A Brief History of Object Detection

From Haar-like features to losing anchors @ PFN Day #4

July 11, 2019

Tommi Kerola

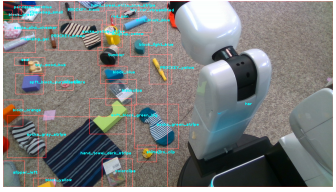
Preferred Networks

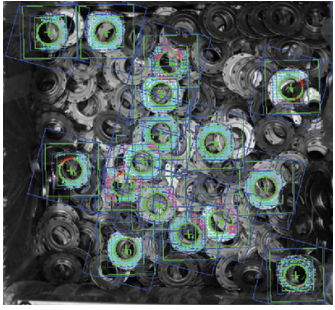
A Brief History of Object Detection

1. Motivation
2. History Before Deep Learning
3. Two-stage Methods
4. Single-shot Methods
5. Anchor-free Methods
6. Problems and Summary

Motivation

- Localizing objects is a crucial task for using computer vision in the real world.
  - Autonomous driving, personal robots, industrial robotics, ...





Aided by a 3D area sensor to locate random parts in a bin, the robot plans its next pick. *(Courtesy of Motion Controls Robotics, Inc.)*

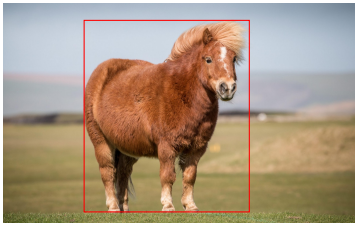
(Figs. from towardsdatascience.com, preferred.jp, robotics.org)

Preferred Networks

Problem definition

- Given an input image, predict the locations of a certain class of objects in the image.
- Locations are usually represented using bounding boxes.
- Evaluation using IoU (intersection over union).

$$\texttt{IoU} \triangleq \frac{A \cap B}{A \cup B}, \text{ typically, } \texttt{IoU} \geq 0.5 \implies \text{match with GT} \tag{1}$$

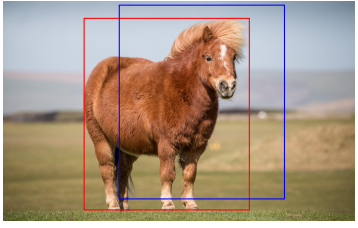


Ground-truth class: Horse
Ground-truth location:
Predicted class: Horse
Predicted location:

Problem definition

- Given an input image, predict the locations of a certain class of objects in the image.
- Locations are usually represented using bounding boxes.
- Evaluation using IoU (intersection over union).

$$\texttt{IoU} \triangleq \frac{A \cap B}{A \cup B}, \text{ typically, } \texttt{IoU} \geq 0.5 \implies \text{match with GT} \tag{1}$$



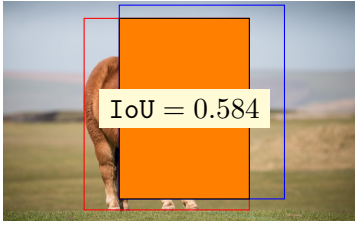
Ground-truth class: Horse
Ground-truth location:
Predicted class: Horse
Predicted location:

(Fig. from telegraph.co.uk)

Problem definition

- Given an input image, predict the locations of a certain class of objects in the image.
- Locations are usually represented using bounding boxes.
- Evaluation using IoU (intersection over union).

$$\texttt{IoU} \triangleq \frac{A \cap B}{A \cup B}, \text{ typically, } \texttt{IoU} \geq 0.5 \implies \text{match with GT} \tag{1}$$



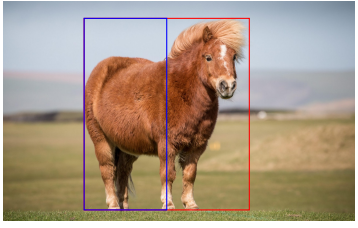
Ground-truth class: Horse
Ground-truth location:
Predicted class: Horse
Predicted location:

(Fig. from telegraph.co.uk)

Problem definition

- Given an input image, predict the locations of a certain class of objects in the image.
- Locations are usually represented using bounding boxes.
- Evaluation using IoU (intersection over union).

$$\texttt{IoU} \triangleq \frac{A \cap B}{A \cup B}, \text{ typically, } \texttt{IoU} \geq 0.5 \implies \text{match with GT} \tag{1}$$



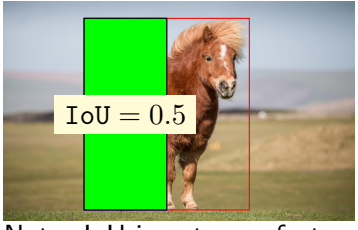
Ground-truth class: Horse
Ground-truth location:
Predicted class: Horse
Predicted location:

(Fig. from telegraph.co.uk)

Problem definition

- Given an input image, predict the locations of a certain class of objects in the image.
- Locations are usually represented using bounding boxes.
- Evaluation using IoU (intersection over union).

$$\texttt{IoU} \triangleq \frac{A \cap B}{A \cup B}, \text{ typically, } \texttt{IoU} \geq 0.5 \implies \text{match with GT} \tag{1}$$



Ground-truth class: Horse
Ground-truth location:
Predicted class: Horse
Predicted location:

- Note: IoU is not a perfect metric, but the current de-facto standard.

