How Object Detection Evolved: From Region Proposals and Haar Cascades to Zero-Shot Techniques

Andrii Polukhin

pandrii000@gmail.com LinkedIn

Data Science UA

June 14, 2023

About me

- ML Engineer in Data Science UA and Samba.TV.
- Mentor in PRJCTR and 10:11.
- Writing about AI in Telegram, Medium, and personal website.
- Master's degree in Mathematics.

Please shoot me an email if you are interested in machine learning collaboration or if you have a cool concept to put into practice.

Table Of Contents

Start with the Basics

Problem Definition

What objects are where?

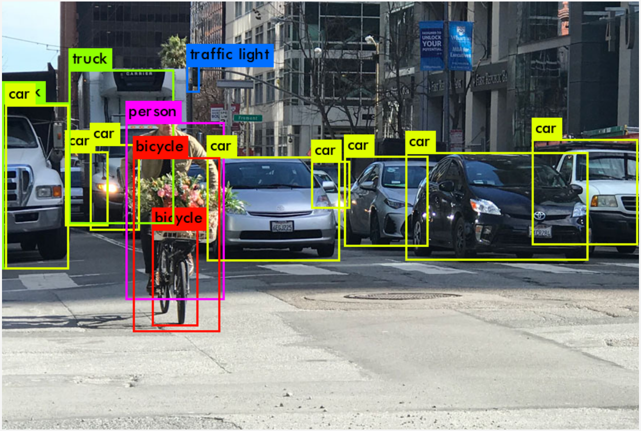


Figure: Object Detection Example. (source)

Importance of Object Detection

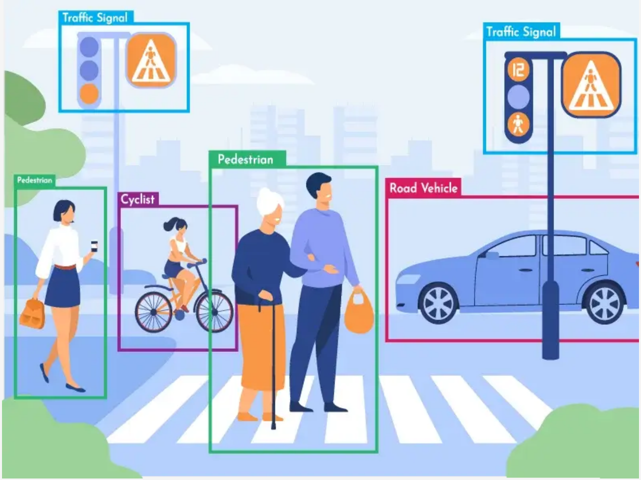


Figure: Object Detection In Real World. (source)

Surveillance Systems

Figure: Object Detection In Surveillance Systems. (source)

Autonomous Vehicles

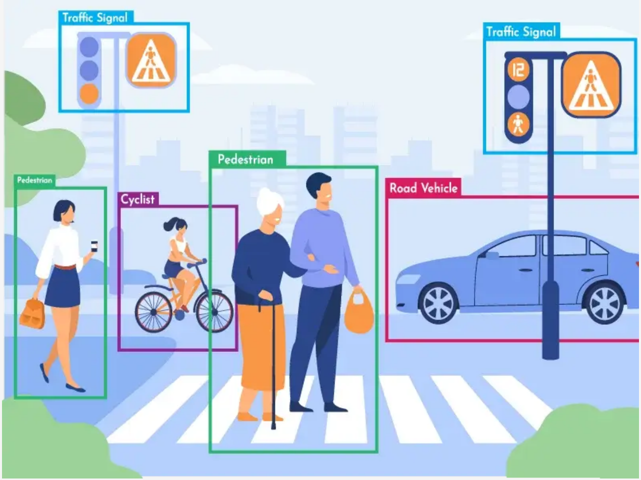


Figure: Object Detection In Real World. (source)

Medical Imaging

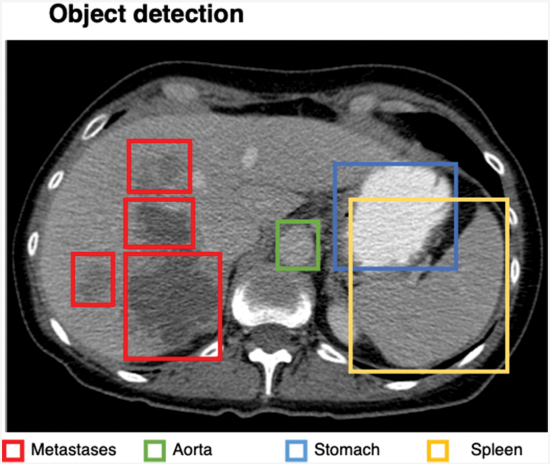


Figure: Object Detection In Medical Imaging. (source)

Retail (Automated Checkout)

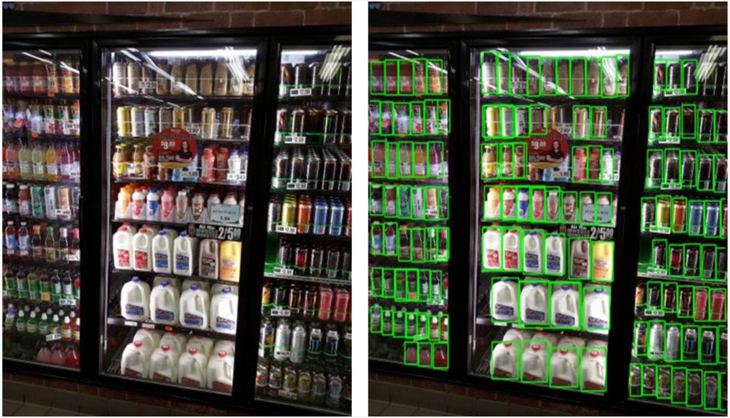


Figure: Object Detection In Retail. (source)

Agriculture (Crop Monitoring)

