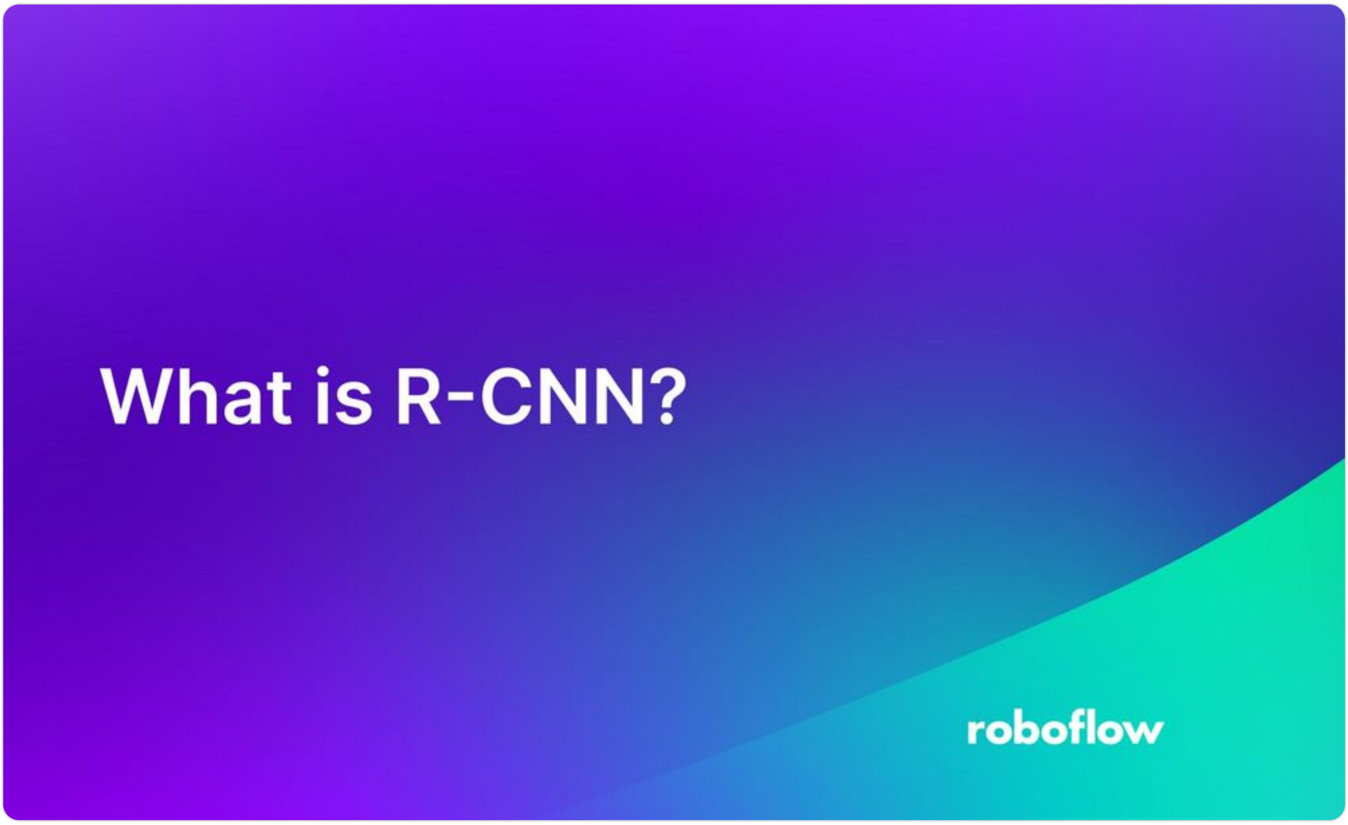
COMPUTER VISION

What is R-CNN?

Petru Potrimba
SEP 25, 2023 6 MIN READ



The year 2014 marked a pivotal moment in the field of computer vision with the publication of the R-CNN paper. This paper not only demonstrated the potential of Convolutional Neural Networks (CNNs) for achieving high performance in object detection but also addressed a challenging problem: the precise localization of objects within images through the creation of bounding boxes.