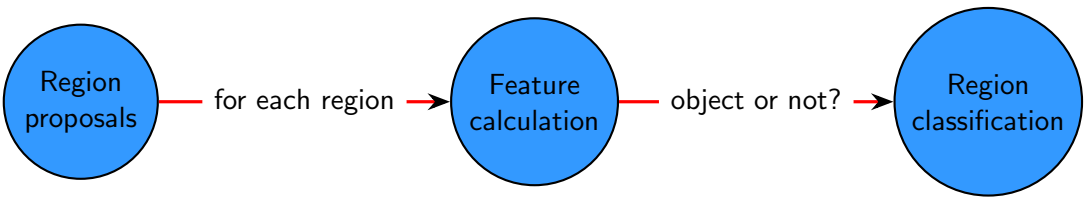
(Fig. from telegraph.co.uk)

Target Audience

People who

- Know the basics of deep learning and CNNs.
- Have not implemented a DL-based object detector before.
  - But want to find out how!

Purpose of this Talk



- Understand general object detection pipeline (above).
- Explain the history of object detection up until recent research.
- Illustrate various types of recent object detectors and their merits.

1 Motivation

2 History Before Deep Learning

3 Two-stage Methods

4 Single-shot Methods

5 Anchor-free Methods

6 Problems and Summary

History Before Deep Learning

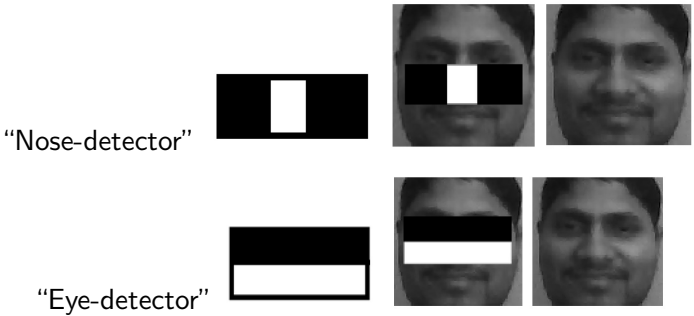
- Early object detectors were based on handcrafted features.
- Sliding window classifier, check if feature response is strong enough, if so: output detection.

- We will review a few representative features pre-deep learning.
  - Haar-like features
  - Histograms of Oriented Gradients
  - Deformable Part Models

(Fig. from researchgate.net)

Haar-like features [Viola and Jones(2001)]

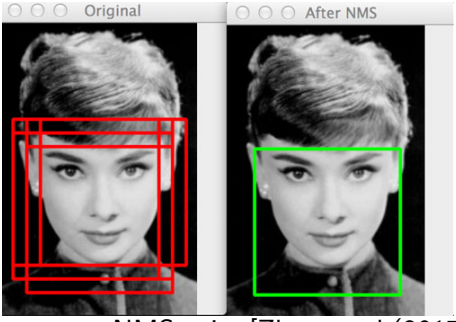
- Hand-crafted weak features, calculate in sliding window, use boosted classifier like AdaBoost. Name comes from similarity to Haar wavelets.
- e.g. “Is the middle part of the image darker than the outer parts?”



(Fig. from wikipedia.org)

Necessary Post-Processing: Non-maximum Suppression

- Sliding window classifiers yield lots of correlated detections.
- Non-maximum suppression (NMS) is a simple, greedy algorithm for turning these into a single detection per object (ideally).