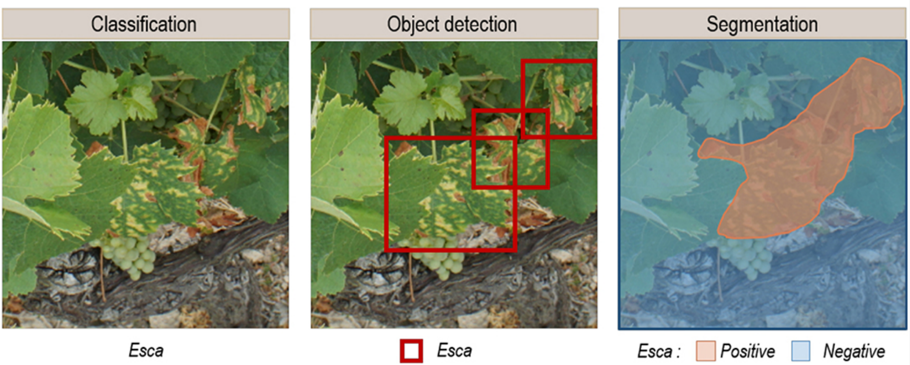


Figure: Object Detection In Agriculture. (source)

A Road Maps of Object Detection

Road Map (general)

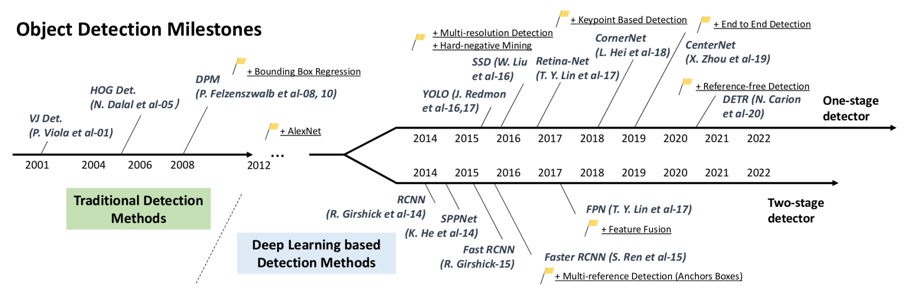


Figure: (source)

Road Map (more traditional methods)

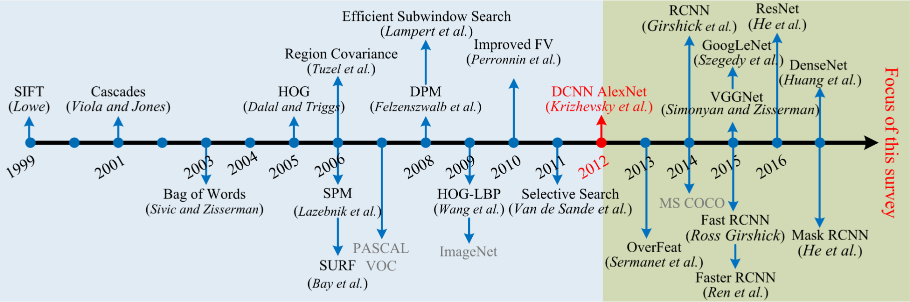


Figure: (source)

Road Map (deep learning methods)

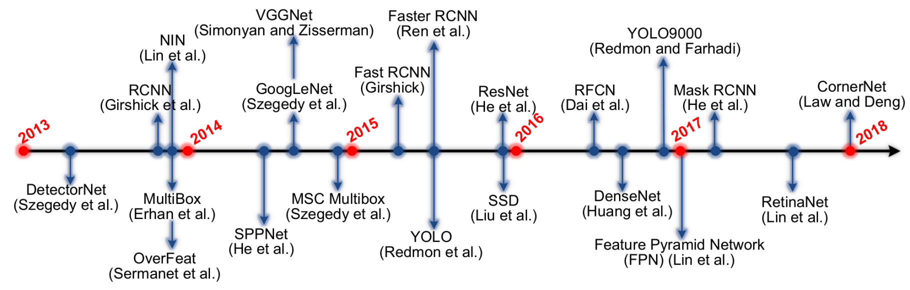


Figure: (source)

Object Detection Metrics Improvements

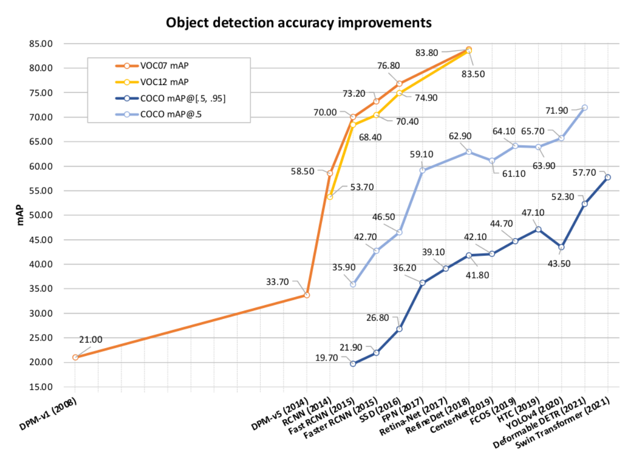


Figure: Accuracy improvement of object detection on VOC07, VOC12 and MS-COCO datasets. (source)

Traditional Detection Methods

Traditional Detection Methods

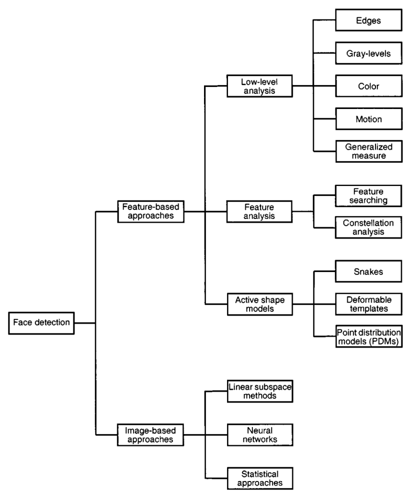


Figure: Face Detection Methods in 2001. (source)

Viola-Jones Detectors (2001)

