Generic Object Detection

From image classification to object detection

Adrien Heinzlé
Eca Robotics

December 9, 2021

Part I

Introduction

Motivation

- Object detection framework are built upon traditionnal Convolutional Neural Networks to extract features from images.
- They use use Region of Interest algorithm to find objects of interest.
- For any remainder on convolution operations please refer to [1]

Take away

- Object Detection is one of the most fundamental and challenging problem in computer vision.
- It seeks to locate object instances from a large number of predefined categories in natural images.
- Since 2012 Deep Learning techniques have emerged as a powerful strategy for learning feature representations directly from data.

Goal

1. Determine wheter there are any instances of objects from given categories (humans, cars, boats, ...) in an image.
2. Return spatial location and extent of each object instance, via bounding boxes for example.
3. Basis for solving tasks such as scene understanding, object tracking, image captioning, activity recognition.

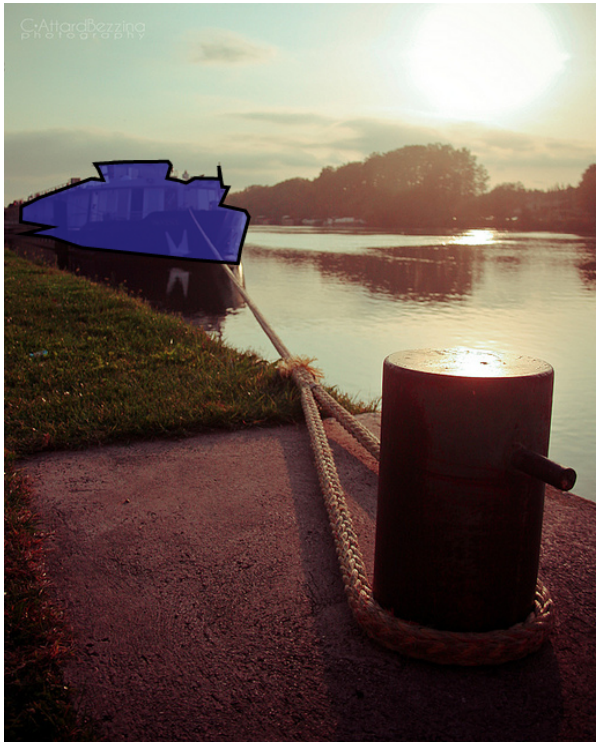


Figure: Detection example from COCO dataset [7].

Two types of object detection

Figure: Matching *particular* objects versus detecting object *categories* in general. Credit: *[8]*

The emergence of Deep Learning for Computer Vision

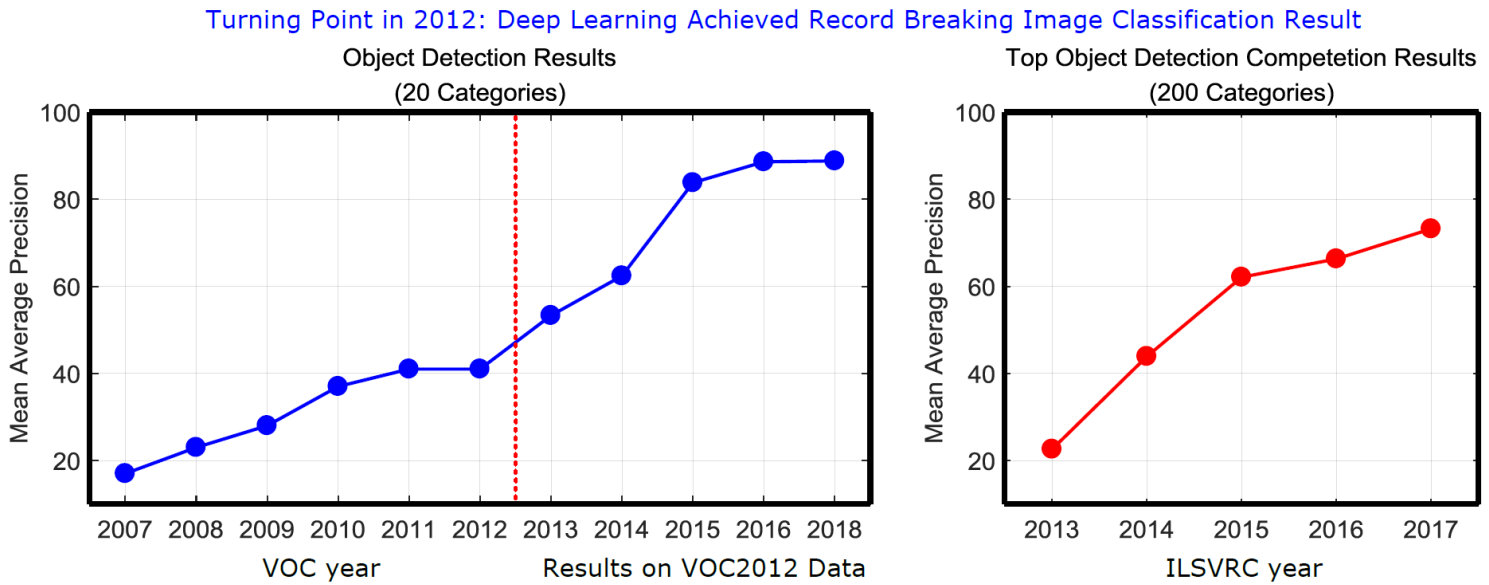


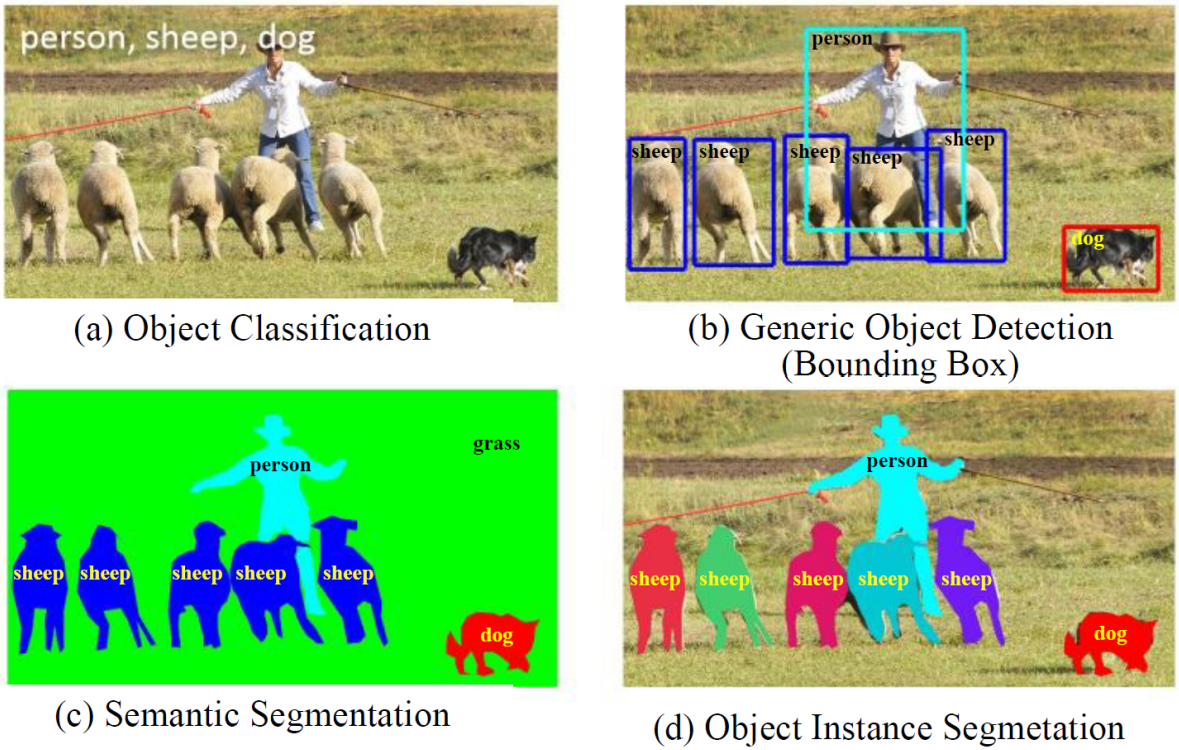
Figure: 2012 - turning point. Credit: *[8]*

- Detection results of winning entries in the VOC2007-2012 competitions
- Top object detection competition results in ILSVRC2013-2017.

Part II

Generic Object Detection

There are many problems closely related to that of generic object detection but we can isolate four main goals.