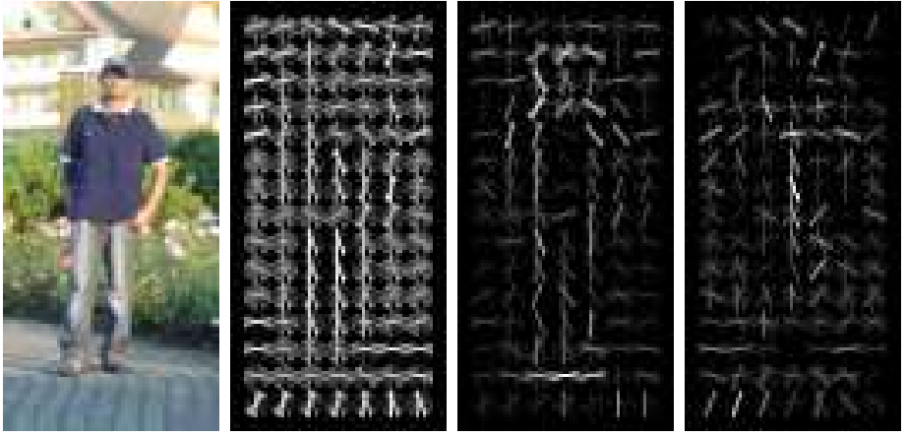


- (Recently, improvements upon NMS exist [Zhou et al.(2017), Bodla et al.(2017)])

(Fig. from pyimagesearch.com)

Histograms of Oriented Gradients [Dalal and Triggs(2005)]

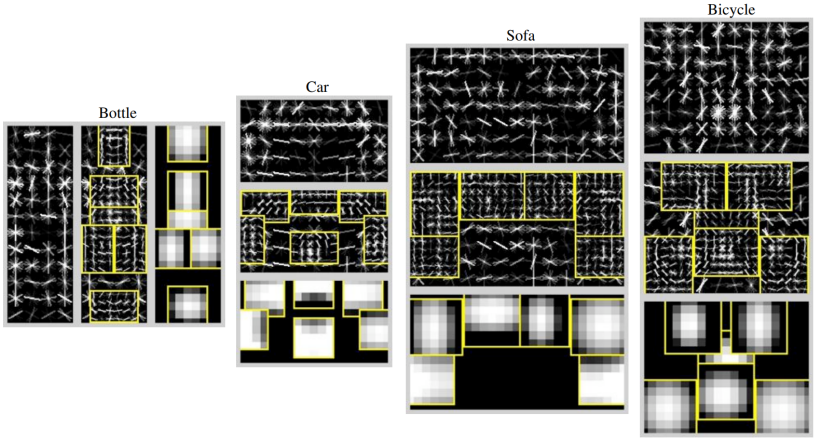
- Computes histogram of gradient orientation (HOG feature) over sub-image blocks.



(Fig. from [Dalal and Triggs(2005)])

Deformable Part Models [Felzenszwalb et al.(2008)]

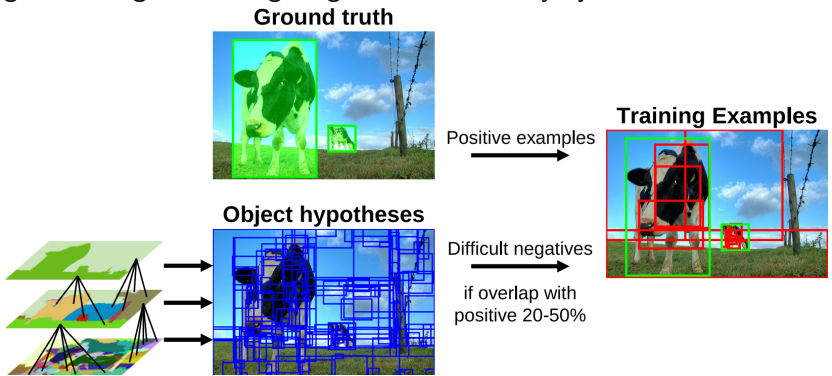
- Learn the relationships between HOG features of object parts via a latent SVM.



(Fig. from [Felzenszwalb et al.(2008)])

Better Region Proposals: Selective Search [Uijlings et al.(2013)]

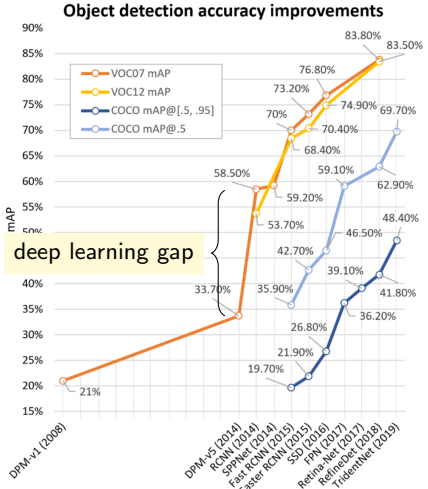
- Instead of sliding window, propose regions that have high "objectness".
- Oversegment image and merge regions hierarchically by color, texture, size and shape.