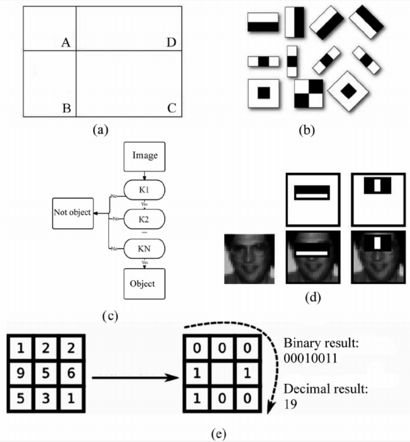


Figure: Viola-Jones algorithm parts: ( ) combination of regions, (b) Haar Features, (c) cascade classifier, (d) Haar feature applies to the image, and (e) LBP feature. (source)

HOG Detector (2005)

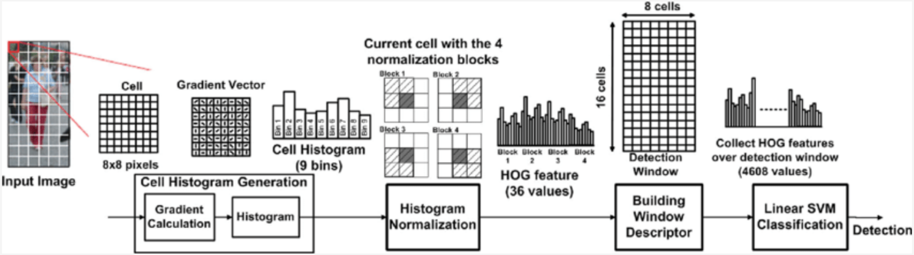


Figure: Object detection algorithm using HOG features. (source)

Part-based Approaches

- Deformable Part-based Model (2008)
- Implicit Shape Model (2008)

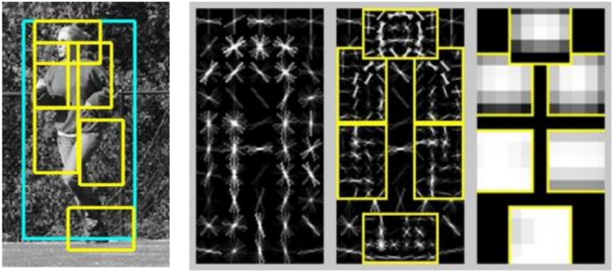


Figure 6: An example detection obtained with the Deformable Part Model proposed by Felzenszwalb et al. (2008). The DPM comprises a coarse as well as multiple high resolution models and a spatial constellation model for constraining the location of each part. Adapted from Felzenszwalb et al. (2008).

Figure: (source)

Deep Learning-based Detection Methods

Deep Learning-based Detection Methods

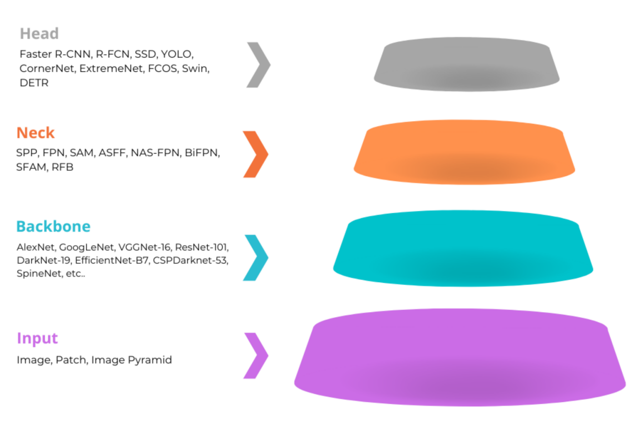
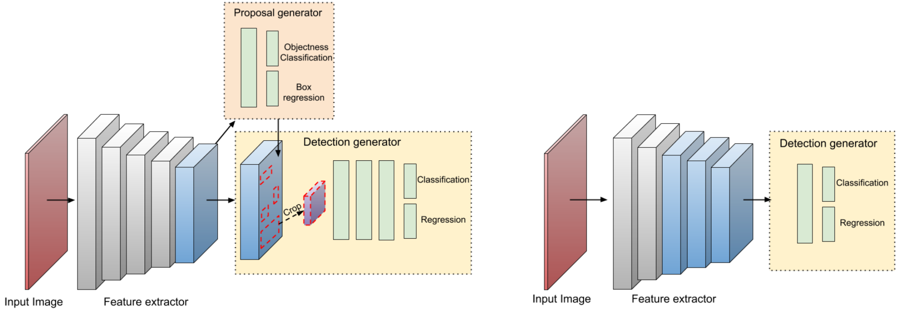


Figure: The components of an ordinary object detection model. (source)

Two- and One- Stage Detectors



(**a**) Two-stage Faster R-CNN

(**b**) One-stage RetinaNet

Figure: Deep learning object detection meta-architectures. (source)

Two-Stage Detectors

RCNN (2014)

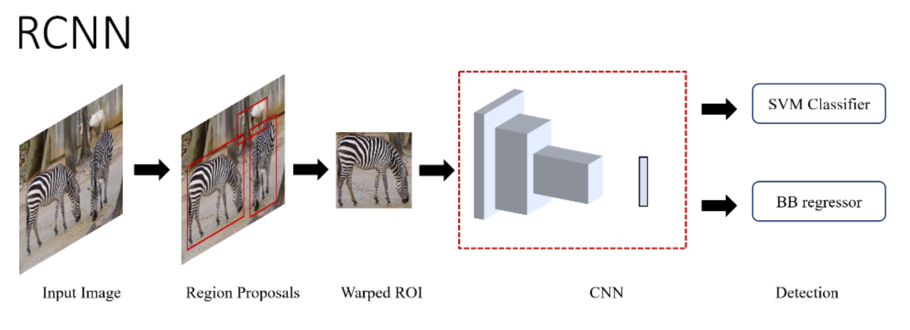


Figure: Illustration of the internal architecture of RCNN. (source)

Fast RCNN (2015)