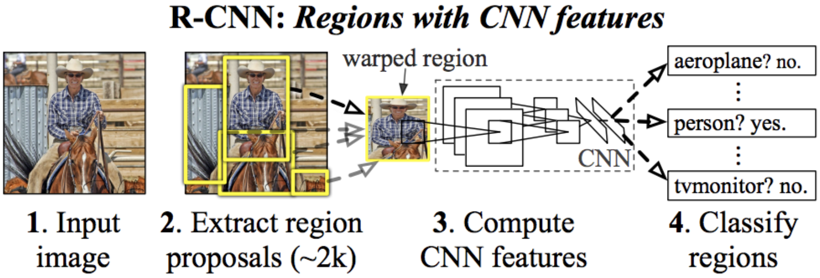
While CNNs had already gained prominence for image classification tasks, object detection required a more intricate solution. This is where the R-CNN algorithm entered the scene, offering a novel approach to this complex problem.

R-CNN paved the way for subsequent innovations in object detection, including Fast R-CNN, Faster R-CNN, and Mask R-CNN, each building upon and enhancing the capabilities of its predecessor. To grasp the nuances of these advanced R-CNN variants, it is essential to establish a solid foundation in the original R-CNN architecture.

In this blog post, we will delve deep into the workings of R-CNN, providing a comprehensive understanding of its inner workings.

What is R-CNN? How Does R-CNN Work?

Region-based Convolutional Neural Network (R-CNN) is a type of deep learning architecture used for object detection in computer vision tasks. RCNN was one of the pioneering models that helped advance the object detection field by combining the power of convolutional neural networks and region-based approaches.



The R-CNN Architecture, featuring the steps of taking in an input image, extracting region proposals, computing CNN features, and classifying regions. Source

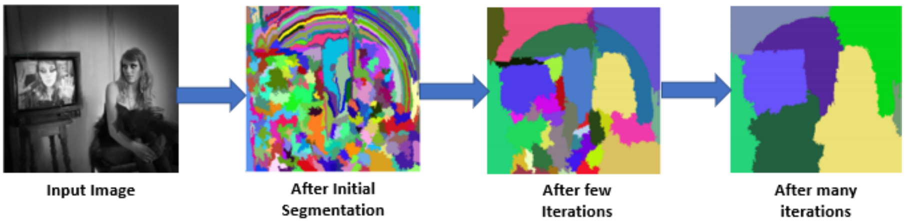
Let's dive deeper into how R-CNN works, step by step.

Region Proposal

R-CNN starts by dividing the input image into multiple regions or subregions. These regions are referred to as "region proposals" or "region candidates." The region proposal step is responsible for generating a set of potential regions in the image that are likely to contain objects. R-CNN does not generate these proposals itself; instead, it relies on external methods like Selective Search or EdgeBoxes to generate region proposals.

Selective Search, for example, operates by merging or splitting segments of the image based on various image cues like color, texture, and shape to create a diverse set of region proposals.

Below we show how Selective Search works, which shows an input image, then an image with many segmented masks, then fewer masks, then masks that comprise the main components in the image.



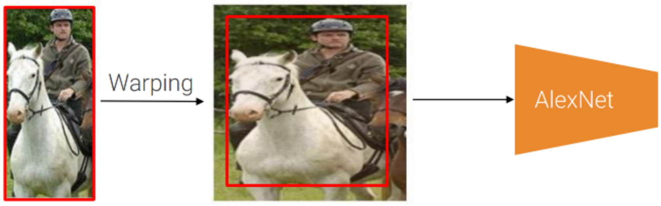
Example of the Selective Search algorithm applied to an image. Source.

Feature Extraction

Once the region proposals are generated, approximately 2,000 regions are extracted and anisotropically warped to a consistent input size that the CNN expects (e.g., 224x224 pixels) and then it is passed through the CNN to extract features.

Before warping, the region size is expanded to a new size that will result in 16 pixels of context in the warped frame. The CNN used is AlexNet and it is typically fine-tuned on a large dataset like ImageNet for generic feature representation.

The output of the CNN is a high-dimensional feature vector representing the content of the region proposal.



An example of how warping works in R-CNN. Note the addition of 16 pixels of context in the middle image.

Object Classification

The extracted feature vectors from the region proposals are fed into a separate machine learning classifier for each object class of interest. R-CNN typically uses Support Vector Machines (SVMs) for classification. For each class, a unique SVM is trained to determine whether or not the region proposal contains an instance of that class.