(Fig. from [Uijlings et al.(2013)])

Deep Learning Era [Krizhevsky et al.(2012)]

- Starting with AlexNet in 2012, deep learning methods significantly improved image classification.
- The same holds for object detection: > 30 pp. gap. Recent methods give $\sim 150\%$ improvement.



(Fig. from [Zou et al.(2019)])

1 Motivation

2 History Before Deep Learning

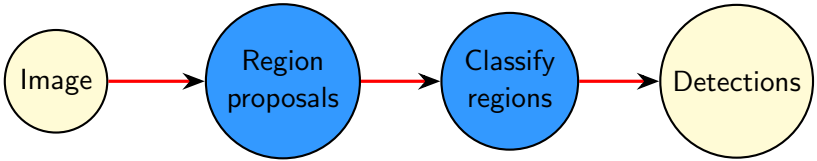
3 Two-stage Methods

4 Single-shot Methods

5 Anchor-free Methods

6 Problems and Summary

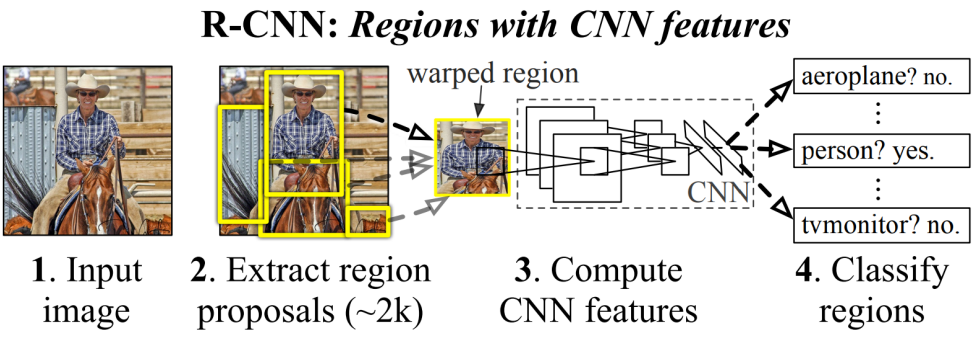
Two-stage Methods



- As implied, operates in two serial stages:
  - Generate region proposals (instead of sliding window).
  - Classify each proposed region, if feature response strong enough, output detection.
- **When to use?** Two-stage methods are accurate but computationally heavy.
- We will look at three historically representative methods:
  - R-CNN
  - Fast R-CNN
  - Faster R-CNN
- (Further reading: [Li et al.(2019), Lin et al.(2017a), Lu et al.(2019), Singh et al.(2018)])

R-CNN [Girshick et al.(2014)]

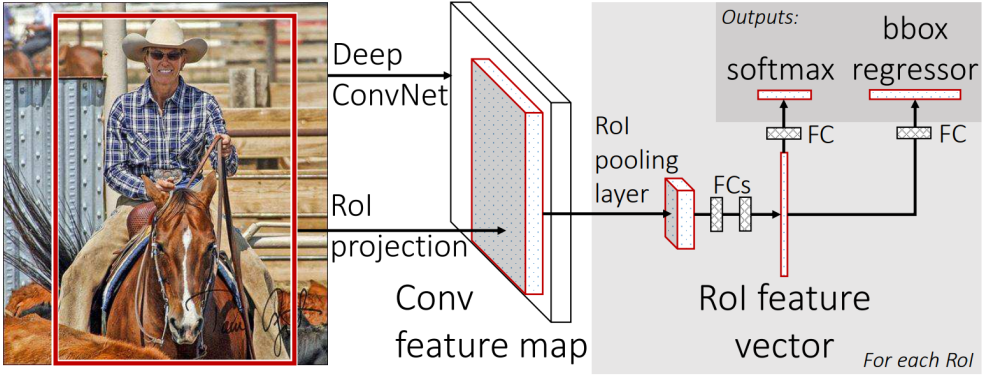
- Extract proposals via selective search [Uijlings et al.(2013)].
- Extract CNN features.
- Classify with an SVM.



(Fig. from [Girshick et al.(2014)])

Fast R-CNN [Girshick(2015)]

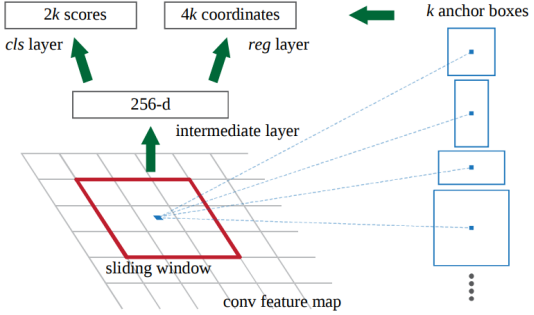
- Extract proposals via selective search.
- Extract features **and** classify with CNN.