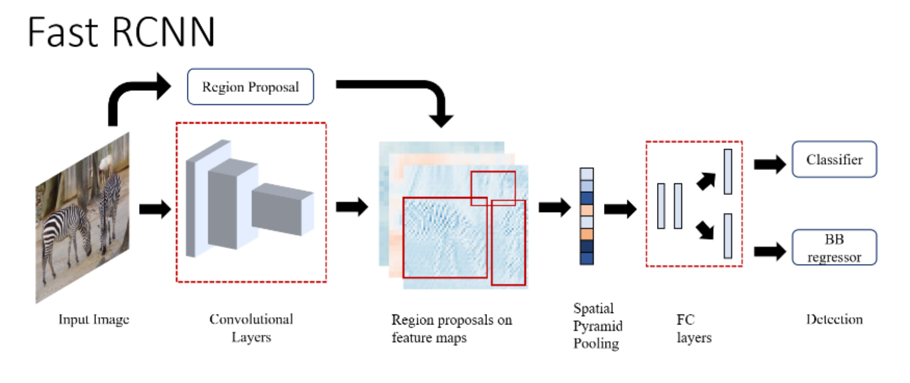


Figure: Illustration of the internal architecture of Fast RCNN. (source)

Faster RCNN (2015)

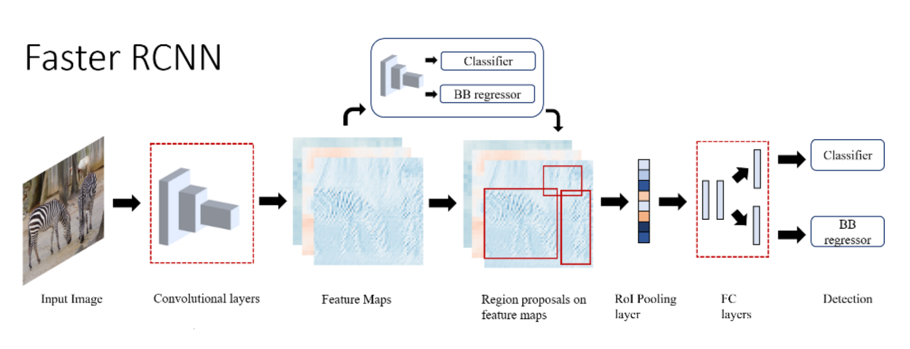


Figure: Illustration of the internal architecture of Faster RCNN. (source)

FPN (2017)

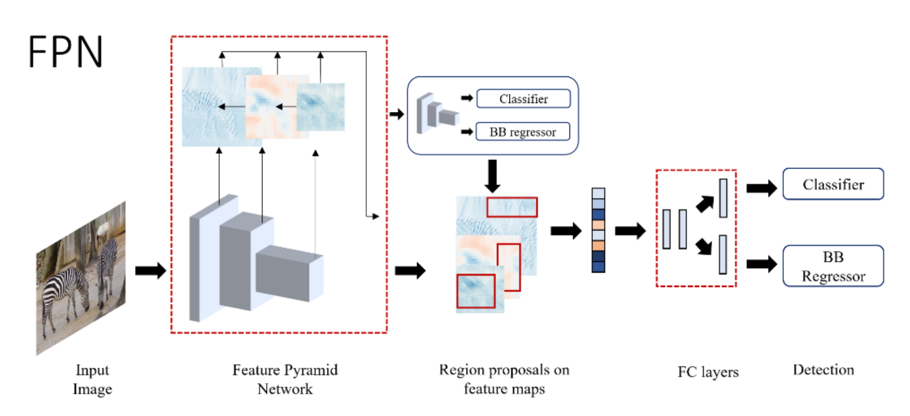


Figure: Illustration of the internal architecture of Feature Pyramid Networks (FPN). (source)

Backbones

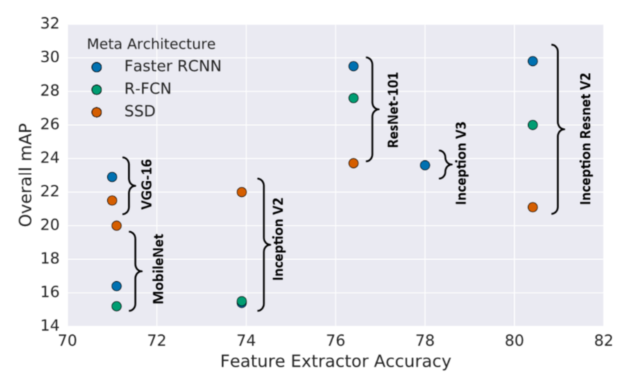


Figure: A comparison of detection accuracy of three detectors: Faster RCNN, R-FCN and SSD on MS-COCO dataset with different detection backbones. (source)

One-Stage Detectors

YOLO (2015)

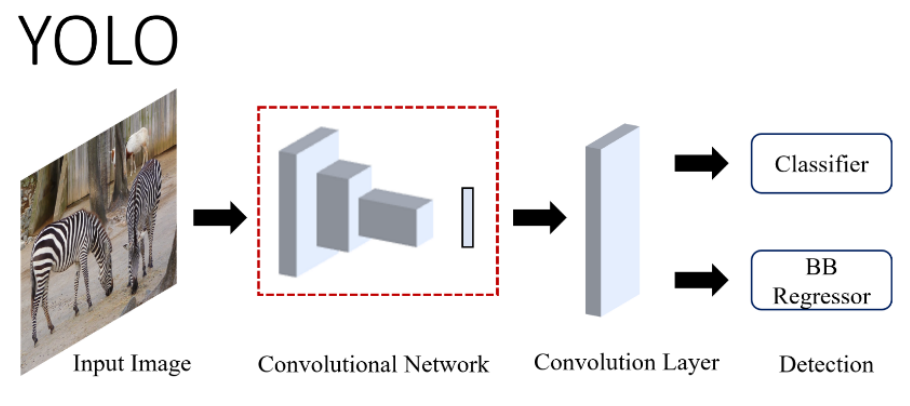


Figure: Illustration of the internal architecture of You Only Look Once (YOLO). (source)

SSD (2015)