(a) Object Classification

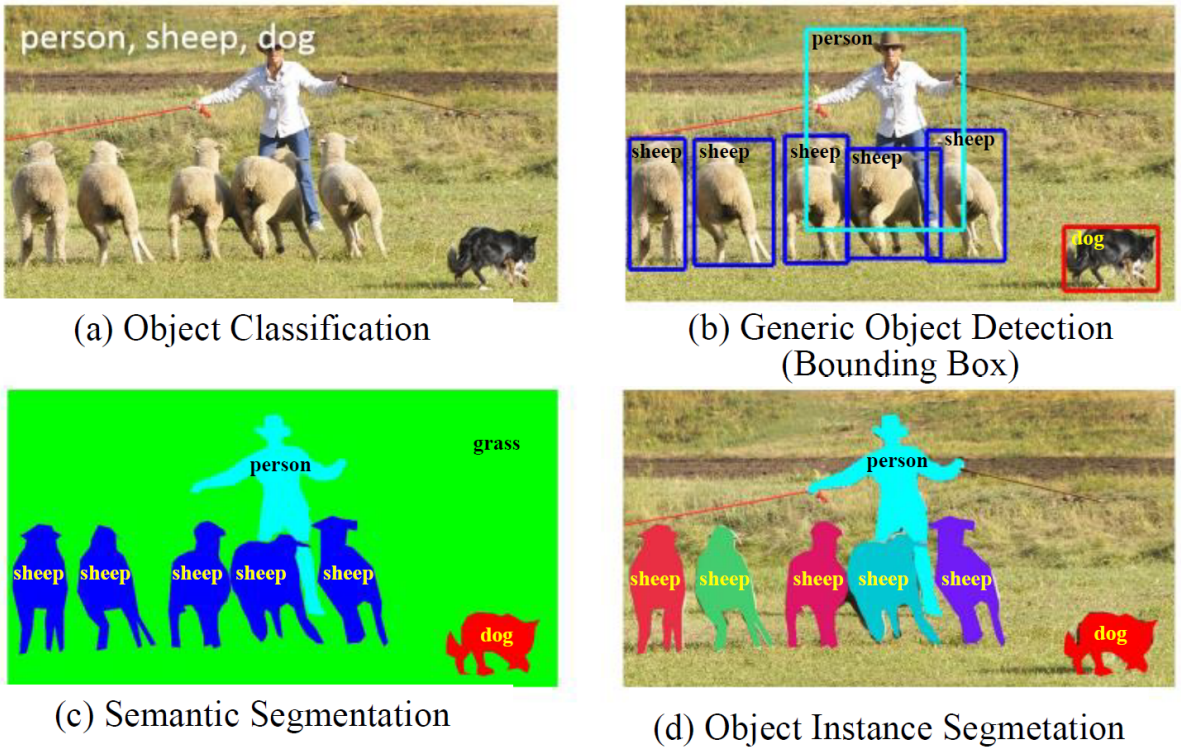
(b) Generic Object Detection (Bounding Box)

(c) Semantic Segmentation

(d) Object Instance Segmetation

Figure: generic object detection Credit: *[8]*

(a) *object classification*: assess the presence of objects from a given number of object classes in an image.



(a) Object Classification

(b) Generic Object Detection (Bounding Box)

(c) Semantic Segmentation

(d) Object Instance Segmetation

Figure: generic object detection Credit: *[8]*

(b) and (c) *Generic object detection* is closely related to *semantic image segmentation*: aims to assign each pixel in an image to a semantic class label.

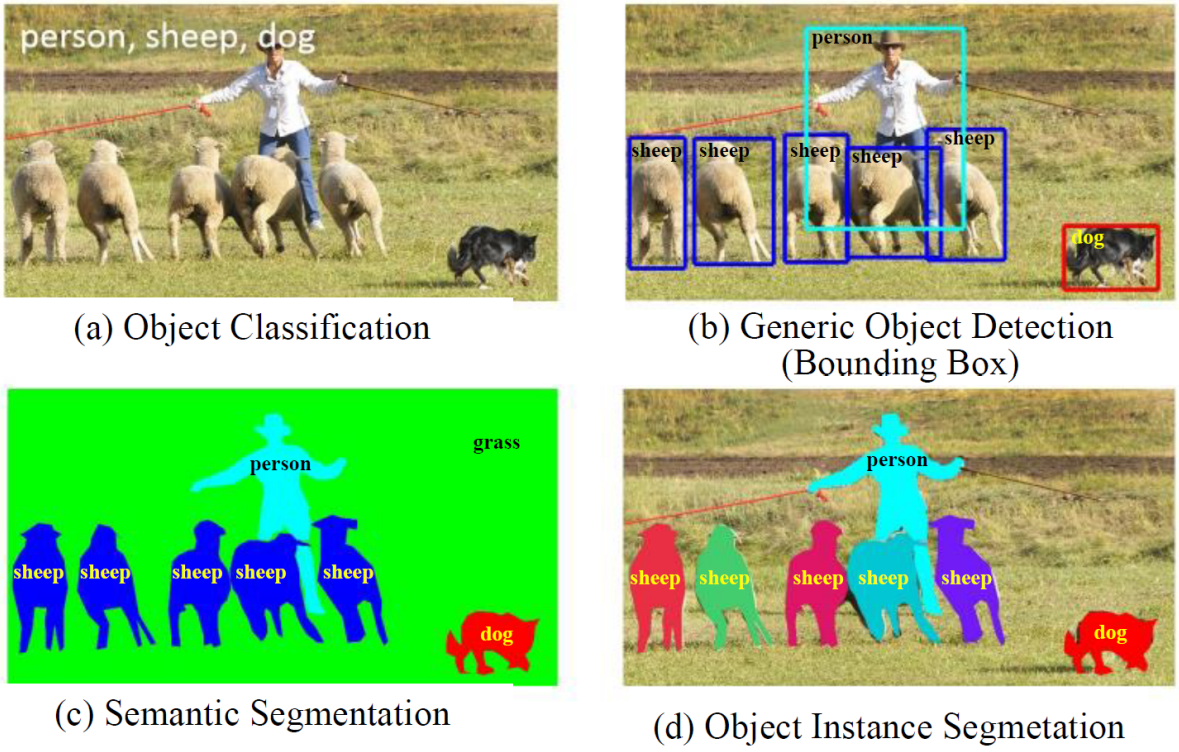


Figure: generic object detection Credit: *[8]*

(d) *Object instance segmentation*: aims to distinguish different instances of the same object class.

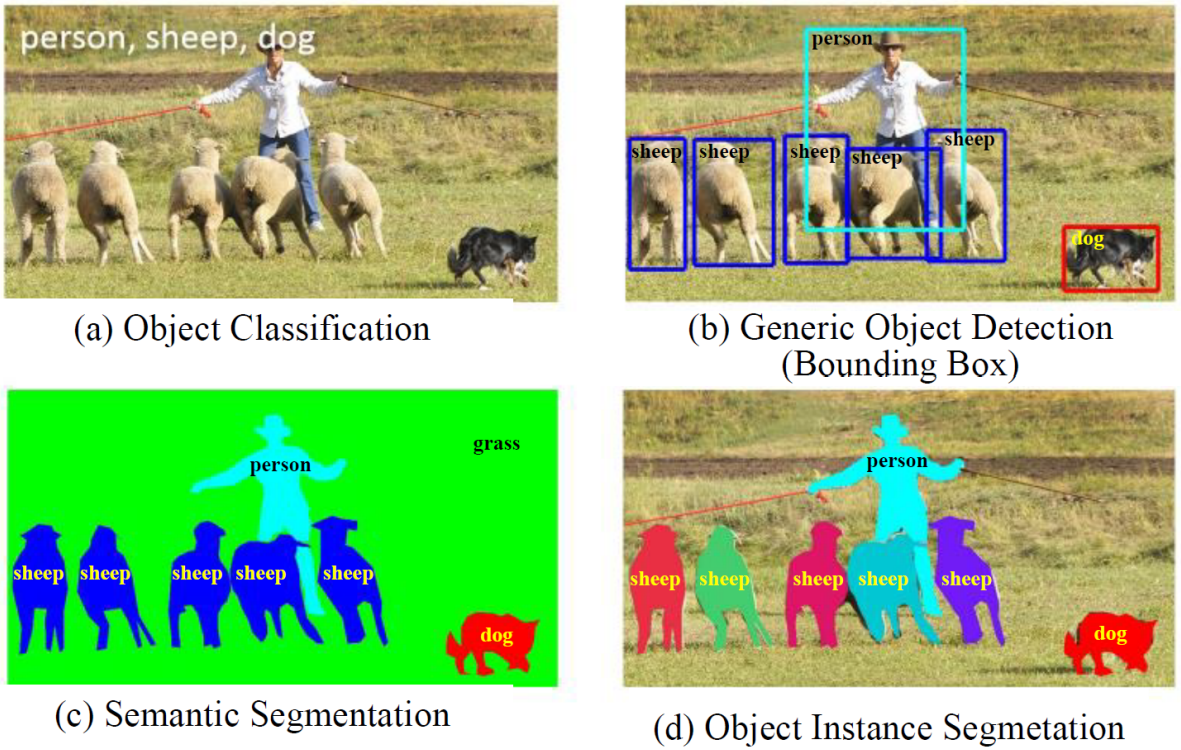


Figure: generic object detection Credit: *[8]*

- *Distinctiveness*: accurately localize and recognize objects in images or video frames.
- *Robustness*: object instances from the same category, subject to intra-class appearance variations, can be localized and recognized.
- *Efficiency*: entire detection task runs in real time with acceptable memory and storage demands.

