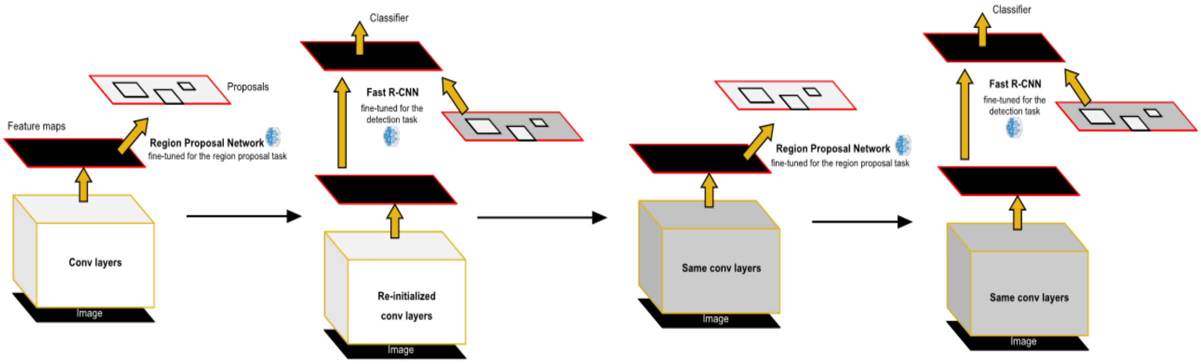
Object Detection Methods Literature Review

Context:

Computer vision and artificial intelligence techniques are being developed to improve the detection and characterization of abnormalities in digital mammography images. When radiologists are provided with this technology-assisted information, it holds the promise of potentially enhancing their performance in mammographic detection, making it potentially more accurate and faster. In this study, we will review various object detection methods with the aim of contributing to the ongoing development of this technology.

1. **Faster R-CNN** [1]

Faster R-CNN is a cost-effective solution for object detection. The two main components are a Region Proposal Network (RPN) used to generate region proposals and the architecture fast R-CNN. It has been proven more accurate than the previous algorithms (R-CNN and Fast R-CNN). The RPN shares convolutional layers with the Fast R-CNN which has enabled to reduce region proposal time from 2s to 10ms, because it shares layers with the following detection stages.



Feature maps
Proposals
Region Proposal Network
fine-tuned for the region proposal task
Conv layers
Image
Classifier
Fast R-CNN
fine-tuned for the detection task
Re-initialized conv layers
Image
Region Proposal Network
fine-tuned for the region proposal task
Same conv layers
Image
Classifier
Fast R-CNN
fine-tuned for the detection task
Same conv layers
Image

__Figure 1__: Faster R-CNN alternating training approach

The architecture of the faster R-CNN is composed firstly of a deep fully convolutional network which proposes the different regions, then used by the Fast R-CNN detector. A backbone CNN generates the feature map that will feed The Region Proposal Network. The RPN is two-headed and learns whether an object is present and its localization.

To generate Region of Interest (RoI), rectangle from many sizes and ratios called anchors are placed on the output feature map. The function loss composed of both a regression loss and a classification loss is minimized using back propagation and stochastic gradient descent (SGD). The output of the classification head of the RPN consists in the presence of the object and the regression head outputs the coordinates of the predicted object bounding box.

The detection network (Fast R-CNN) is trained separately from the RPN, on the output RoIs. The network tuned by Fast R-CNN is used to initialize the RPN. At that point, the two networks share convolutional layers. The RPN is then re-trained using these layers. This process (Alternative training) is then iterated several times.

2. **Mask R-CNN** [2]

Mask R- CNN extends Faster R-CNN by solving instance segmentation problem. In object recognition, generating high quality segmentation mask is a difficult task and Mask R-CNN shows a simple and efficient improvement of well-known and documented architectures. The work relies on previous baseline systems, namely Faster R-CNN, and Fully Convolutional Network (FCN).

Mask R-CNN adds to the previous Faster R-CNN architecture a branch for predicting segmentation mask on each Region of Interest (RoI). This branch is parallel with the existing branches for classification and bounding box regression in the RPN. This parallel branch is a Fully connected branch (FCN) performing instance segmentation in a pixel-to-pixel manner.

In Faster R-CNN, boxes are predicted based on a feature map which relies on spatial quantization and creates a misalignment. The RoIAlign layer proposed in the Mask R-CNN solution preserves exact spatial locations. In this architecture, mask and class prediction are decoupled. The parallel prediction of masks and class labels is a simple and efficient method compared to multiple-stage cascade architectures.