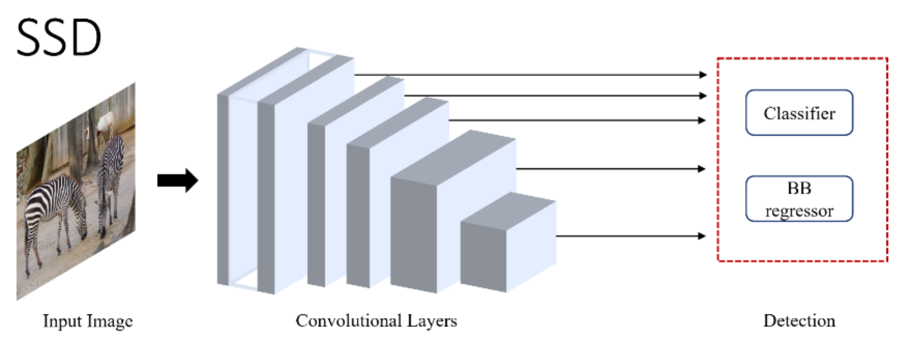


Figure: Illustration of the internal architecture of Single Shot MultiBox Detector (SSD). (source)

RetinaNet (2017)

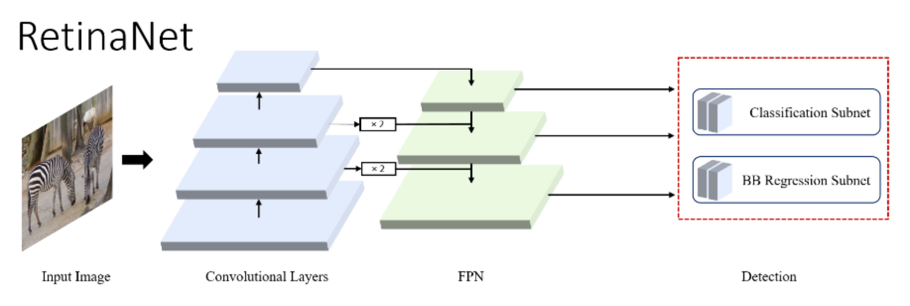


Figure: Illustration of the internal architecture of RetinaNet. (source)

CenterNet (2019)

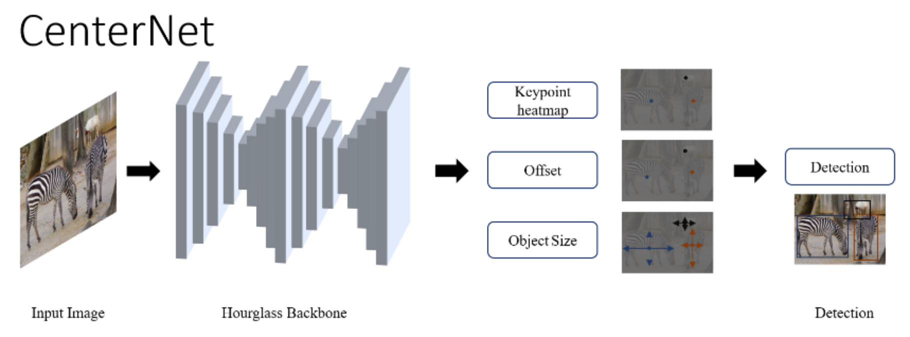


Figure: Illustration of the internal architecture of CenterNet. (source)

Object Detectors by Category

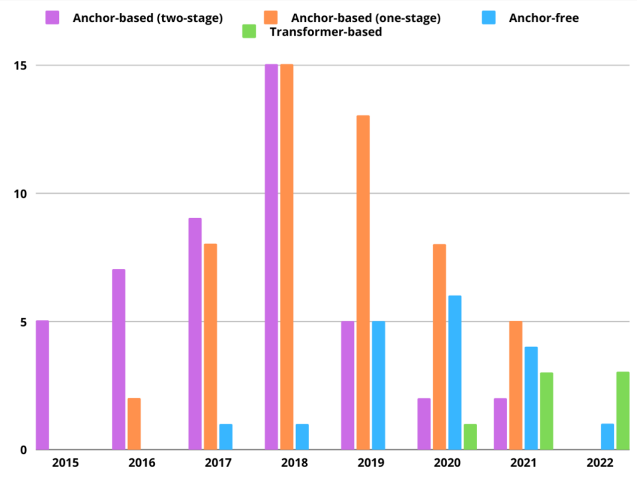


Figure: The number of state-of-the-art object detectors, by category, published in top journals and evaluated on MS-COCO. (source)

Transformer-based Detectors

Transformer-based Detectors

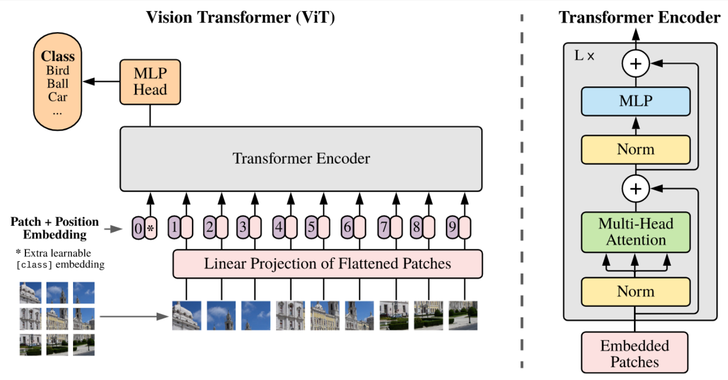


Figure: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. (source)

DETR (2020)

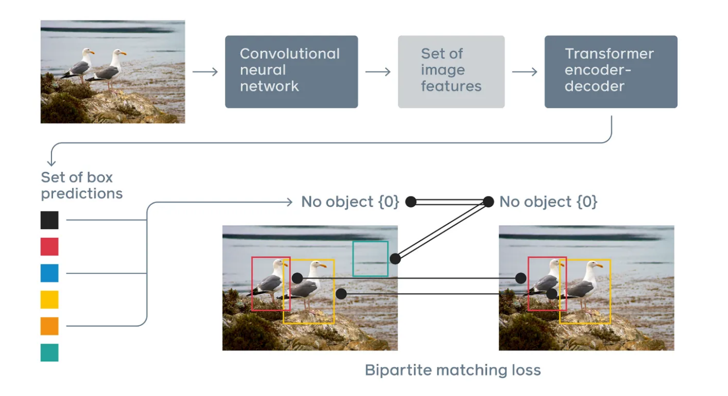


Figure: DETR directly predicts (in parallel) the final set of detections by combining a common CNN with a transformer architecture. During training, bipartite matching uniquely assigns predictions with ground truth boxes. Prediction with no match should yield a “no object” class prediction. (source)