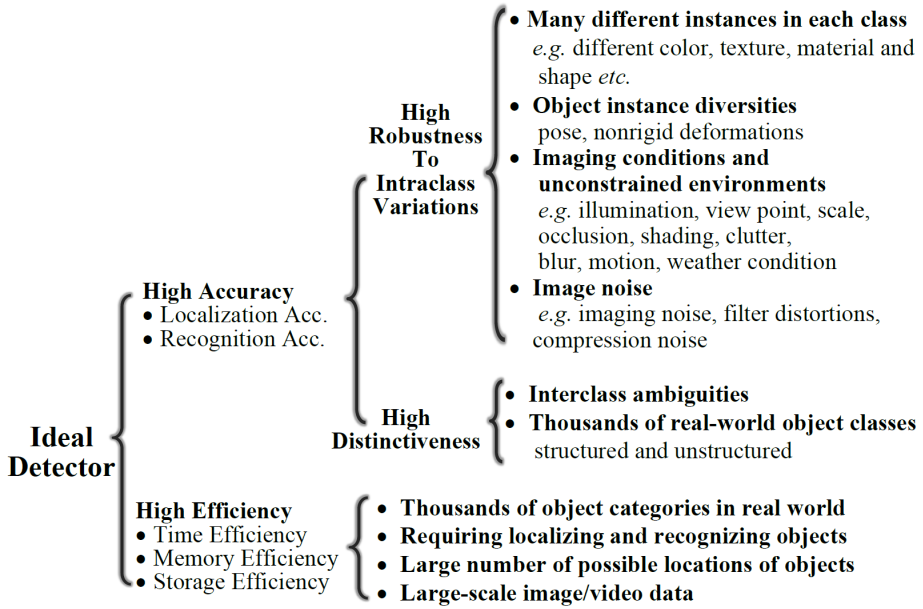


Figure: Taxonomy of challenges in generic object detection. Credit: *[8]*

Figure: (a-j) Intra-class appearance variations. (j) interclasses. Credit: *[8]*

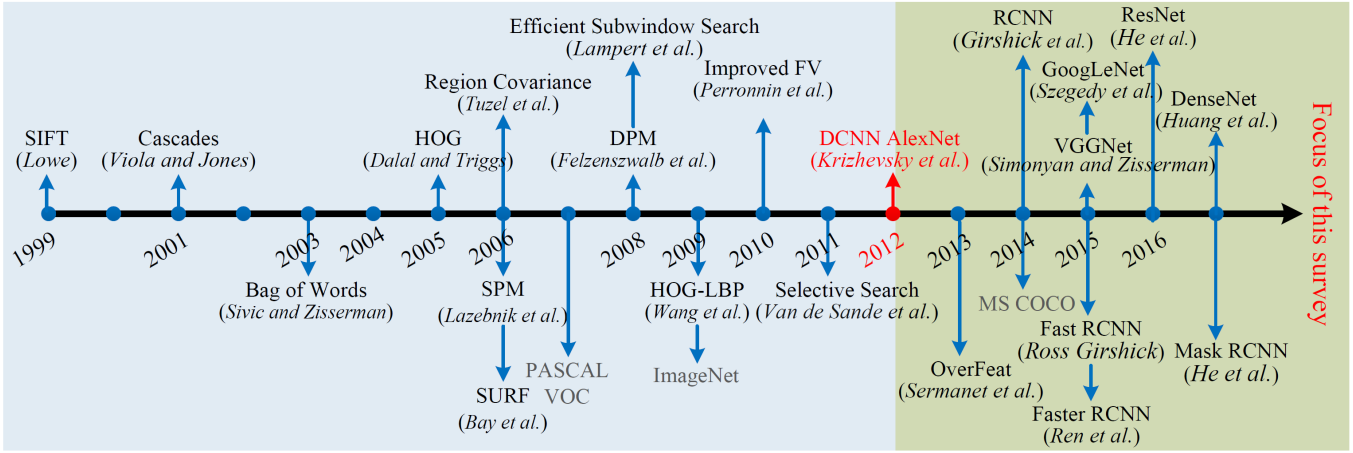


Figure: Evolution of generic object detection framework. Credit: *[8]*

- Before the 90's leading paradigm based on geometric representations.
- Since the 90's two main eras are highlighted, SIFT vs DCNN.
  - Appearance features moved from global to local representations, designed to be invariant to translation, scale, rotation, illumination, viewpoint and occlusion.
  - Handcrafted local invariant features gained tremendous popularity, starting from the Scale Invariant and Feature Transform (SIFT).

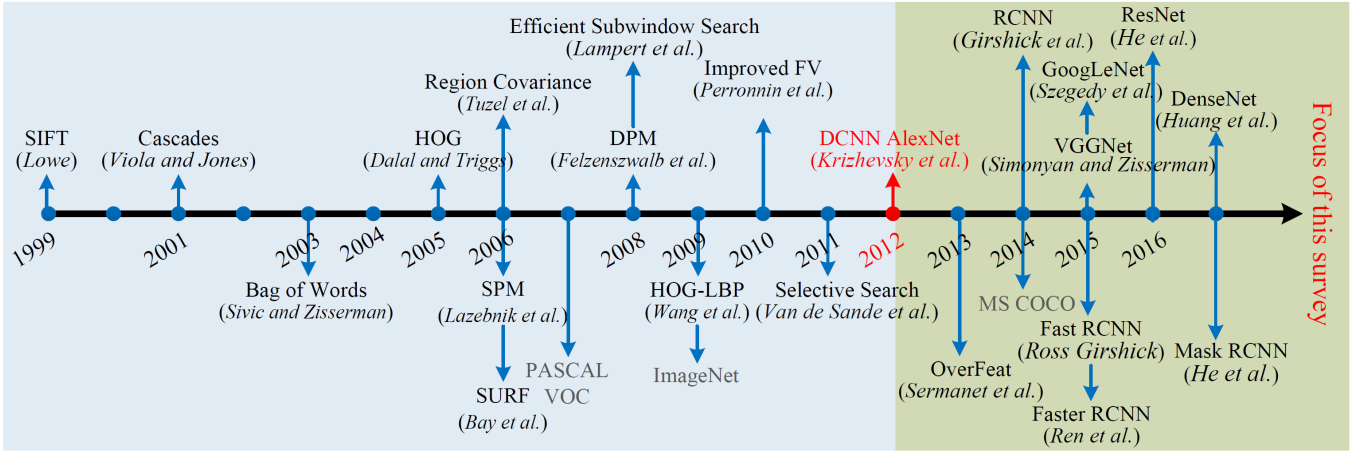


Figure: Evolution of generic object detection framework. Credit: *[8]*

- Significant turning point in 2012 when DCNNs achieved their record-breaking results in image classification.
- deeper CNNs have lead to record-breaking improvements in detection of more general object categories, with the milestone Region-based CNN.
- Availability of GPUs and large datasets such as ImageNet [13] or MS COCO [7] play a key role in their success.

Part III

Datasets and Performance Evaluation

4 Datasets

5 Evaluation Criteria

- Intersection Over Union
- Precision-Recall Curve
- Average Precision
- Mean Average Precision - mAP

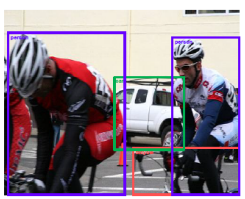
Four famous large high quality datasets set milestones: PASCAL VOC [2], ImageNet [5], MS COCO [7] and Open Images [6]. For creating large-scale annotated datasets:

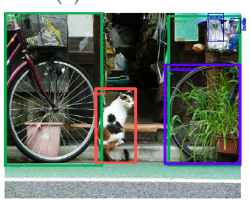
1. determining the set of target object categories,
2. collecting a diverse set of candidate images to represent the selected categories on the Internet,
3. annotating the large amount of collected images, typically by designing crowdsourcing strategy.

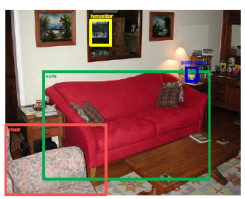
The four datasets form the backbone of their respective detection challenges. Each challenge consists of a publicly available dataset of images together with ground truth annotation and standardized evaluation software, and an annual competition and corresponding workshop.

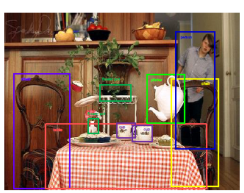
Dataset Name	Total Images	Categories	Images Per Category	Objects Per Image	Image Size	Started Year	Highlights
PASCAL VOC (2012) [67]	11,540	20	303 ∼ 4087	2.4	470 × 380	2005	Covers only 20 categories that are common in everyday life; Large number of training images; Close to real-world applications; Significantly larger intra-class variations; Objects in scene context; Multiple objects in one image; Contains many difficult samples; Creates the precedent for standardized evaluation of recognition algorithms in the form of annual competitions.
ImageNet [230]	14 millions+	21,841	–	1.5	500 × 400	2009	Considerably larger number of object categories; More instances and more categories of objects per image; More challenging than PASCAL VOC; Popular subset benchmarks ImageNet1000; The backbone of ILSVRC challenge; Images are object-centric.
MS COCO [162]	328,000+	91	–	7.3	640 × 480	2014	Even closer to real world scenarios; Each image contains more instances of objects and richer object annotation information; Contains object segmentation notation data that is not available in the ImageNet dataset; The next major dataset for large scale object detection and instance segmentation.
Places [311]	10 millions+	434	–	–	256 × 256	2014	The largest labeled dataset for scene recognition; Four subsets Places365 Standard, Places365 Challenge, Places 205 and Places88 as benchmarks.
Open Images [139]	9 millions+	6000+	–	–	varied	2017	A dataset of about 9 million images that have been annotated with image level labels, object bounding boxes and visual relationships; Support large scale object detection; Support visual relationship detection;

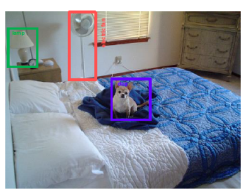
(a) PASCAL VOC



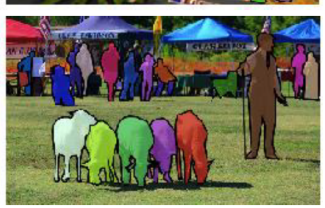




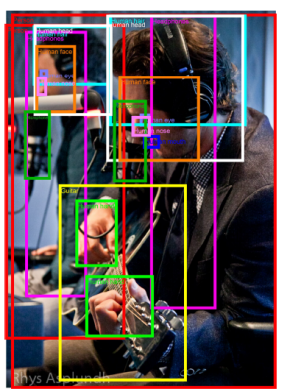


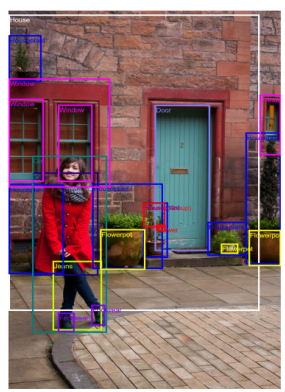


(b) ILSVRC



(c) MS COCO





(d) Open Images Detection

Figure: Main available datasets. Credit: *[8]*

Question

Your boss ask you to train an algorithm able to detect saiboats at sea. What is your first idea for creating a train set ?