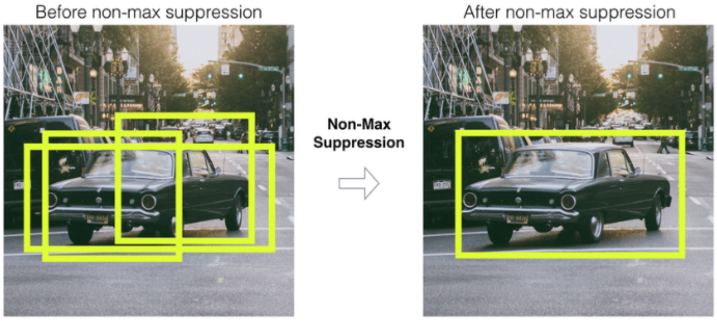
During training, positive samples are regions that contain an instance of the class. Negative samples are regions that do not.

Bounding Box Regression

In addition to classifying objects, R-CNN also performs bounding box regression. For each class, a separate regression model is trained to refine the location and size of the bounding box around the detected object. The bounding box regression helps improve the accuracy of object localization by adjusting the initially proposed bounding box to better fit the object's actual boundaries.

Non-Maximum Suppression (NMS)

After classifying and regressing bounding boxes for each region proposal, R-CNN applies non-maximum suppression to eliminate duplicate or highly overlapping bounding boxes. NMS ensures that only the most confident and non-overlapping bounding boxes are retained as final object detections.



Example of Non-Maximum Suppression. Source.

R-CNN Strengths and Disadvantages

Now that we've covered what R-CNN is and how it works, let's delve into the strengths and weaknesses of this popular object detection framework. Understanding the strengths and disadvantages of R-CNN can help you make an informed decision when choosing an approach for your specific computer vision tasks.

Strengths of R-CNN

Below are a few of the key strengths of the R-CNN architecture.

- **Accurate Object Detection**: R-CNN provides accurate object detection by leveraging region-based convolutional features. It excels in scenarios where precise object localization and recognition are crucial.
- **Robustness to Object Variations**: R-CNN models can handle objects with different sizes, orientations, and scales, making them suitable for real-world scenarios with diverse objects and complex backgrounds.
- **Flexibility**: R-CNN is a versatile framework that can be adapted to various object detection tasks, including instance segmentation and object tracking. By modifying the final layers of the network, you can tailor R-CNN to suit your specific needs.

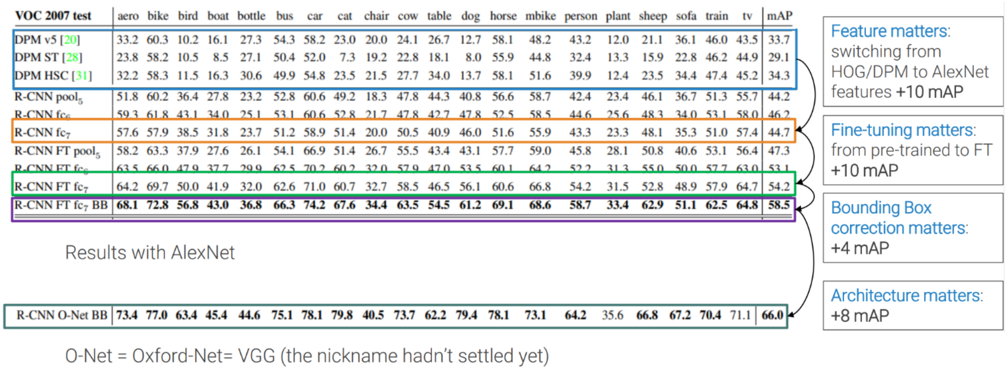
Disadvantages of R-CNN

Below are a few disadvantages of the R-CNN architecture.

- **Computational Complexity**: R-CNN is computationally intensive. It involves extracting region proposals, applying a CNN to each proposal, and then running the extracted features through a classifier. This multi-stage process can be slow and resource-demanding.
- **Slow Inference**: Due to its sequential processing of region proposals, R-CNN is relatively slow during inference. Real-time applications may find this latency unacceptable.
- **Overlapping Region Proposals**: R-CNN may generate multiple region proposals that overlap significantly, leading to redundant computation and potentially affecting detection performance.
- **R-CNN is Not End-to-End**: Unlike more modern object detection architectures like Faster R-CNN, R-CNN is not an end-to-end model. It involves separate modules for region proposal and classification, which can lead to suboptimal performance compared to models that optimize both tasks jointly.