Swin (2021)

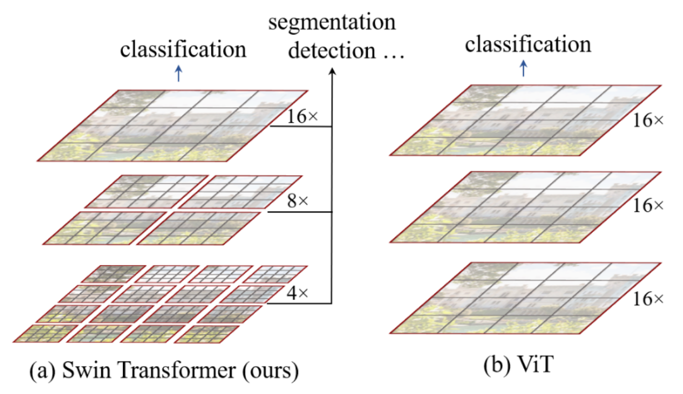


Figure: The proposed Swin Transformer builds hierarchical feature maps by merging image patches (shown in gray) in deeper layers and has linear computation complexity to input image size due to computation of self-attention only within each local window (shown in red). (source)

Non-Max Suppression (NMS)

Non-Max Suppression (NMS)

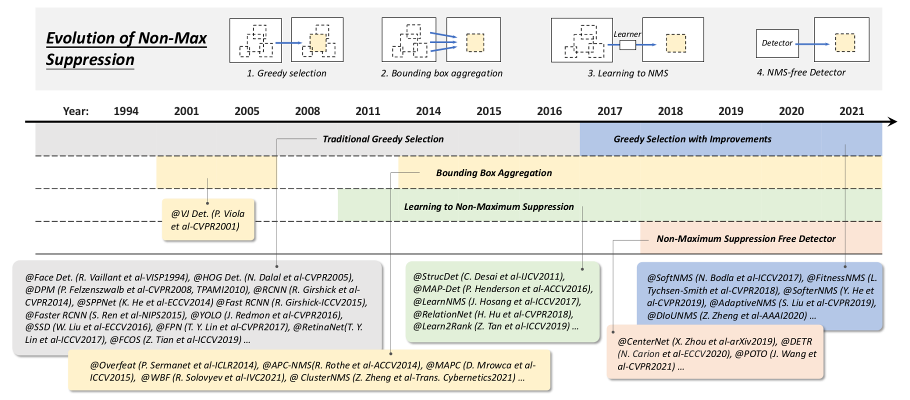


Figure: Evolution of non-max suppression (NMS) techniques in object detection from 1994 to 2021: 1) Greedy selection, 2) Bounding box aggregation, 3) Learning to NMS, and 4) NMS-free detection. (source)

(Zero | One | Few) - Shot Object Detection

Multimodality

Multimodality

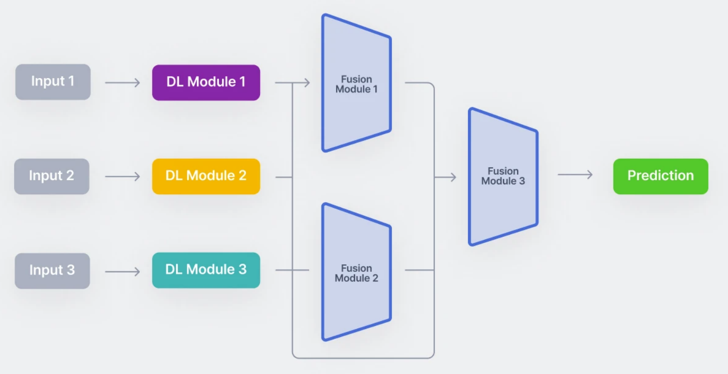


Figure: Workflow of a typical multimodal. Three unimodal neural networks encode the different input modalities independently. After feature extraction, fusion modules combine the different modalities (optionally in pairs), and finally, the fused features are inserted into a classification network. (source)

(Zero | One | Few) - Shot Object Detection

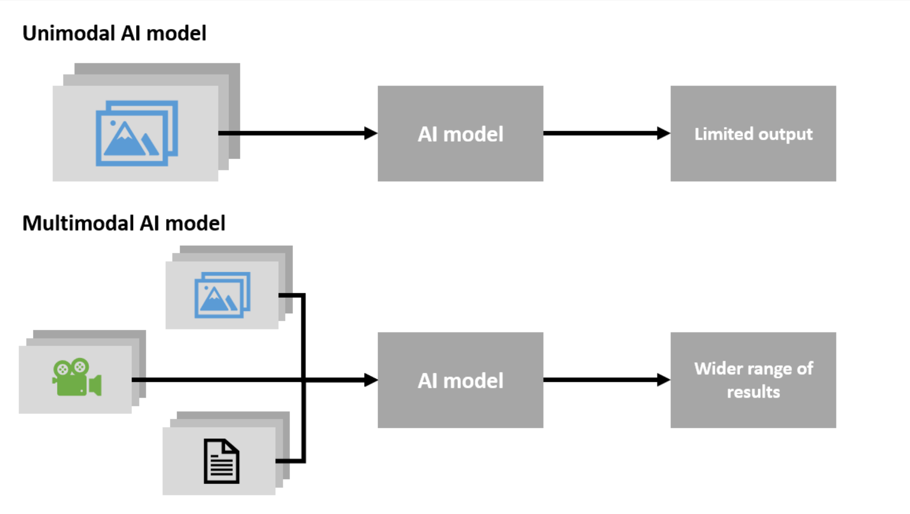


Figure: (source)

CLIP (2021)

CLIP adds **image-text connection** to understand **the content** of the image.