4 Datasets

5 Evaluation Criteria

- Intersection Over Union
- Precision-Recall Curve
- Average Precision
- Mean Average Precision - mAP

Evaluating the performance of detection algorithms:

- Detection speed in Frame Per Second (FPS)
- $Precision = \frac{TP}{TP+FP}$ - What proportion of positive identifications was actually correct?
- $Recall = \frac{TP}{TP+FN}$ - What proportion of actual positives was identified correctly?

The standard outputs of a detector applied to a testing image I are the predicted detections $\{(b_j; c_j; p_j)\}_j$, indexed by j and denotes

- the predicted location (i.e., the Bounding Box, BB) b,
- its predicted category label c,
- its confidence level p.

Figure: Example of a detection. Credit: *Eca Robotics*

A predicted detection $(b; c; p)$ is TP if

- The predicted category label $c ==$ the ground truth label c_g i.e. if $p > \epsilon$, where ϵ is a chosen threshold.
- The overlap ratio Intersection Over Union $IOU(b, b^g) = \frac{area(b \cap b^g)}{area(b \cup b^g)} > \beta$, typically 0.5.

Otherwise, it is considered as FP.

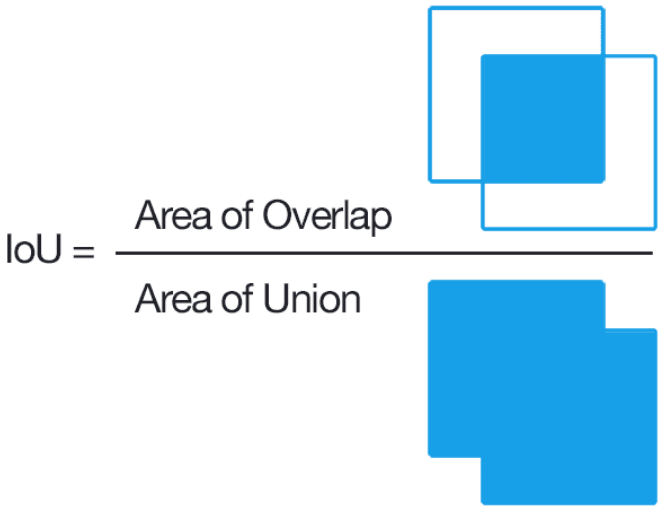


Figure: IoU formula

- Precision-recall curve the shows the tradeoff between the precision and recall values for different thresholds, here β.
- Helps to select the best threshold to maximize both metrics.
- For creating that curve you need:
  - the ground-truth labels,
  - the prediction scores of the samples,
  - some thresholds to convert the prediction scores into class labels.

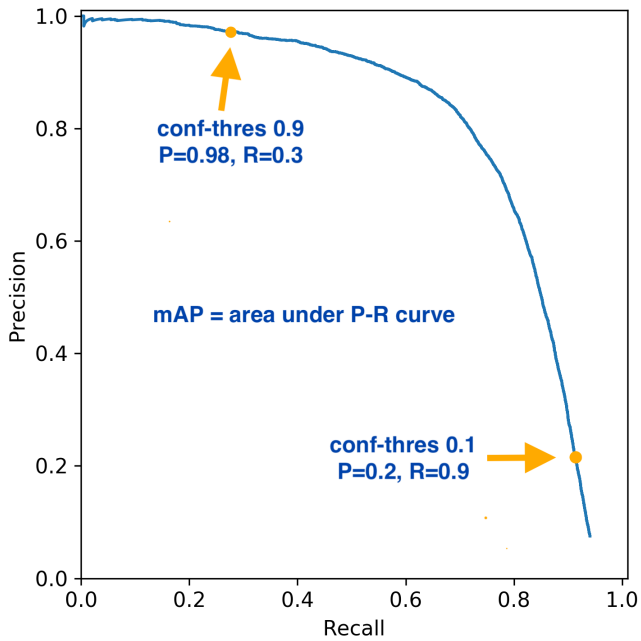


Figure: Precision-recall curve example

The most commonly used metric is *Average Precision* (AP), derived from precision and recall and computed by class.

1. For a given class c and a testing image I, let $(b_j; p_j)_{j=1}^{M}$ denote the detections returned by a detector, ranked by the confidence p_j in decreasing order.
2. Each detection $(b_j; p_j)$ is either a TP or a FP using IoU calculus.
3. Based on the TP and FP detections, the precision $P(\beta)$ and recall $R(\beta)$ can be computed as a function of IoU threshold β.
4. Finally AP is a sum of the increases on recall weighted by the precision.
5. For example for thresholds $\beta \in \{0.5, .., 0.9\}$,

$$AP = \sum_{\beta=0.5}^{0.9} (R(\beta) - R(\beta + 1)) * P(\beta)$$

, with $R(0) = 0$ and $P(1) = 1$

- AP is a way to summarize the precision-recall curve into a single value representing the average of all precisions.
- It is the weighted sum of precisions at each threshold where the weight is the increase in recall.
- Question: Could you imagine a single metric for evaluating object detector on a whole dataset ?

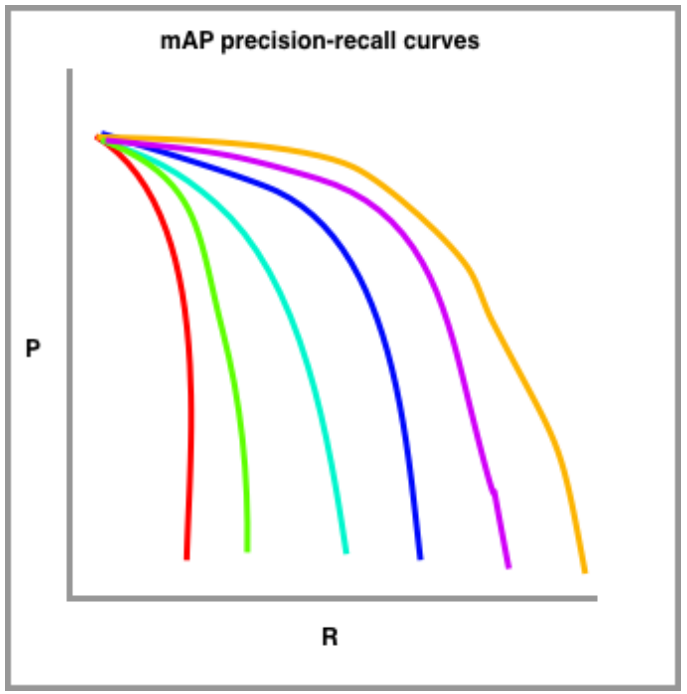


Figure: Average Precision

- Object detection models are evaluated with different IoU thresholds where each threshold may give different predictions from the other thresholds.
- For example for COCO: $IoU = .50 : .05 : .95$
- $$mAP = \frac{1}{n} \sum_{k=1}^{n} AP_k$$

  , where AP_k is the AP of class k and n is the number of classes.

	AP	AP^{50}	AP^{75}	AP^{S}	AP^{M}	AP^{L}
Noah CV Lab (Huawei)	0.588	0.766	0.649	0.407	0.616	0.720
mmdet	0.578	0.770	0.637	0.399	0.605	0.706
DeepAR(ETRIxKAIST_AIM)	0.553	0.746	0.609	0.378	0.583	0.668
DetectoRS	0.550	0.736	0.604	0.377	0.578	0.669
KiwiDet2	0.547	0.728	0.597	0.362	0.576	0.685
360 AI Research	0.546	0.737	0.600	0.369	0.583	0.674
ZFTurbo	0.544	0.728	0.600	0.367	0.579	0.671
Hyundai Mobis AD Lab	0.538	0.721	0.591	0.343	0.571	0.675
ByteDance_VC	0.534	0.723	0.589	0.351	0.572	0.663
ResNeSt	0.533	0.720	0.580	0.351	0.562	0.668

Figure: Average Precision on COCO detection leaderboard

Part IV

Detection frameworks

Detectors organized into two main categories:

1. Two stage detection framework, which includes
   1. region proposal,
   2. objects classification and localization.
2. One stage detection framework, or region proposal free framework, which is method to propose region, classify and localize objects in a single step.