The approach is instance-first and opposed to segmentation-first methods: Classification does not depend on mask predictions. Compared to Faster R-CNN, the loss function is slightly modifieded by adding a binary cross entropy loss for the mask. The FCN used for mask prediction has proven itself more accurate than convolution layers. Different backbone also has been studied and Feature Pyramid Network (FPN) appears also more effective as a backbone as previous Restnet solution for Faster R-CNN. The mask branch has a straightforward architecture unlike the FPN. Mask R-CNN has shown a 5-points AP improvement compared to FCIS+++ solution on Coco test images and therefore appears a great solution for instance segmentation.

3. **YOLO** [3]

YOLO, which stands for "You Only Look Once" is an object detection algorithm. It was first introduced by Joseph Redmon and Santosh Divvala in 2016. YOLO revolutionized object detection by significantly improving both speed and accuracy compared to traditional methods.

Main idea is to process an entire image in a single pass through a neural network. This makes YOLO much faster than sliding window and region proposal-based techniques that requires multiple passes such as the R-CNN approach.

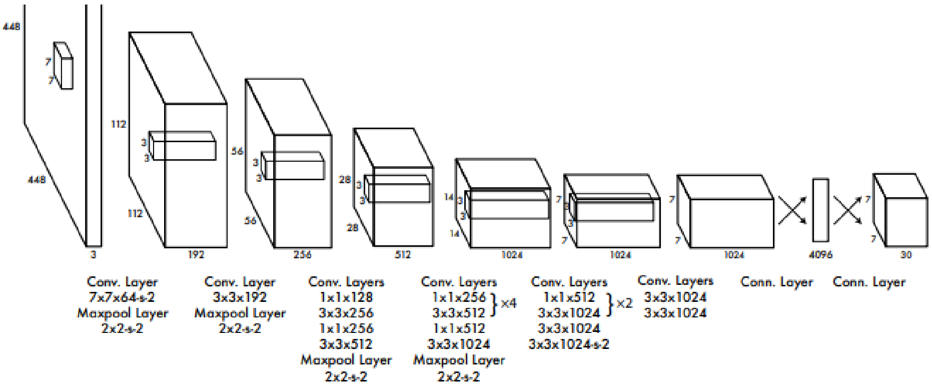


***Figure 2**: **The Architecture.** The detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1 × 1 convolutional layers reduce the features space from preceding layers. The convolutional layers are pretrained on the ImageNet classification task at half the resolution (224 × 224 input image) and then double the resolution for detection*

YOLO’s speed and reduced computation make it suitable for real-time object detection and applications like autonomous driving, surveillance, and robotics. The architecture struggles to detect small objects and objects close to each other because of the spatial limitations of the bounding boxes. YOLO divides the input image into an SxS grid and realize the prediction within each grid cell. Each grid cell predicts B bounding boxes and determine its characteristics: position of the center of the box relative to the grid, box’s width and height, IoU score and C conditional class probabilities.
The output prediction then consists of an SxSxBx(5+C) tensor.

This grid-based approach helps in predicting multiple bounding boxes and class probabilities for those boxes and detecting objects at different scales and locations while managing an overall high accuracy with the reduction of background errors. These bounding boxes can be bigger than the size of the grid cells.
To keep only the few most relevant boxes, the non-max suppression algorithm is applied to generate the final prediction.

4. **RetinaNet** [4]

Contrary to the previous network we’ve seen (Faster R-CNN, Mask R-CNN) and just like YOLO, RetinaNet is a one-stage object detector, which means it directly tries to find a fitting bounding box among all possible boxes, whereas two-stages object detectors use some proposal mechanism to reduce that number. While YOLO traded a lot of accuracy for the stake of speed, RetinaNet aims at catching up two-stages networks in terms of accuracy while maintaining higher speed.

To tackle the issues faced by one-stage networks, the authors engineered a new loss designed to force the network to dynamically train examples that are hard to classify. The most obvious solution is to adapt the commonly used cross-entropy and to weigh its values differently for each class. RetinaNet’s innovation is named Focal Loss and weighs the categorical cross-entropy dynamically. This means that the more a sample is predicted correctly, the more its loss is going to get shrunk to 0. This way, samples of background that is very easy to classify almost don’t contribute to the loss and let hard examples send a signal for the network to learn properly. The Focal loss reads:

$$p_t = \begin{cases} p & if\ y = 1 \\ 1 - p & otherwise \end{cases}$$

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

By choosing $\gamma = 0$ we get the classic cross-entropy. The more we increase γ the less easy-to-classify samples will contribute to the loss. The authors have found that $\gamma = 2$ yields the best results. This way, a class with $p_t = 0.9$ will contribute 100 times less to the loss than it should have without that modulating factor whereas samples with $p_t \leq 0.5$ will be scaled down by at most 4.