CLIP (2021)

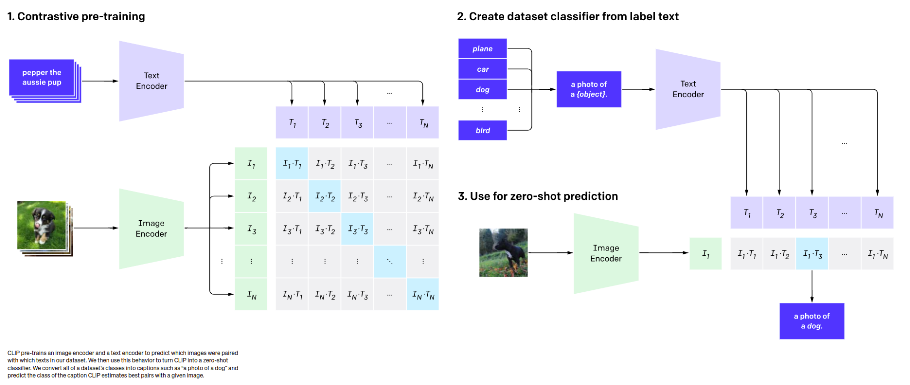


Figure: CLIP by OpenAI. (source)

OWL-ViT (2022)

OWL-ViT adds **image-level patches** to understand **the location** of the objects.

OWL-ViT (2022)

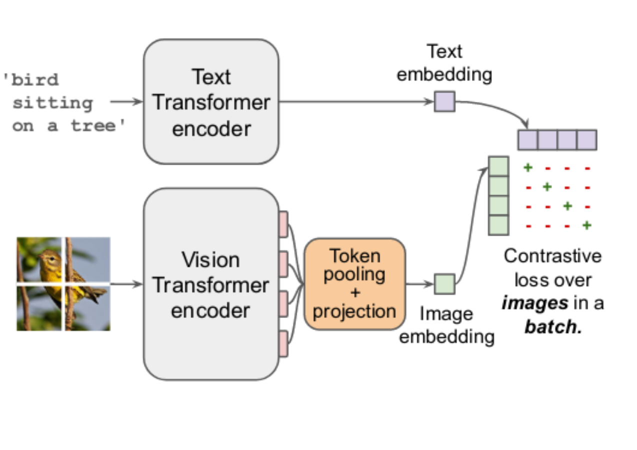


Figure: OWL-ViT: Image-level contrastive pre-training. (source)

OWL-ViT (2022)

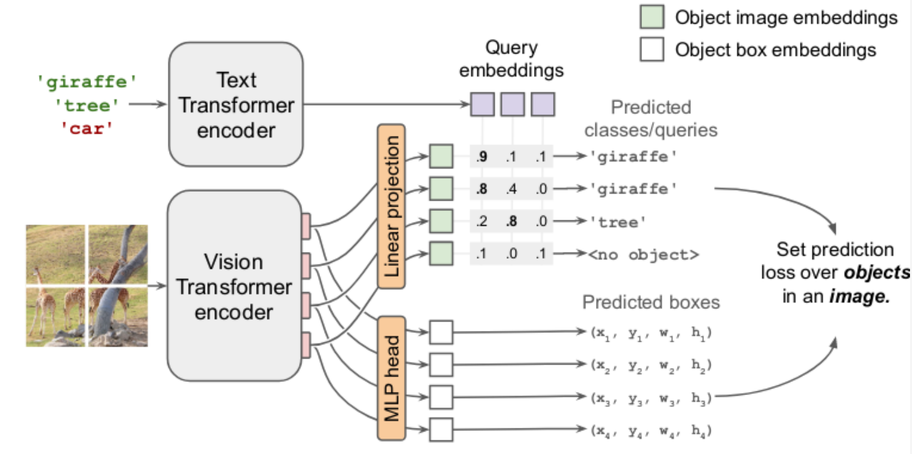


Figure: OWL-ViT: Transfer to open-vocabulary detection. (source)

OWL-ViT (2022)

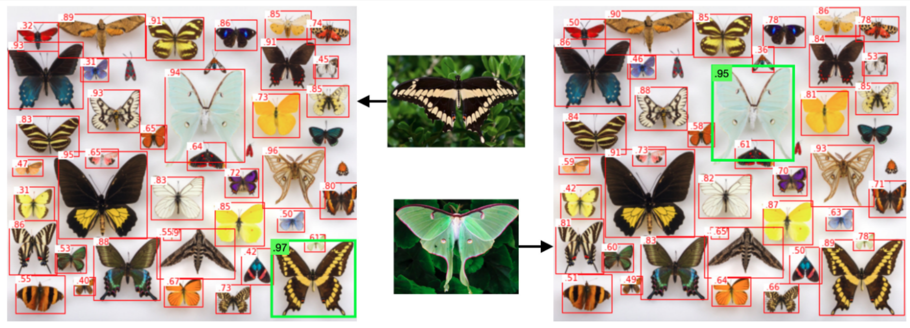


Figure: OWL-ViT: Example of one-shot image-conditioned detection. (source)

GLIP (2022)

GLIP adds **word-level understanding** to find the objects **by the semantics** of the prompt.

GLIP (2022)

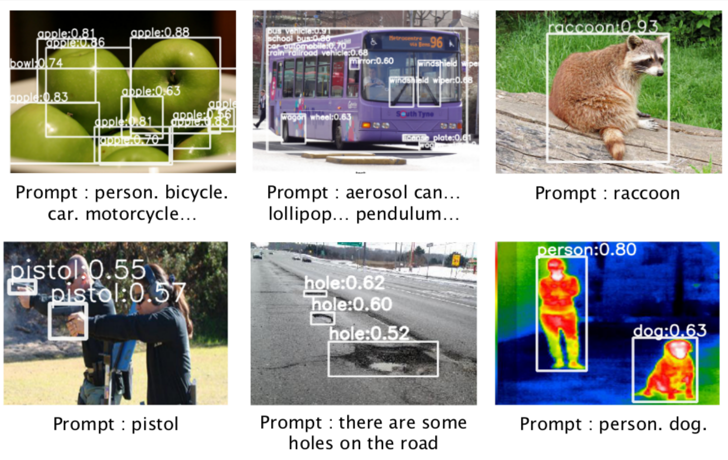


Figure: GLIP zero-shot transfers to various detection tasks, by writing the categories of interest into a text prompt.