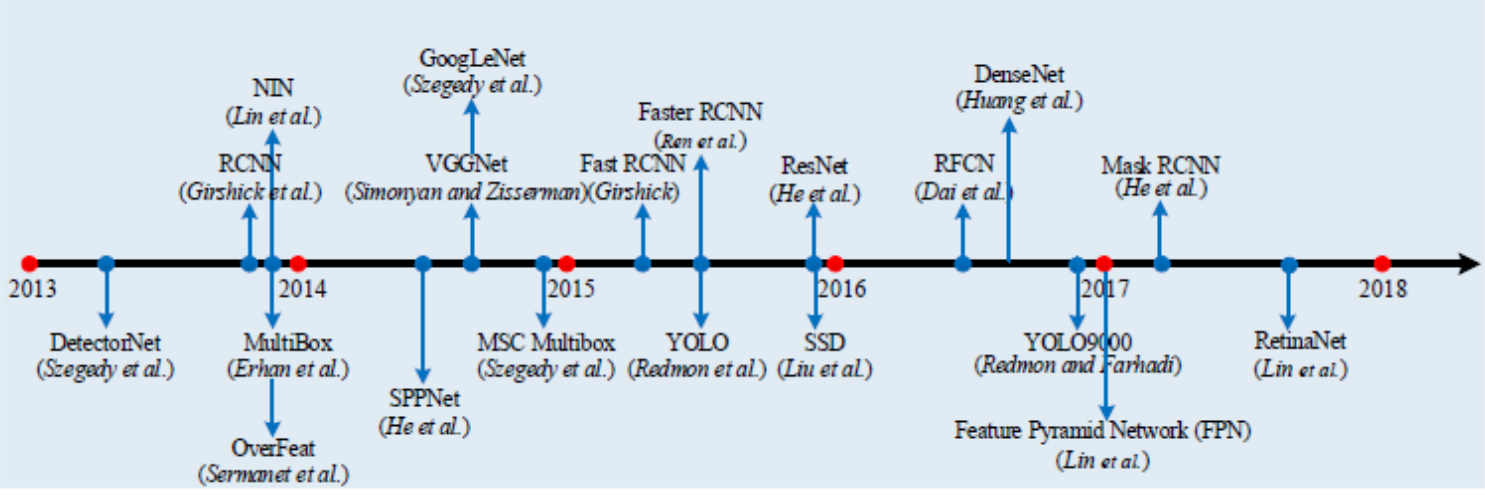


Figure: Milestones of object detection frameworks Credit: *[8]*

6 Region based (Two Stage) Frameworks

- Philosophy
- RCNN: Girshick *et al. [4]*
- Fast RCNN: Girshick [3]
- A word on RoI Pooling
- Faster RCNN: Ren *et al. [12]*
- A word on RPN

7 Unified (One Stage) Frameworks

- Philosophy
- Overfeat: Sermanet *et al. [14]*
- YOLO - You Only Look Once: Redmon *et al. [11]*
- YOLOV2 and YOLO9000: Redmon and Farhadi [10]
- SSD - Single Shot Multibox Detector: Liu *et al. [9]*

- Category-independent region proposals are generated from an image,
- features extracted from these regions,
- category-specific classifiers used to determined the category of the proposals.

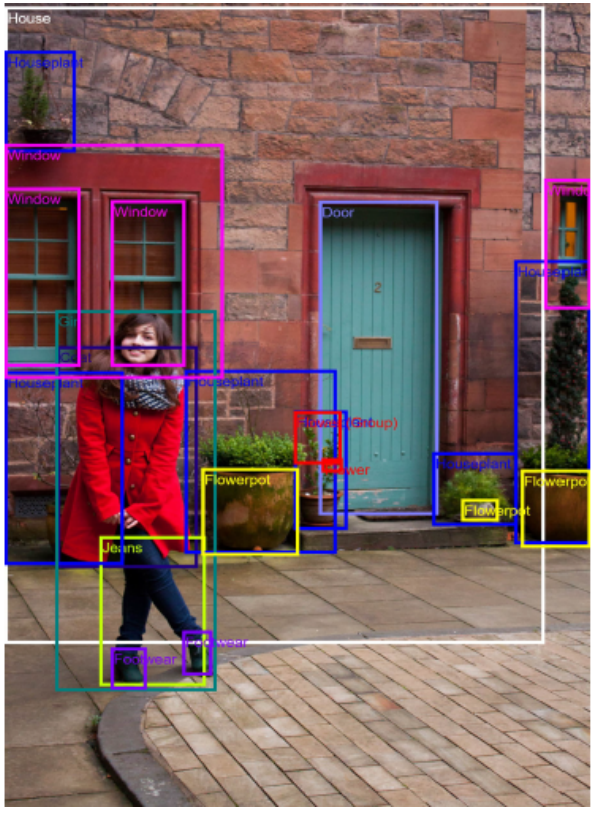


Figure: Regions and classes.
Credit: *[8]*

In 2013, Girshick proposed CNN for object detection.

1. *Region proposal computation*: Class agnostic region proposals via selective search.
2. *CNN model finetuning*: Finetuning CNN pretrained model with region proposals, cropped from the image and warped into the same size.

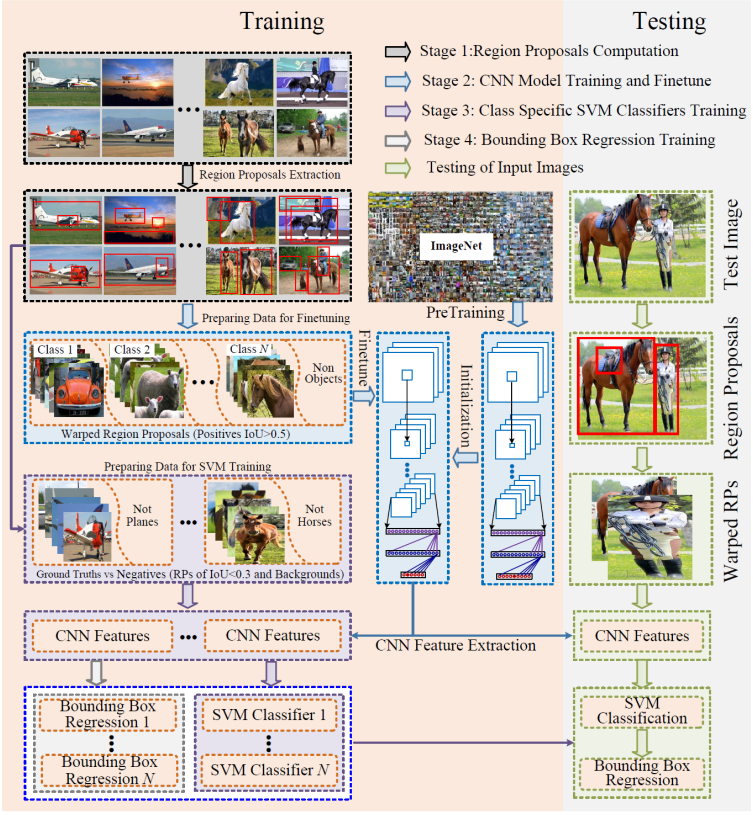


Figure: RCNN. Credit: *[8]*

3 *Class specific SVM classifiers training*. A set of class specific linear SVM classifiers are trained using fixed length features extracted with CNN, replacing the softmax classifier learned by finetuning.

4 *Class specific bounding box regressors training*. Bounding box regression is learned for each object class with CNN features.

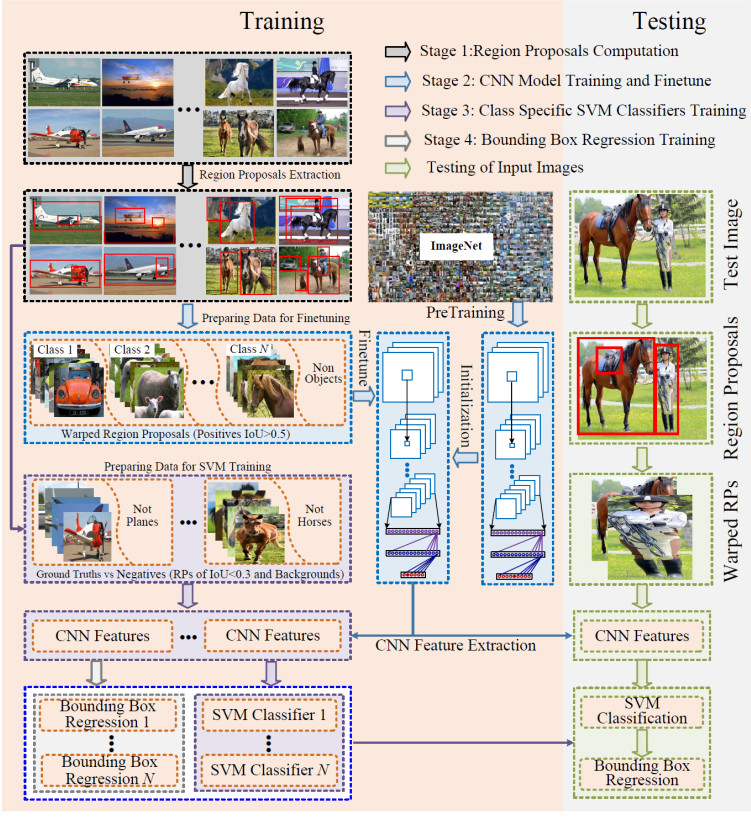


Figure: RCNN. Credit: *[8]*

Drawbacks

1. Each individual stage must be trained separately which is slow and hard to optimize.
2. For SVM classifier and bounding box regressor training, it is expensive in both disk space and time, because CNN features need to be extracted from each object proposal in each image and saved to disk, posing great challenges for large scale detection.
3. Testing is slow, since CNN features are extracted per object proposal in each testing image, without sharing computation.

- Simultaneously learns a softmax classifier and a class-specific bounding box regression via 2 siblings output layers.
- Shares the computation of convolution across region proposals.
- Adds a Region of Interest (RoI) pooling layer between the last CONV layer and the first FC layer to extract a fixed-length feature for each region proposal.

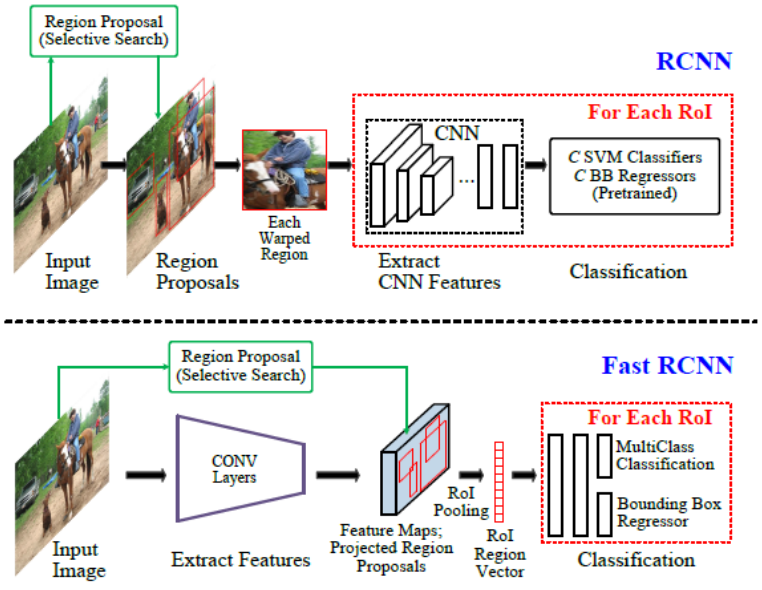


Figure: From RCNN to Fast RCNN.
Credit: *[8]*

- 3 times faster in training and 10 times faster in testing than RCNN.
- In summary:
  - higher detection quality,
  - single-stage training process that updates all network layers,
  - no storage required for feature caching.