R-CNN Performance

In the following segment, we will delve into the performance of R-CNN, accompanied by visual demonstrations showcasing its object detection capabilities using the Pascal VoC 2007 dataset.



VOC 2007 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM v5 [20]	33.2	60.3	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7	12.7	58.1	48.2	43.2	12.0	21.1	36.1	46.0	43.5	33.7
DPM ST [28]	23.8	58.2	10.5	8.5	27.1	50.4	52.0	7.3	19.2	22.8	18.1	8.0	55.9	44.8	32.4	13.3	15.9	22.8	46.2	44.9	29.1
DPM HSC [31]	32.2	58.3	11.5	16.3	30.6	49.9	54.8	23.5	21.5	27.7	34.0	13.7	58.1	51.6	39.9	12.4	23.5	34.4	47.4	45.2	34.3
R-CNN pool$_5$	51.8	60.2	36.4	27.8	23.2	52.8	60.6	49.2	18.3	47.8	44.3	40.8	56.6	58.7	42.4	23.4	46.1	36.7	51.3	55.7	44.2
R-CNN fc$_6$	59.3	61.8	43.1	34.0	25.1	53.1	60.6	52.8	21.7	47.8	42.7	47.8	52.5	58.5	44.6	25.6	48.3	34.0	53.1	58.0	46.2
R-CNN fc$_7$	57.6	57.9	38.5	31.8	23.7	51.2	58.9	51.4	20.0	50.5	40.9	46.0	51.6	55.9	43.3	23.3	48.1	35.3	51.0	57.4	44.7
R-CNN FT pool$_5$	58.2	63.3	37.9	27.6	26.1	54.1	66.9	51.4	26.7	55.5	43.4	43.1	57.7	59.0	45.8	28.1	50.8	40.6	53.1	56.4	47.3
R-CNN FT fc$_6$	63.5	66.0	47.9	37.7	29.9	62.5	70.2	60.2	32.0	57.9	47.0	53.5	60.1	64.2	52.2	31.3	55.0	50.0	57.7	63.0	53.1
R-CNN FT fc$_7$	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7	54.2
R-CNN FT fc$_7$ BB	**68.1**	**72.8**	**56.8**	**43.0**	**36.8**	**66.3**	**74.2**	**67.6**	**34.4**	**63.5**	**54.5**	**61.2**	**69.1**	**68.6**	**58.7**	**33.4**	**62.9**	**51.1**	**62.5**	**64.8**	**58.5**

Results with AlexNet

R-CNN O-Net BB	**73.4**	**77.0**	**63.4**	**45.4**	**44.6**	**75.1**	**78.1**	**79.8**	**40.5**	**73.7**	**62.2**	**79.4**	**78.1**	**73.1**	**64.2**	35.6	**66.8**	**67.2**	**70.4**	71.1	**66.0**

O-Net = Oxford-Net= VGG (the nickname hadn't settled yet)

Feature matters: switching from HOG/DPM to AlexNet features +10 mAP

Fine-tuning matters: from pre-trained to FT +10 mAP

Bounding Box correction matters: +4 mAP

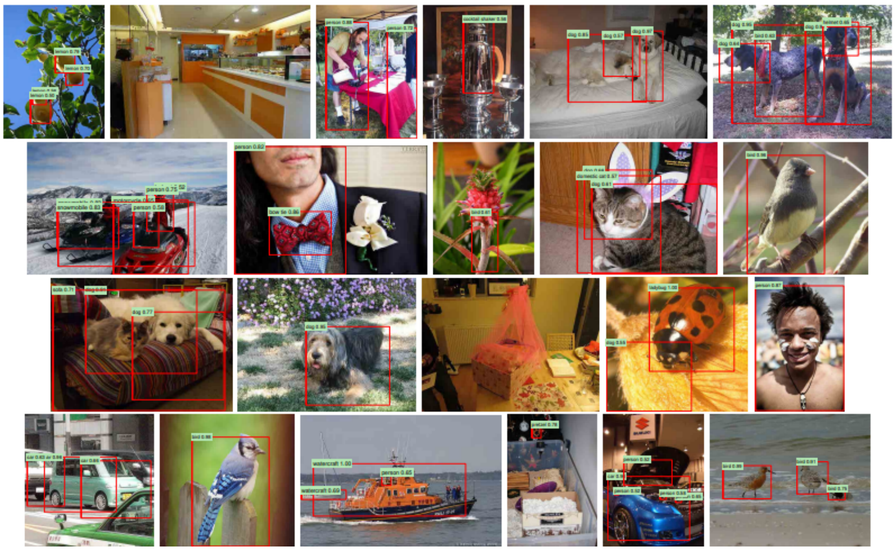
Architecture matters: +8 mAP

R-CNN performance on Pascal VoC 2007 dataset. Source.