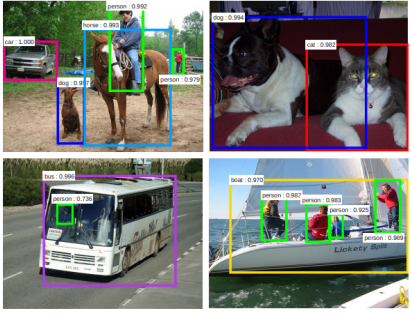


(Fig. from [Girshick(2015)])

Faster R-CNN [Ren et al.(2015)]

- Proposes novel RPN: Region Proposal Network – no more selective search.
- End2End: Extract proposals, features and classify with CNN.
- Note: PFDet [Akiba et al.(2018)] is based on this type of detector.



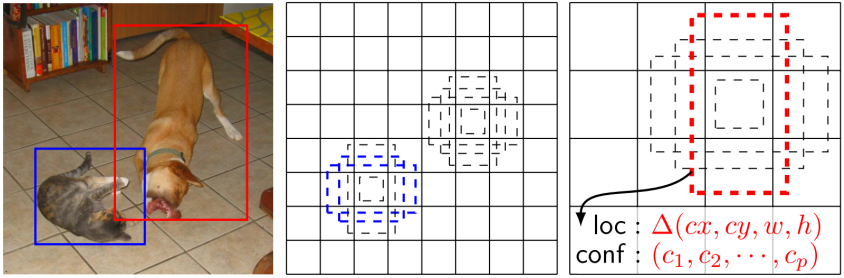


(Fig. from [Ren et al.(2015)])

Preferred Networks

Interlude: What is an anchor?

- Human-designed prior on object shapes.
- Designing anchors is deciding which and how many prior anchor shapes to use.
- Anchors are typically selected to be close in shape to typical objects. Upright rectangle for pedestrian, lying rectangle for car, etc.



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

(Fig. from [Liu et al.(2016)])

1 Motivation

2 History Before Deep Learning

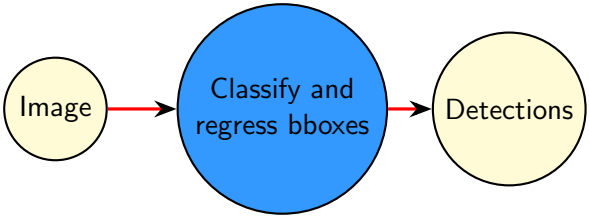
3 Two-stage Methods

4 Single-shot Methods

5 Anchor-free Methods

6 Problems and Summary

Single-shot Methods



- Unlike Faster R-CNN et al., only has a single stage.
- Can be thought of as that the RPN **both** localizes and classifies the object.
- **When to use?** Single-shot methods are generally fast and have moderate accuracy.
- We will look at two representative methods:
  - SSD: Single Shot MultiBox Detector
  - YOLO: You Only Look Once
- (Out of scope, but recommended reading: [Zhang et al.(2018), Lin et al.(2017b)])

SSD: Single Shot MultiBox Detector [Liu et al.(2016)]

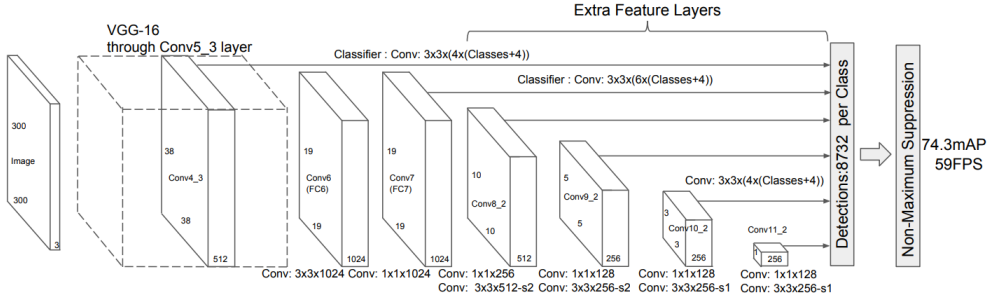
- At each feature scale, predict class and bounding box regression (via Huber loss).