Usually, neural Networks have their weights initialized so that the probability of predicting each class is equal. In the case of object detection in RetinaNet, we use a prior so that the probability of predicting the rare class is low. This will ensure that we don’t downscale rare sample’s loss too much and improve stability.

5. **DEtection Transformers (DETR)** [5]

DETR is an object detection algorithm using an encoder-detector architecture based on transformers. It was made to simplify the complicated postprocessing pipelines using direct set prediction problem. Self-attention mechanisms of transformers enable dropping of components requiring prior knowledge like special anchors or non-maximal suppression. DETR can therefore be reproduced easily in different frameworks. Main features are bipartite matching loss and transformers with parallel decoding.

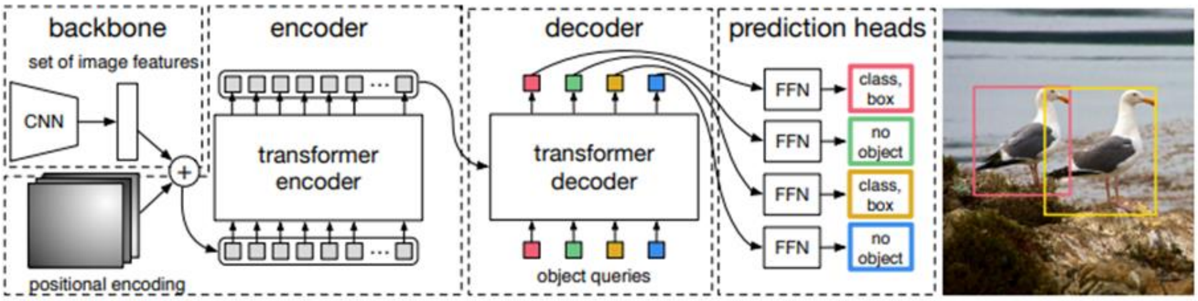


Figure 3: DETR directly predicts (in parallel) the final set of detections by combining a common CNN with a transformer architecture. During training, bipartite matching uniquely assigns predictions with ground truth boxes. Prediction with no match should yield a "no object" (Ø) class prediction

The three main components are CNN backbone to extract a compact feature representation, an encoder-decoder transformer, and a simple feed forward network (FFN) that makes the final detection prediction. Each encoder layer has a standard architecture and consists of a multi-head self-attention module and a feed forward network (FFN).

The CNN backbone enables 2D representation of input image. Positional embedding is added before passing to the encoder. The decoder then decodes the N objects in parallel at each decoder layer. A FFN then performs the classification.

This method has shown similar evaluation scores as recent Faster R-CNN and even appears to perform better on large objects due to the attention mechanisms.

Conclusion

In conclusion, Faster R-CNN, Mask R-CNN, YOLO, RetinaNet, and DETR represent significant advancements in the field of computer vision and object detection. Each of these models offers unique strengths and capabilities. Known for its accuracy and robustness, Faster R-CNN excels in object detection tasks. It introduced the concept of Region Proposal Networks (RPNs), improving both speed and precision. Building upon Faster R-CNN, Mask R-CNN extends its capabilities to instance segmentation, enabling precise object boundary delineation. It's a go-to choice for tasks that require pixel-level accuracy. YOLO is a real-time object detection system known for its speed. It's well-suited for applications where low latency is critical but may sacrifice some precision compared to other models. RetinaNet combines the speed of one-stage detectors like YOLO with the accuracy of two-stage detectors like Faster R-CNN. It's an excellent choice for a wide range of object detection tasks. DETR leverages transformers and is unique for its ability to handle object detection as a set prediction task, eliminating the need for anchor boxes. It shows promise in improving detection performance while requiring less labelled data.
Ultimately, the choice among these models depends on the specific requirements of the task, including speed, accuracy, and data availability. Researchers and practitioners should consider the trade-offs and strengths of each model to select the most suitable one for their needs.

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans Pattern Anal Mach Intell*, vol. 39, no. 6, 2017, doi: 10.1109/TPAMI.2016.2577031.

[2] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017. doi: 10.1109/ICCV.2017.322.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once : Unified, real-time object detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016. doi: 10.1109/CVPR.2016.91.

[4] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," *IEEE Trans Pattern Anal Mach Intell*, vol. 42, no. 2, 2020, doi: 10.1109/TPAMI.2018.2858826.

[5] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2020. doi: 10.1007/978-3-030-58452-8_13.