The image above provides insights into several key factors:

1. **Significance of features**: Transitioning from a HOG/DPM algorithm to a Convolutional Neural Network like AlexNext results in a substantial increase in mAP, approximately 10 points higher.
2. **Impact of Fine-Tuning**: Opting for fine-tuning, rather than starting from solely pre-trained weights, also yields a notable improvement, raising the mAP by around 10 points.
3. **Influence of Bounding Box Correction**: The introduction of bounding box correction further enhances the mAP by approximately 4 points.
4. **Role of architecture**: Shifting from AlexNet to VGG (referred to as O-Net in the table, reflecting the evolving nomenclature) brings yet another improvement, boosting the mAP by an additional 8 points.

Next, we present visual illustrations of R-CNN's object detection performance on the Pascal VoC 2007 dataset.

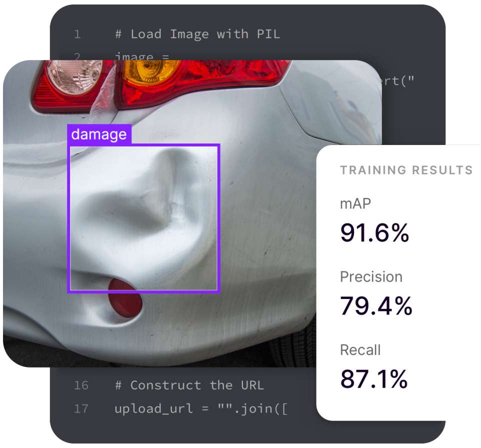


Some detections by R-CNN displayed on images from the Pascal VoC 2007 dataset. Source.

Conclusion

In this blog post, we have embarked on a journey through the intricacies of this object detection framework. R-CNN's architecture has demonstrated its prowess in delivering good accuracy in object detection tasks. In particular, R-CNN examined the pivotal factors that influence the mAP score, shedding light on the importance of features, fine-tuning, bounding box correction, and architectural choices.

However, R-CNN is not without its challenges. Its computational complexity and slower inference times can be daunting, particularly in real-time scenarios. As we conclude this journey through R-CNN, it's clear that this framework has left an indelible mark on the field of computer vision. Its contributions to object detection and related tasks have paved the way for subsequent advancements, each building upon its foundations.



Petru Potrimba

Petru Potrimba is a Machine Learning Engineer, with a passion for computer vision because a picture is worth a thousand words.

VIEW MORE POSTS

TC

Co

MORE ABOUT COMPUTER VISION

VIEW ALL

Monitoring Plant Growth using Computer Vision

DEC 12, 2023

Launch: Advanced Ontology Management with Roboflow

DEC 11, 2023

Google's Gemini Multimodal Model: What We Know

DEC 7, 2023

Comparing Custom Models to Google Cloud Vision API

DEC 6, 2023

How to Detect Objects with YOLOv8

DEC 6, 2023

Use Cases for Computer Vision in Analytics

DEC 4, 2023

Want to learn more about Roboflow? Email sales@roboflow.com or book a demo with our sales team.

PRODUCT	DEVELOPERS	COMPANY
Sign In / Sign Up	User Forum	About Us
Universe	Templates	Careers
Annotate	Blog	Press

For sales inquiries:

sales@roboflow.com

Book a demo

aws PARTNER ML Software Competency

aws Qualified Software

AICPA SOC

Train

Deploy

Integrations

Pricing

ECOSYSTEM

Notebooks

Autodistill

Supervision

Roboflow 100

Contact Sales

Learn Computer Vision

Convert Annotation Formats

Computer Vision Models

Computer Vision Utilities

INDUSTRIES

Manufacturing

Oil & Gas

Retail

Safety & Security

Transportation

All Industries

Media Kit

Terms of Service

Privacy Policy

Sitemap