  Linear scale target center x $\hat{g}_j^{cx}$

  Log scale target width $\hat{g}_j^{w}$

$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^{w} \qquad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^{h}$$

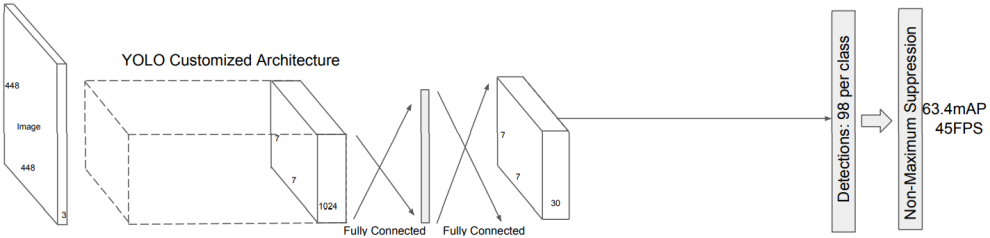
$$\hat{g}_j^{w} = \log\left(\frac{g_j^{w}}{d_i^{w}}\right) \qquad \hat{g}_j^{h} = \log\left(\frac{g_j^{h}}{d_i^{h}}\right)$$



(Fig. from [Liu et al.(2016)])

YOLO: You Only Look Once [Redmon et al.(2016)]

- Unlike SSD, much simpler. Just a single scale of features and fully connected layers.



- YOLOv2 [Redmon and Farhadi(2017)] makes the method more like SSD, removes fully connected layers (+ other tricks).

(Fig. from [Liu et al.(2016)])

1 Motivation

2 History Before Deep Learning

3 Two-stage Methods

4 Single-shot Methods

5 Anchor-free Methods

6 Problems and Summary

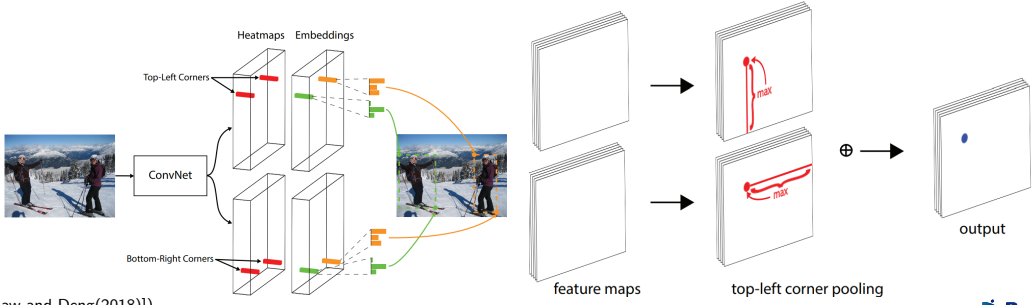
Anchor-free Methods

- Recently, researchers have tried to remove anchors from object detection methods.
- Why are anchors bad?
  - Human-designed prior. (cf. hand-crafted featured vs deep learning on ILSVRC in 2012)
  - Anchor-free methods provide more flexibility to leverage large-scale data.
- Why are we able to remove the anchors? Mainly thanks to progress in
  - deep keypoint detection research [Newell et al.(2017), Newell and Deng(2017)].
  - detection loss formulation, such as focal loss [Lin et al.(2017b)].
- Unlike previous architectures, more similar to semantic segmentation.
- **When to use?** Instead of single-shot methods for fast *and* accurate inference.
  - If you were thinking of using SSD for your project – think again.
  - Caveat: Two-stage methods tend to still be more accurate.

CornerNet: Detecting Objects as Paired Keypoints [Law and Deng(2018)]

- First competitive anchor-free method – detect corners of objects.
- Key contributions are corner pooling and improving focal loss for keypoint detection:

$$L_{det} = \frac{-1}{N} \sum_{c=1}^{C} \sum_{i=1}^{H} \sum_{j=1}^{W} \begin{cases} (1 - p_{cij})^{\alpha} \log (p_{cij}) & \text{if } y_{cij} = 1 \\ (1 - y_{cij})^{\beta} (p_{cij})^{\alpha} \log (1 - p_{cij}) & \text{otherwise} \end{cases}$$

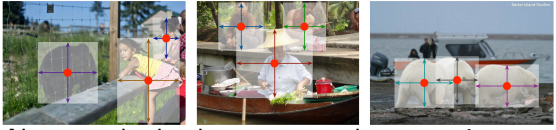


(Fig. from [Law and Deng(2018)])

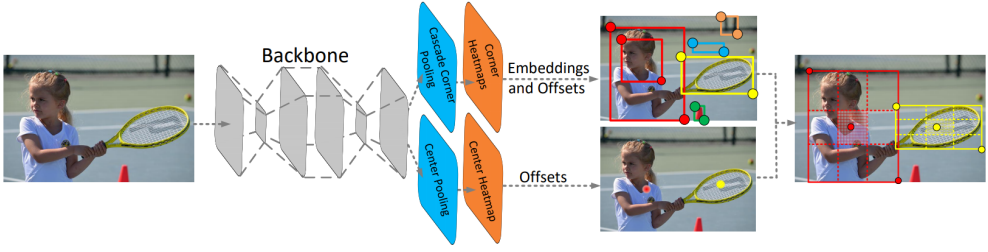
Preferred Networks

CenterNet(s): Objects as Points

- Found there are actually **two** CenterNet papers: posted on arXiv 1 day apart.
- Apr 16: Predict center, regress object size.