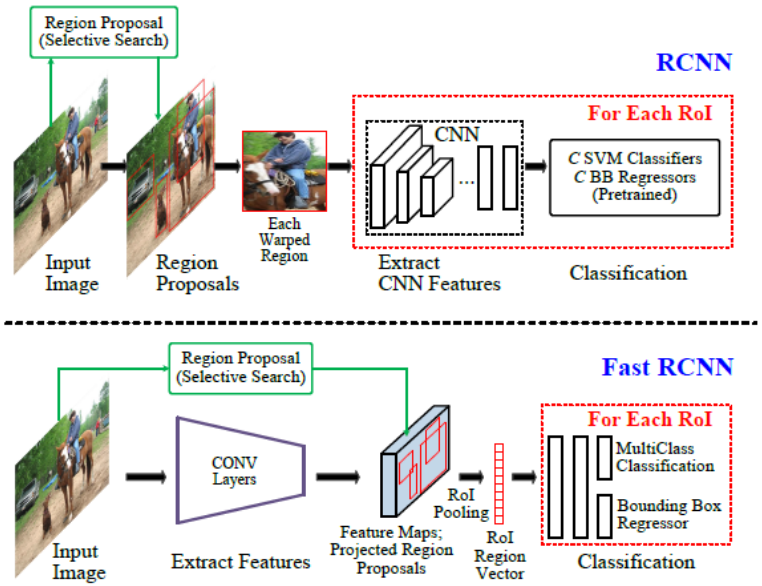
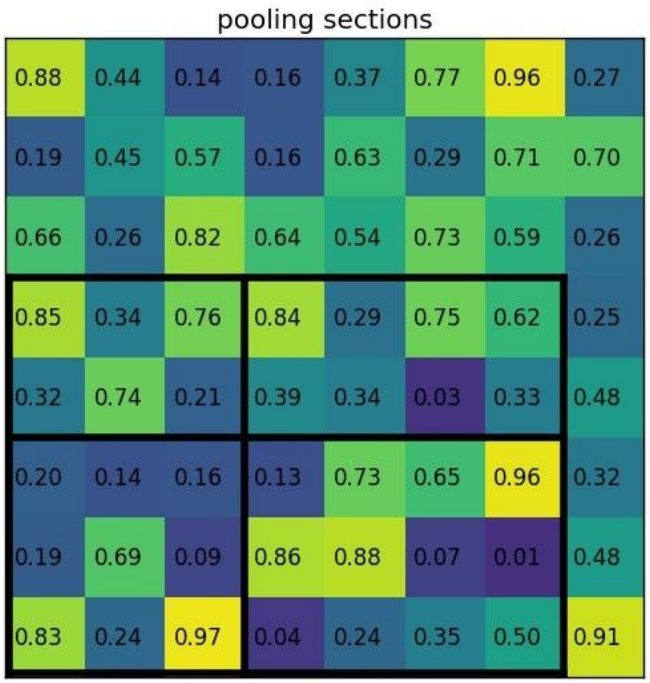


Figure: From RCNN to Fast RCNN.
Credit: *[8]*

Region of interest pooling is used to generate fixed-sized feature maps from CONV features maps to be used in FC layers. For example we want to RoI pool from a single 8×8 feature map, one region of interest (big black square) to an output feature map of 2×2.



- Selective Search (speed bottleneck) replaced by a CNN in producing region proposals, efficient and accurate Region Proposal Network (RPN).
- Same backbone network to accomplish the task of RPN for region proposal and Fast RCNN for region classification.

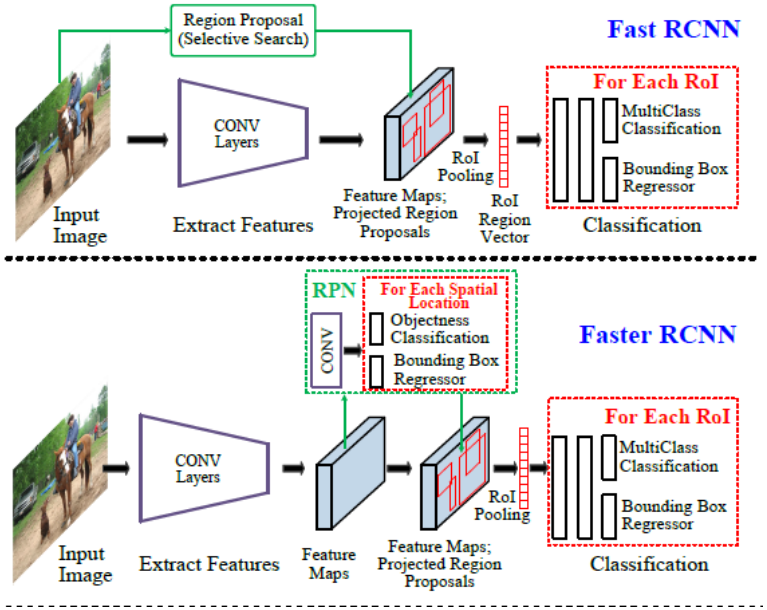


Figure: From Fast RCNN to Faster RCNN. Credit: *[8]*

We use RPN for:

- *image classification*: generate the candidate boxes (which might have our objects to detect) and classify those boxes as one of the objects.
- *bounding box regression*: box shape adjustments learn to properly fit the actual object.

This is done in three steps:

1. Generate anchor boxes.
2. Classify each anchor box whether it is foreground or background.
3. Learn the shape offsets for anchor boxes to fit them for objects.

Anchor boxes

- Every point in the feature map generated by the backbone network is an anchor point.
- From these points we generate candidate boxes using two parameters — scales and aspect ratios.
- The boxes need to be at image dimensions, whereas the feature map is reduced depending on the backbone.
- Ex: If anchor scales are [8,16,32] and ratios are [0.5,1,2] and stride is 16, then we use the combination of these scales and ratios to generate 9 anchor boxes for each anchor point and then take a stride of 16 over the image to take the next anchor box.

Anchor boxes

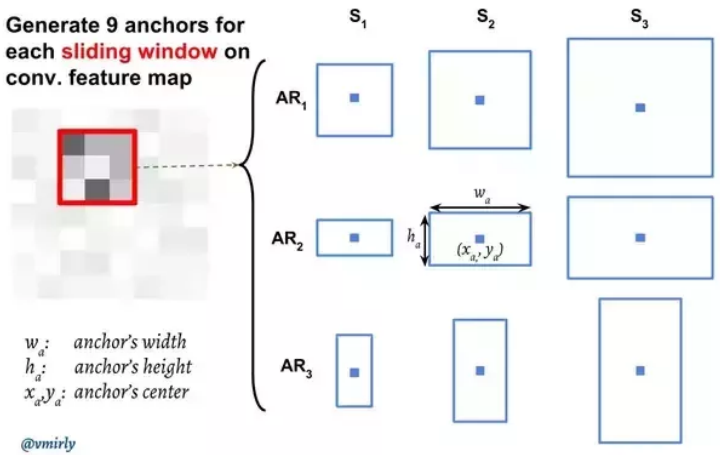


Figure: Anchor boxes generation.

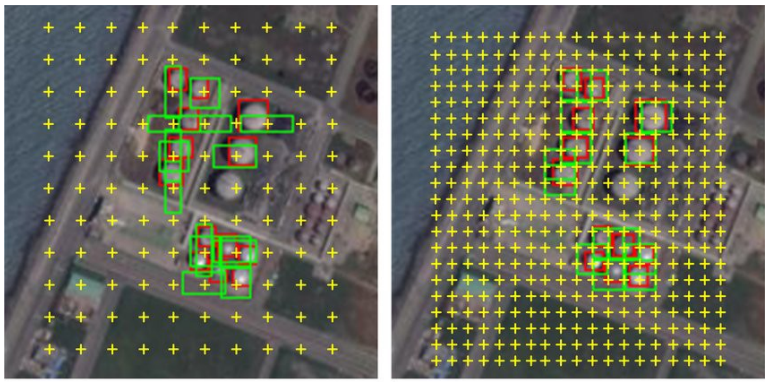


Figure: Anchor points with different scales.

Foreground/Background and Bbox regression

- Learn whether the given box is foregroud (object) or background and the offsets for the foreground boxes to adjust for fitting the objects.
- These tasks are achieved by two convolution layers on the feature map obtained from the backbone network.

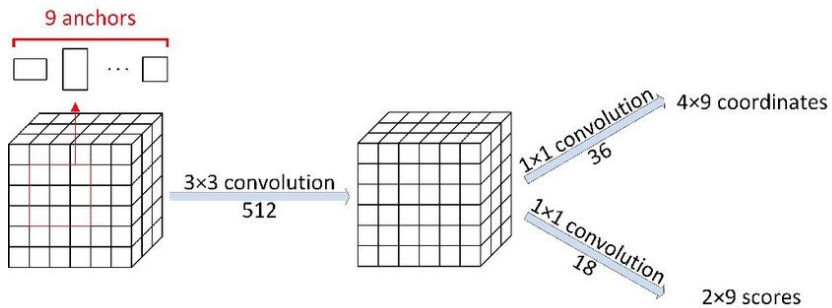


Figure: 2 scores (fg and bg) and 4 offsets for coordinates.