GLIP (2022)

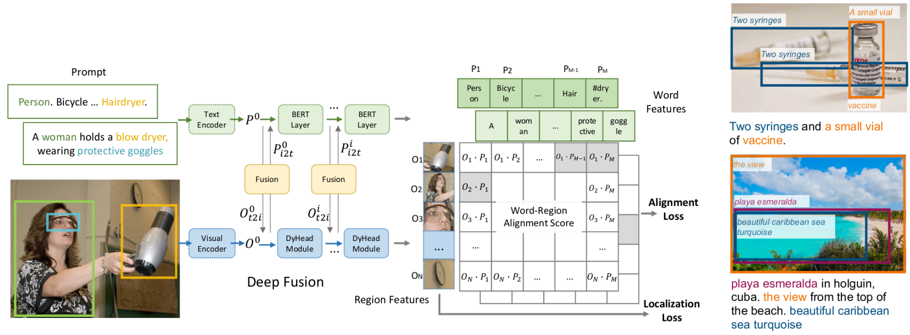


Figure: We reformulate detection as a grounding task by aligning each region/box to phrases in a text prompt. We add the cross-modality deep fusion to early fuse information from two modalities and to learn a language-aware visual representation. (source)

GLIP (2022)

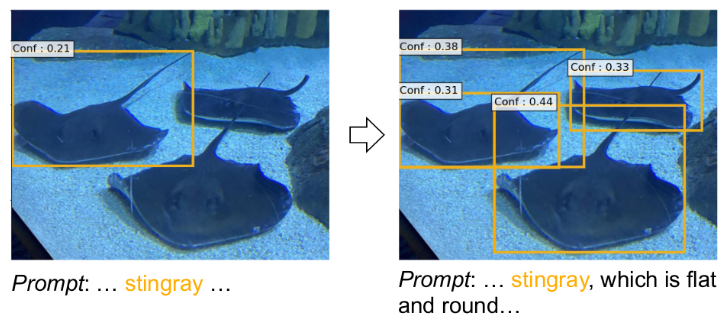


Figure: A manual prompt tuning example from the Aquarium dataset in ODinW. Given an expressive prompt ("flat and round"), zero-shot GLIP can detect the novel entity "stingray" better. (source)

Segment Anything (2023)

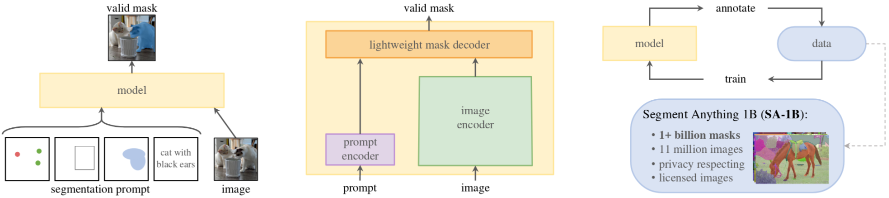
Segment Anything (SAM) adds **masks** to see **the pixel-level location** of the objects.

Segment Anything (2023)

Segment Anything

Alexander Kirillov[1,2,4] Eric Mintun[2] Nikhila Ravi[1,2] Hanzi Mao[2] Chloe Rolland[3] Laura Gustafson[3]
Tete Xiao[3] Spencer Whitehead Alexander C. Berg Wan-Yen Lo Piotr Dollár[4] Ross Girshick[4]
[1]project lead [2]joint first author [3]equal contribution [4]directional lead

Meta AI Research, FAIR



(a) **Task**: promptable segmentation (b) **Model**: Segment Anything Model (**SAM**) (c) **Data**: data engine (top) & dataset (bottom)

Figure 1: We aim to build a foundation model for segmentation by introducing three interconnected components: a promptable segmentation *task*, a segmentation *model* (SAM) that powers data annotation and enables zero-shot transfer to a range of tasks via prompt engineering, and a *data* engine for collecting SA-1B, our dataset of over 1 billion masks.

Figure: (source)

Segment Anything (2023)

Figure: (source)

Good Visual Tokenizers (2023)

GVT adds **usage of the Large Language Model** to **investigate** the image with the text.

Good Visual Tokenizers (2023)

Figure: Different tasks require visual understanding of different perspectives. Mainstream vision-language tasks, e.g., (a) VQA and (b) Image Captioning mainly focus on semantic understanding of the image. In this work, we also study two fine-grained visual understanding tasks: (c) Object Counting (OC) and (d) Multi-Class Identification (MCI). (source)

Good Visual Tokenizers (2023)

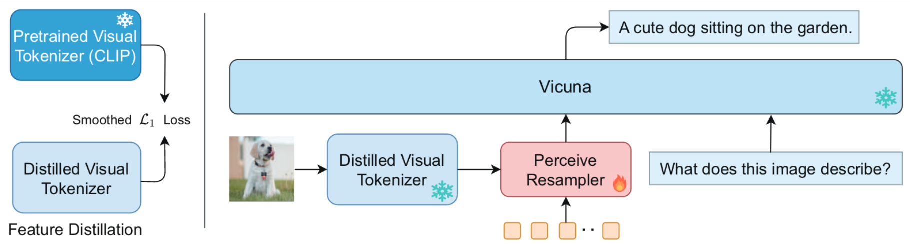


Figure: Framework of GVT. First distill the features of a pretrained CLIP via smoothed L1 loss. Then, use it to encode images into a set of tokens, which are fed into the Perceiver Resampler as soft prompts. Together with language instructions, these prompts are fed into LLM to generate responses. Only the Perceiver Resampler is optimized in this process. (source)

Good Visual Tokenizers (2023)

1. CLIP adds **image-text connection** to understand **the content** of the image.
2. OWL-ViT adds **image-level patches** to understand **the location** of the objects.
3. GLIP adds **word-level understanding** to find the objects **by the semantics** of the prompt.
4. SAM adds **masks** to see **the pixel-level** location of the objects.
5. GVT adds **usage of the Large Language Model** to **investigate** the image with the text.

Q&A

Thank you for your attention!
I am ready to answer your questions now.