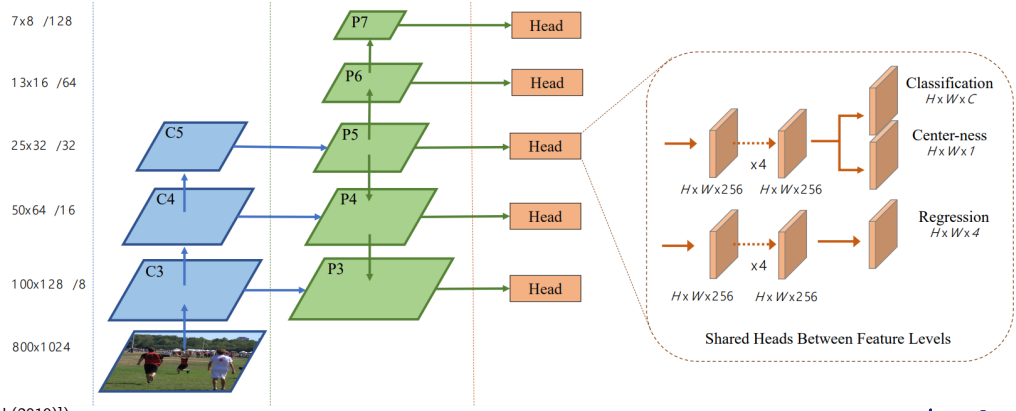
- Apr 17: Extend CornerNet to take both corners and centers into account.



(Figs. from [Zhou et al.(2019b), Duan et al.(2019)])

FCOS: Fully Convolutional One-Stage Object Detection [Tian et al.(2019)]

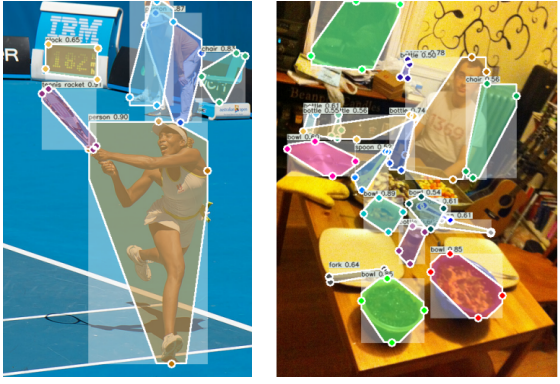
- Similar to the Apr 16 CenterNet – predict center, regress object size.



(Figs. from [Tian et al.(2019)])

Bottom-up Object Detection by Grouping Extreme and Center Points [Zhou et al.(2019a)]

- Detect not only bboxes, but convex octagons enclosing objects. Same author as Apr 16 CenterNet.



(Figs. from [Zhou et al.(2019a)])

Preferred Networks

1 Motivation

2 History Before Deep Learning

3 Two-stage Methods

4 Single-shot Methods

5 Anchor-free Methods

6 Problems and Summary

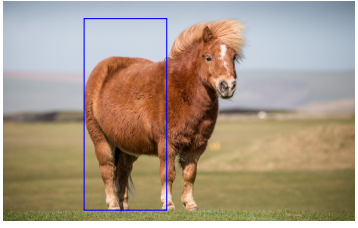
Problems for Current Object Detection Methods

- Hand-crafted post-processing (NMS) is still required due to correlated detections
  - Attempts at "truly" end-to-end object detectors exist, but not practical [Hu et al.(2018), Hosang et al.(2017)].
- Bounding box representation not optimal – remember the horse?

(Fig. from telegraph.co.uk)

Problems for Current Object Detection Methods

- Hand-crafted post-processing (NMS) is still required due to correlated detections
  - Attempts at "truly" end-to-end object detectors exist, but not practical [Hu et al.(2018), Hosang et al.(2017)].
- Bounding box representation not optimal – remember the horse?



- You can miss half the horse – and still considered a correct detection!

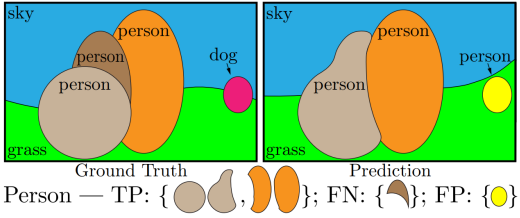
(Fig. from telegraph.co.uk)

Problems – and solutions?