Foreground/Background and Bbox regression

- In anchor target generation, we calculate the IOU of GT boxes with anchor boxes to check if it is fg/bg and then the difference in the coordinates are calculated as targets to be learned by the regressor. Then these targets are used as input for cross-entropy loss and smooth l1 loss.
- Once these fg/bg scores and offsets are learned using convolution layers, some portions of fg and bg boxes are considered according to confidence scores. The offsets are applied to those boxes to get the actual ROIs to be processed further. This post-processing of anchor boxes using offsets is called proposal generation.

6 Region based (Two Stage) Frameworks
- Philosophy
- RCNN: Girshick *et al. [4]*
- Fast RCNN: Girshick [3]
- A word on RoI Pooling
- Faster RCNN: Ren *et al. [12]*
- A word on RPN

7 Unified (One Stage) Frameworks
- Philosophy
- Overfeat: Sermanet *et al. [14]*
- YOLO - You Only Look Once: Redmon *et al. [11]*
- YOLOV2 and YOLO9000: Redmon and Farhadi [10]
- SSD - Single Shot Multibox Detector: Liu *et al. [9]*

- Region-based approaches are computationally expensive for current mobile/wearable devices.
- Instead of trying to optimize individual components of a complex region-based frameworks, researchers have begun to develop *unified* detection strategies.
- Directly predict classes probabilities and bouding box offsets from full images with a single feed-forward CNN in a monolithic setting that does net involve region proposal generation of post classification / feature resampling, encapsulating all computation in a single network.

- First single-stage object detectors based on fully convolutional deep networks.
- Performs object detection via a single forward pass though the fully convolutional layers in the network.
  1. Generate object candidates by performing object classification via a sliding window fashion on multiscale images.
  2. Increase the number of predictions by offset max pooling.
  3. Bounding box regression.
  4. Combine predictions using greedy merge strategy.
- Overfeat has a significant speed advantage, but is less accurate than RCNN.
- Difficult to train fully convolutional networks at the time.

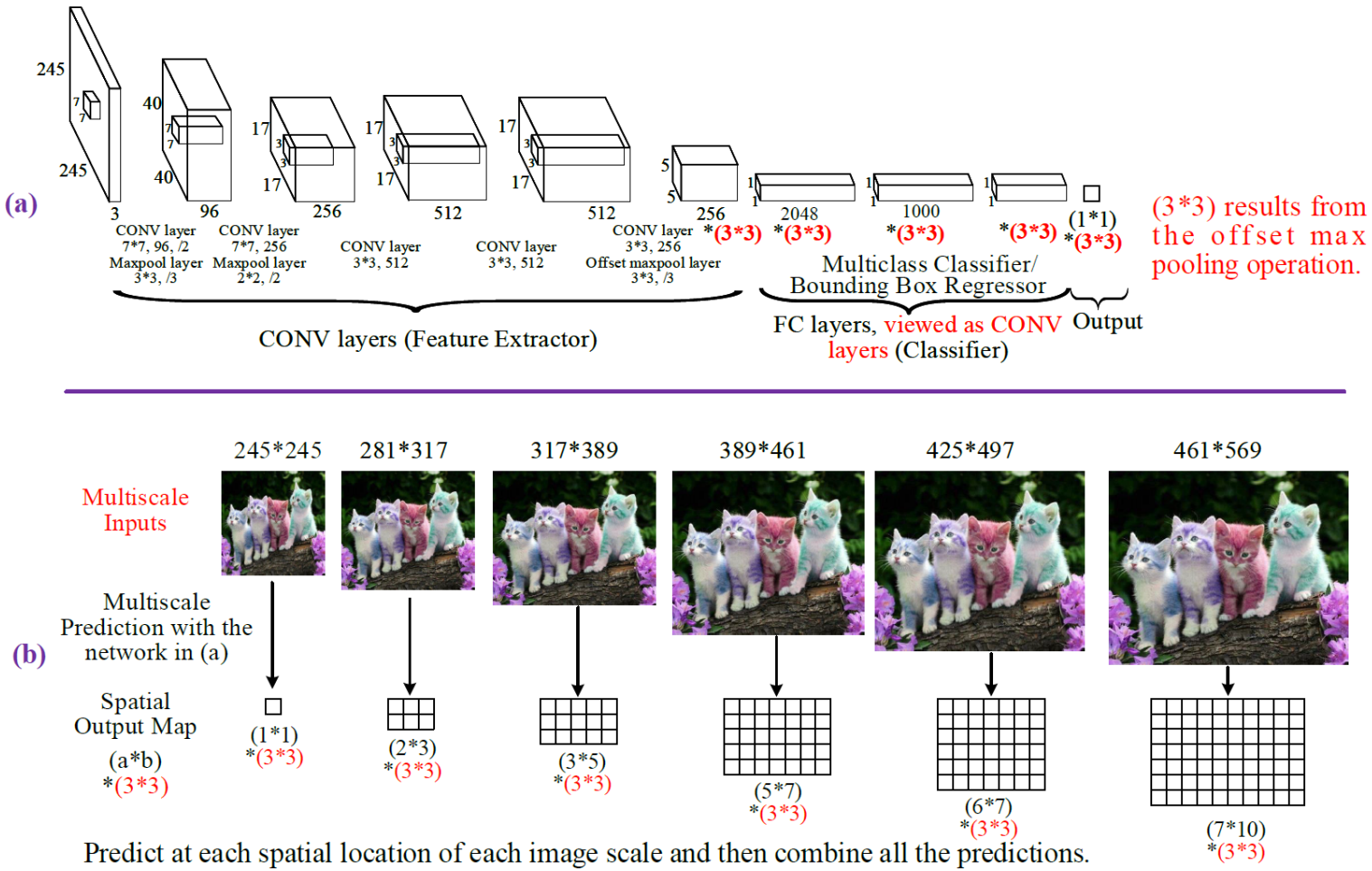


Figure: Overfeat. Credit: *[8]*

- Directly predicts detections using a small set of candidate regions.
- Divides an image into an SxS grid, each predicting C class probabilities, B bounding box locations, and confidence scores.
- Uses GoogLeNet as backbone.
- Sees the entire image, encodes contextual information about object classes, is less likely to predict false positives in the backgroud.

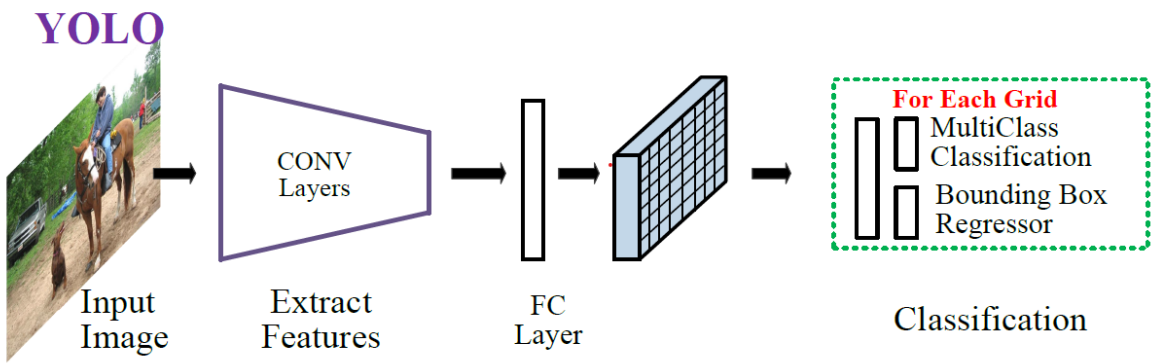


Figure: YOLO. Credit: *[8]*

YOLOV2 Improves YOLO:

- Custom GoogLeNet is replaced with the simpler DarkNet19 plus batch normalization.
- Anchor boxes are learned via kmeans and multiscale training.

YOLO9000 can detect over 9000 object categories in real time

- Joint optimization method to train simultaneously on an ImageNet classification dataset and a COCO detection dataset.
- Allow to perform weakly supervised detection (detecting object classes that do not have bounding boxes annotations)

Combines ideas from RPN (competitive accuracy), YOLO (but faster) and multiscale CONV features.

- CNN fully convolutional, early stages based on a standard architecture such as VGG.
- Several auxiliary CONV layers, progressively decreasing in size.
- Information on the last layer may be too coarse spatially to allow precise localization so SSD performs detection over multiple scales.
- Uses Non-Maximum Suppression (NMS) step to produce final detection.

