottom-up pathway and a top-down pathway.

The bottom-up is the usual CNN for feature extraction (e.g., ResNet).

Some recent research adopted a similar encoder-decoder architecture with skip connections (e.g., U-Net, SegNet), which were quite popular. The main difference with these networks is that there are multiple prediction layers, instead of one at the last stage.

The P2+P3+P4 feature maps are used to generate final classification and regression scores (e.g., {P2, P3, P4, P5} in a ResNet).



An encoder-decoder architecture with lateral connections is developed for building high-level semantic feature maps at all scales.

© Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, Serge J. Belongie: Feature Pyramid Networks for Object Detection CVPR 2017

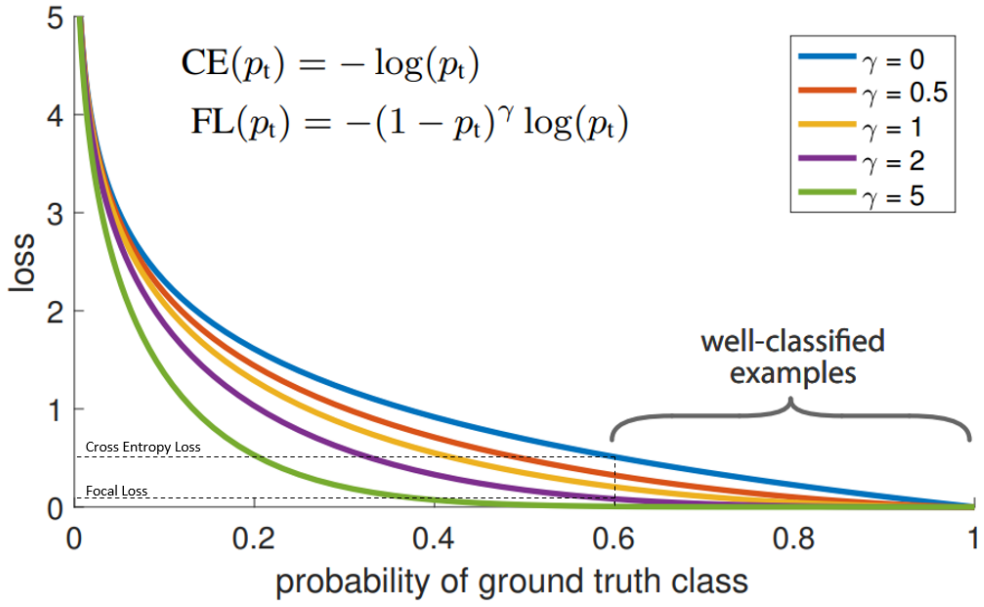
RetinaNet

- RetinaNet makes two improvements over previous single stage object detection models:
  - Focal loss

Focal Loss (FL) is introduced to handle the extreme foreground-background class imbalance problem between background that contains no objects, and foreground that holds objects of interest.

$$FL(p_t) = -\alpha(1-p_t)^{\gamma}\ln(p_t)$$

- No learning problem: Inefficient training since most locations in the image are easy negatives. The weighting parameter (α) is usually the inverse class frequency.
- Cumulative effect problem: Easy negatives, although results in small loss values, can collectively overwhelm the overall loss. The modulating factor (γ) decrease their contribution to the loss.

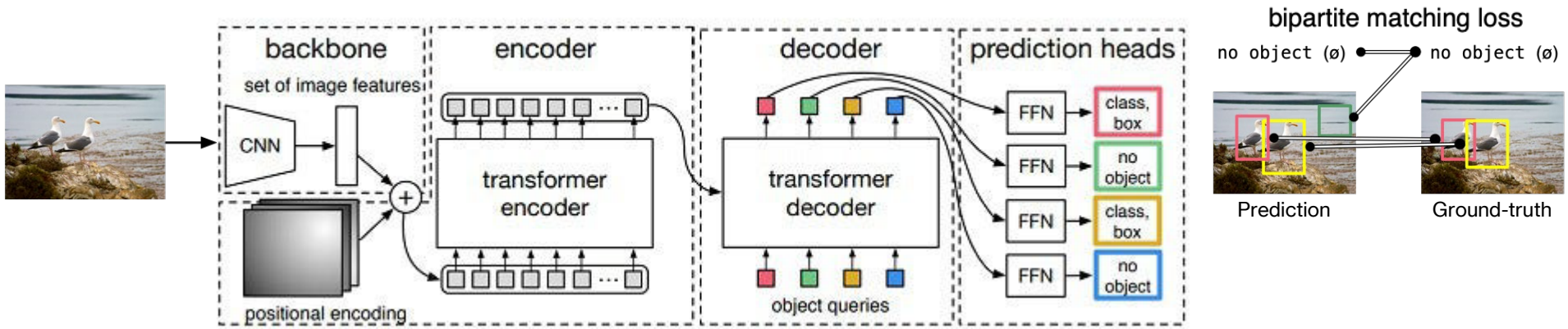


Comparing Focal Loss γ=2 (purple curve) to Cross Entropy Loss (blue curve).

© Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, Piotr Dollár: Focal Loss for Dense Object Detection ICCV 2017

DEtection TRansformer

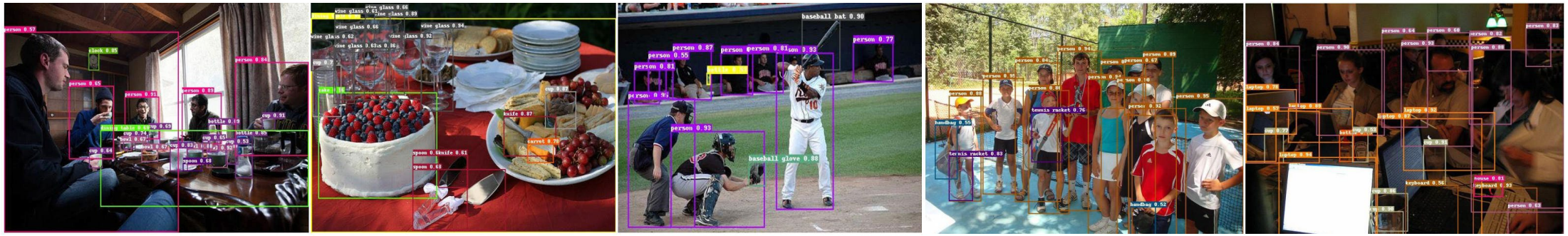
- DETR replaces the traditional object detection pipeline with self-attention mechanisms.
  - Like traditional detectors, DETR starts with a CNN backbone to extract feature maps
  - The extracted feature vectors are passed through am encoder that learns global relationships using self-attention
  - A fixed number of object queries (e.g., 100) are passed through the decoder, learning to predict one object per query, whether they exist or not.
  - It matches predictions with ground truth objects using Hungarian matching (an optimal assignment algorithm).



© Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov: End-to-End Object Detection with Transformers ECCV 2020

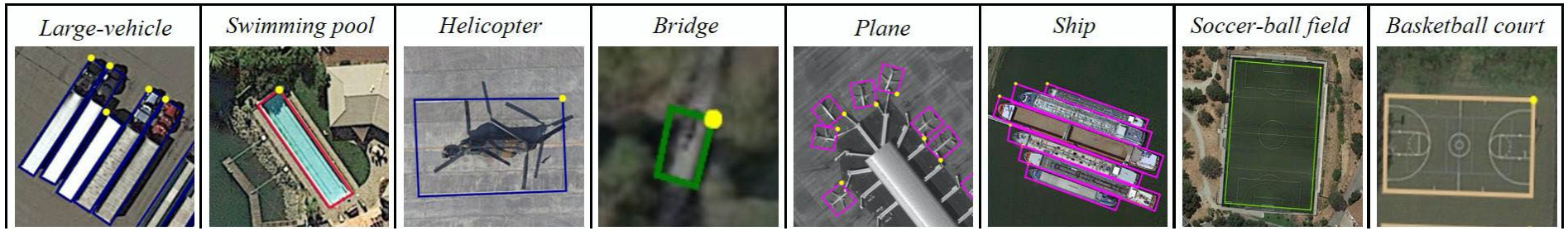
DEtection TRansformer

- DETR simplifies the detection pipeline by dropping spatial anchors or non-maximum suppression.
- Better performance on large objects but lower on small objects.
- Slow training convergence, requires lot of data for training.
- DETR is great for complex scenes where global context matters (e.g., crowded environments, autonomous driving).
- If you need real-time performance, YOLO is still the best option.



Future work

- Improving efficiency and speed.
  - DETR-like models require high computational resources due to transformers' quadratic complexity.
- Enhancing small object detection.
  - Future models could integrate hierarchical multi-scale attention to better detect tiny objects.
- Handling overlapping objects is still a challenge.



- Training on limited labeled data is an open challenge.
  - Future object detectors should work like GPT for text - capable of detecting any object without retraining.

© Gong Cheng, Xiang Yuan, Xiwen Yao, Kebing Yan, Qinghua Zeng, Xingxing Xie, Junwei Han: Towards Large-Scale Small Object Detection: Survey and Benchmarks PAMI 2023