- Bounding box representation and hand-crafted NMS not optimal
  - New field, panoptic segmentation, tackles this [Kirillov et al.(2019)].
  - No need for NMS, as the task requires explaining each pixel, including background classes.
  - Proposes new metric PQ: Panoptic Quality for replacing IoU metric.

(a) image (b) semantic segmentation

(c) instance segmentation (d) panoptic segmentation



(Fig. from [Kirillov et al.(2019)])

Summary

- Deep learning has resulted in a significant ($\sim 150\%$) improvement in object detection.
- Three main architectures of deep object detectors
  - Two-stage
  - Single-shot
  - Anchor-free (recent research!)
- Issues
  - Hand-crafted post-processing (NMS) is still required due to correlated detections.
  - Issue of using bounding boxes to represent objects remains.
  - $\mapsto$ Recent new field of panoptic segmentation aims to mitigate these issues.
- Takeaway message:
  - Do not use SSD! There are much better methods available for your project.
  - Anchor-free methods are promising for fast and accurate object detection.

References I

T. Akiba et al.

Pfdet: 2nd place solution to open images challenge 2018 object detection track.
arXiv preprint arXiv:1809.00778, 2018.

N. Bodla et al.

Soft-nms–improving object detection with one line of code.
In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5561–5569, 2017.

N. Dalal and B. Triggs.

Histograms of oriented gradients for human detection.
In *International Conference on computer vision & Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE Computer Society, 2005.

K. Duan et al.

Centernet: Object detection with keypoint triplets.
arXiv preprint arXiv:1904.08189, 2019.

P. Felzenszwalb et al.

A discriminatively trained, multiscale, deformable part model.
In *CVPR*, 2008.

R. Girshick.

Fast r-cnn.
In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

References II

R. Girshick et al.

Rich feature hierarchies for accurate object detection and semantic segmentation.

In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

J. Hosang et al.

Learning non-maximum suppression.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4507–4515, 2017.

H. Hu et al.

Relation networks for object detection.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3588–3597, 2018.

A. Kirillov et al.

Panoptic segmentation.

In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

A. Krizhevsky et al.

Imagenet classification with deep convolutional neural networks.

In *Advances in neural information processing systems*, pages 1097–1105, 2012.

H. Law and J. Deng.

Cornernet: Detecting objects as paired keypoints.

In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018.

References III

Y. Li et al.

Scale-aware trident networks for object detection.

arXiv preprint arXiv:1901.01892, 2019.

T.-Y. Lin et al.

Feature pyramid networks for object detection.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017a.

T.-Y. Lin et al.

Focal loss for dense object detection.

In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017b.

W. Liu et al.

Ssd: Single shot multibox detector.

In *European conference on computer vision*, pages 21–37. Springer, 2016.

X. Lu et al.

Grid r-cnn.

In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

A. Newell and J. Deng.

Pixels to graphs by associative embedding.

In *Advances in neural information processing systems*, pages 2171–2180, 2017.

References IV

A. Newell et al.

Associative embedding: End-to-end learning for joint detection and grouping.
In *Advances in Neural Information Processing Systems*, pages 2277–2287, 2017.

J. Redmon and A. Farhadi.

Yolo9000: better, faster, stronger.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

J. Redmon et al.

You only look once: Unified, real-time object detection.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

S. Ren et al.

Faster r-cnn: Towards real-time object detection with region proposal networks.
In *Advances in neural information processing systems*, pages 91–99, 2015.

B. Singh et al.

Sniper: Efficient multi-scale training.
In *Advances in Neural Information Processing Systems*, pages 9310–9320, 2018.

Z. Tian et al.

Fcos: Fully convolutional one-stage object detection.
arXiv preprint arXiv:1904.01355, 2019.

References V

J. Uijlings et al.
Selective search for object recognition.
International journal of computer vision, 104(2):154–171, 2013.

P. Viola and M. Jones.
Rapid object detection using a boosted cascade of simple features.
CVPR, 2001.

S. Zhang et al.
Single-shot refinement neural network for object detection.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4203–4212, 2018.

H. Zhou et al.
Cad: Scale invariant framework for real-time object detection.
In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

X. Zhou et al.
Bottom-up object detection by grouping extreme and center points.
In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019a.

X. Zhou et al.
Objects as points.
arXiv preprint arXiv:1904.07850, 2019b.

References VI

Z. Zou et al.
Object detection in 20 years: A survey.
arXiv preprint arXiv:1905.05055, 2019.

Thank you for listening! Questions?