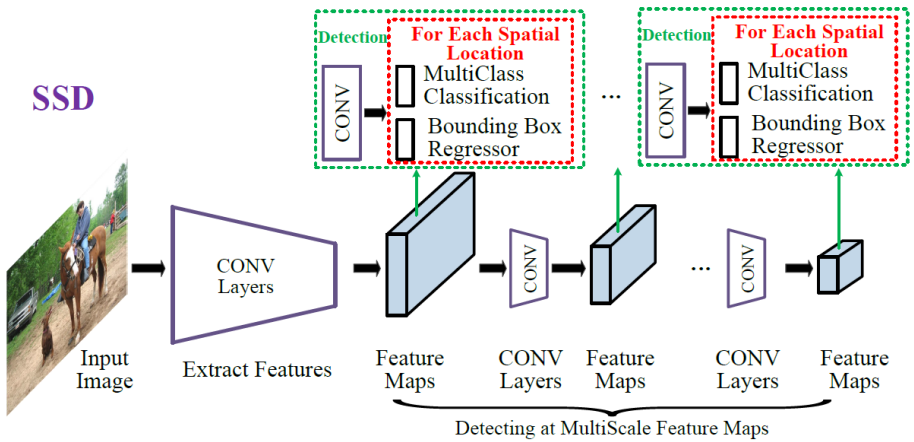


Figure: SSD. Credit: *[8]*

A word on Non-Maximum Suppression

Algorithm 1 Non-Max Suppression

procedure NMS(B,c)
  $B_{nms} \leftarrow \emptyset$ Initialize empty set
  for $b_i \in B$ **do** => Iterate over all the boxes
    $discard \leftarrow$ False Take boolean variable and set it as false. This variable indicates whether b(i) should be kept or discarded
    for $b_j \in B$ **do** Start another loop to compare with b(i)
      if same$(b_i, b_j) > \boldsymbol{\lambda_{nms}}$ **then** If both boxes having same IOU
        if score$(c, b_j) >$ score(c, b_i) **then**
          $discard \leftarrow$ True Compare the scores. If score of b(i) is less than that of b(j), b(i) should be discarded, so set the flag to True.
    if not $discard$ **then** Once b(i) is compared with all other boxes and still the discarded flag is False, then b(i) should be considered. So add it to the final list.
      $B_{nms} \leftarrow B_{nms} \cup b_i$
  return B_{nms} Do the same procedure for remaining boxes and return the final list

Figure: Algorithm

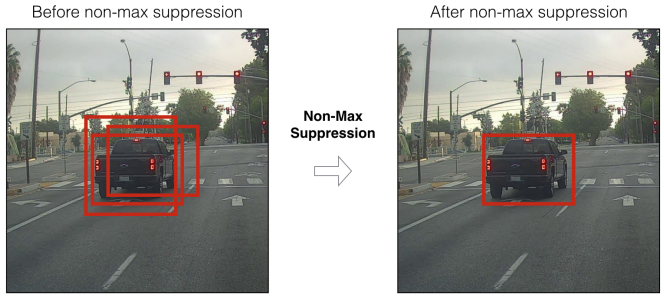


Figure: result

Part V

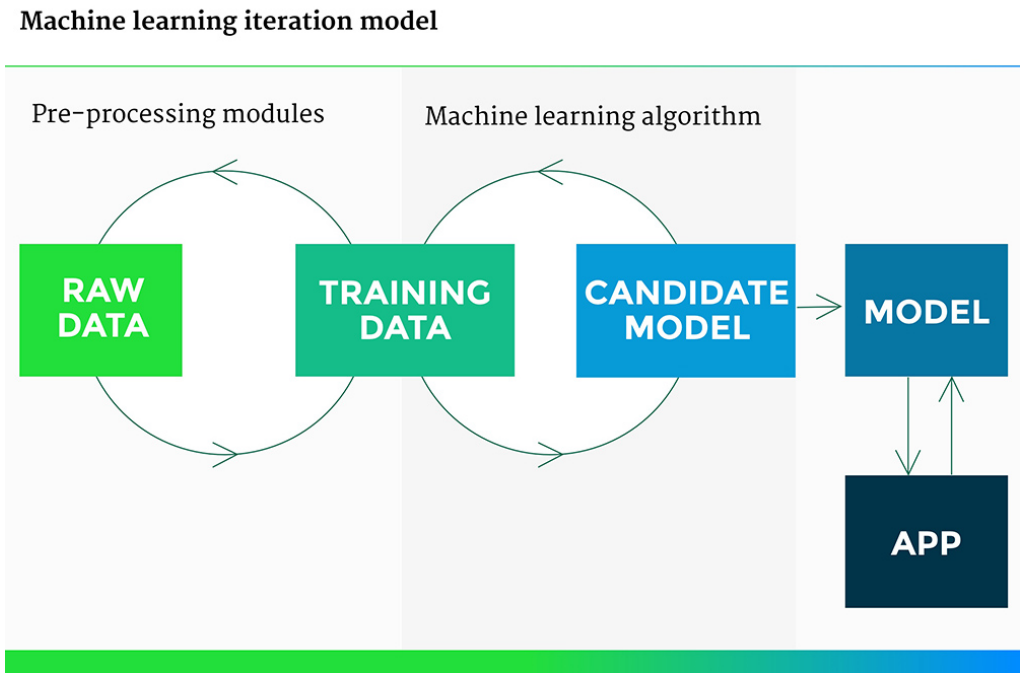
Industry advices

Questions to answer before working

- What is the target ? What should you detect ?
- What is the hardware to run the detector on ?
- Is there any annotated data ? Is there a way to get some ? Is there any open dataset on the subject ?
- Do I work from scratch or reuse the work of a colleague/GitHub ?
- Is there a minimal performance/accuracy to achieve ? A test/scenario the detector must pass ?

Advices

- Annotate data is very long. It should be avoided as soon as is it possible.
- But bad labels will end in bad models. Be sure of the quality or your data.
- Iterate/version your model.



Advices

- Communicate your work. Use jupyter notebooks, dashboards, graphs, ...
- Evaluation metrics are very useful to monitor your model and the training process but not for communicating with non specialists.
- A demonstration is better than tables of numbers.
- Use docker to ease the deployment of your environment and allow IT teams to put your model into production.

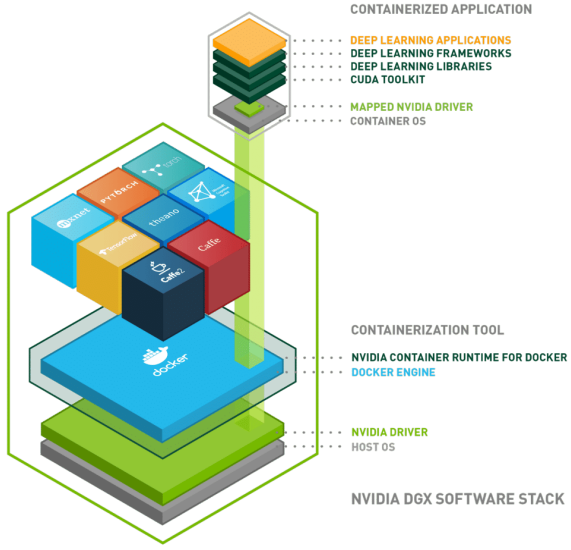


Figure: Docker Credit: *Nvidia*

Part VI

TP

- Link to google colab

[1] Vincent Dumoulin and Francesco Visin. "A guide to convolution arithmetic for deep learning". In: *arXiv:1603.07285 [cs, stat]* (Jan. 2018). arXiv: 1603.07285. URL: http://arxiv.org/abs/1603.07285 (visited on 10/26/2021).

[2] Mark Everingham et al. "The Pascal Visual Object Classes (VOC) Challenge". en. In: *Int J Comput Vis* 88.2 (June 2010), pp. 303–338. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-009-0275-4. URL: http://link.springer.com/10.1007/s11263-009-0275-4 (visited on 11/22/2021).

[3] Ross Girshick. "Fast R-CNN". In: *arXiv:1504.08083 [cs]* (Sept. 2015). arXiv: 1504.08083. URL: http://arxiv.org/abs/1504.08083 (visited on 11/23/2021).

[4] Ross Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *arXiv:1311.2524 [cs]* (Oct. 2014). arXiv: 1311.2524. URL: http://arxiv.org/abs/1311.2524 (visited on 11/23/2021).

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet classification with deep convolutional neural networks". en. In: *Commun. ACM* 60.6 (May 2017), pp. 84–90. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/3065386. URL: https://dl.acm.org/doi/10.1145/3065386 (visited on 11/22/2021).

[6] Alina Kuznetsova et al. "The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale". In: *Int J Comput Vis* 128.7 (July 2020). arXiv: 1811.00982, pp. 1956–1981. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-020-01316-z. URL: http://arxiv.org/abs/1811.00982 (visited on 11/22/2021).

[7] Tsung-Yi Lin et al. "Microsoft COCO: Common Objects in Context". In: *arXiv:1405.0312 [cs]* (Feb. 2015). arXiv: 1405.0312. URL: http://arxiv.org/abs/1405.0312 (visited on 11/22/2021).

[8] Li Liu et al. "Deep Learning for Generic Object Detection: A Survey". In: *arXiv:1809.02165 [cs]* (Aug. 2019). arXiv: 1809.02165 version: 3. URL: http://arxiv.org/abs/1809.02165 (visited on 10/26/2021).

[9] Wei Liu et al. "SSD: Single Shot MultiBox Detector". In: *arXiv:1512.02325 [cs]* 9905 (2016). arXiv: 1512.02325, pp. 21–37. DOI: 10.1007/978-3-319-46448-0_2. URL: http://arxiv.org/abs/1512.02325 (visited on 11/23/2021).

[10] Joseph Redmon and Ali Farhadi. "YOLO9000: Better, Faster, Stronger". In: *arXiv:1612.08242 [cs]* (Dec. 2016). arXiv: 1612.08242. URL: http://arxiv.org/abs/1612.08242 (visited on 11/23/2021).

[11] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *arXiv:1506.02640 [cs]* (May 2016). arXiv: 1506.02640. URL: http://arxiv.org/abs/1506.02640 (visited on 11/23/2021).

[12] Shaoqing Ren et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *arXiv:1506.01497 [cs]* (June 2015). arXiv: 1506.01497 version: 1. URL: http://arxiv.org/abs/1506.01497 (visited on 11/23/2021).

[13] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *arXiv:1409.0575 [cs]* (Jan. 2015). arXiv: 1409.0575. URL: http://arxiv.org/abs/1409.0575 (visited on 11/22/2021).

[14] Pierre Sermanet et al. “OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks”. In: *arXiv:1312.6229 [cs]* (Dec. 2013). arXiv: 1312.6229 version: 1. URL: http://arxiv.org/abs/1312.6229 (visited on 11/23/2021).