Deep Convolutional Network for Object Detection

A Report to final project submitted to the professor of
San Francisco State University
In partial fulfillment of
the requirement of
the Degree

by
Poornank Purohit
San Francisco, California
Fall Semester 2022

Abstract

In the last few years, the deep learning computing paradigm has been deemed a high standard in machine leaning. Moreover, it has gradually become the most widely used computational approach in the field of ML, thus achieving outstanding results on several complex cognitive tasks. Object detection, as one of the three main tasks of computer vision, is significant for the development of artificial intelligence in the future. The rapid advancement of convolutional neural networks and deep learning have provided a broader area for object detection. Compared with traditional handcrafted feature-based method, the deep learning-based object detection methods can learn both low-level and high-level image features. The image features learned through deep learning techniques are more representative than the handcrafted features. Therefore, this review paper focuses on the object detection algorithms based on deep convolutional neural networks, while the traditional object detection algorithms will be simply introduced as well. Through the review and analysis of deep learning-based object detection techniques in recent years, this work includes fundamental visual recognition problems in the field of computer vision and object detection and compare these methods with deep learning-based methods.

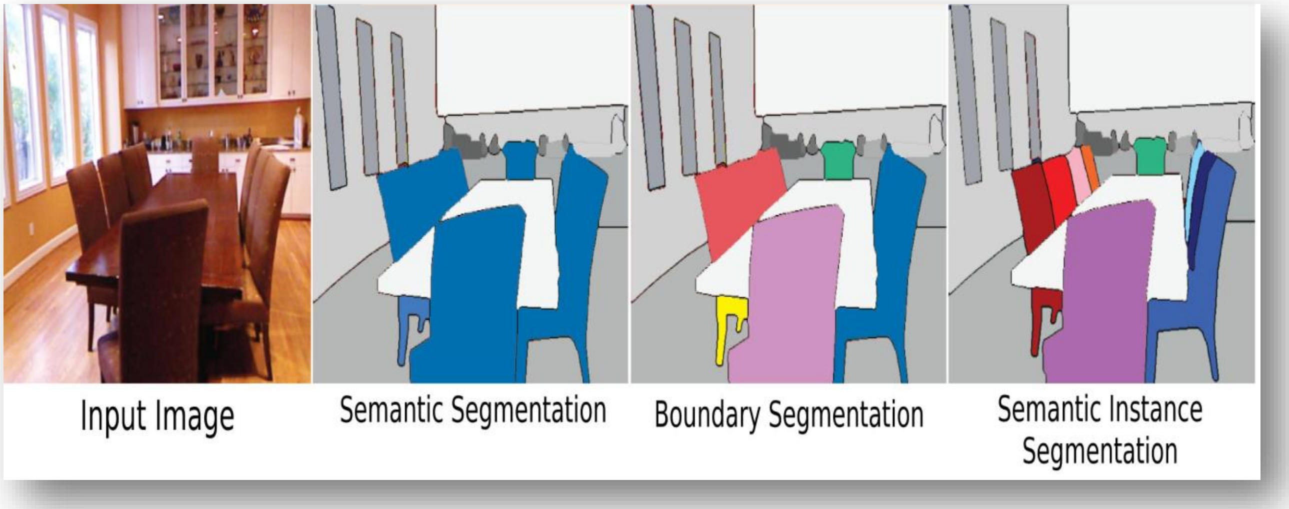
1. Introduction

Computer Vision is an interdisciplinary field that has been gaining huge amounts of traction in recent years and self-driving cars have taken center stage. Another integral part of computer vision is object detection. Object detection aids in pose estimation, vehicle detection, surveillance etc. The difference between object detection algorithms and classification algorithms is that in detection algorithms, we try to draw a bounding box around the object of interest to locate it within the image. Also, you might not necessarily draw just one bounding box in an object detection case, there could be many bounding boxes representing different objects of interest within the image and you would not know how many beforehand.

This report examines the work including fundamental visual recognition problems in the field of computer vision and object detection and compare these methods with deep learning-based methods along with the key differences between these methods of R-CNN, Fast R-CNN, Faster R-CNN and so forth with three important object detection algorithms. All these algorithms are well-known and widely used in applications such as autonomous driving and facial recognition. This report will discuss the differences between these algorithms and how each one has improved on the previous iteration.

Object detection not only recognizes the object categories, but also predicts the location of each object instance via bounding boxes whose implementation can be seen in applications of autonomous driving. CNNs saw heavy use in the 1990s, but then fell out of fashion with the rise *of support vector machines. In 2012, Krizhevsky et al. [23] rekindled interest in CNNs by showing substantially higher image classification accuracy on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Their success resulted from training a large CNN on 1.2 million labeled images, together with a few twists on LeCun's CNN (e.g., max (x, 0) rectifying non-linearities and "dropout" regularization). The significance of the ImageNet result was vigorously debated during the ILSVRC 2012 workshop. The central issue can be distilled to the following:*

To what extent does the CNN classification result on ImageNet generalize to object detection results on the PASCAL VOC Challenge? We answer this question by bridging the gap between image classification and object detection.



So, object detection has some relations with object classification, semantic segmentation and instance segmentation. The limitation of computing resources along with datasets and basic with theories have limited development over and application of deep neural networks. Thus, the field of computer vision with traditional object detection algorithms are still popular which include DPM, Selective Search, MKLs and so forth. The basic architecture of traditional object *detection algorithms focuses on dividing the region into selected features and classifiers which determine the region of interest as shown in the above figure.*

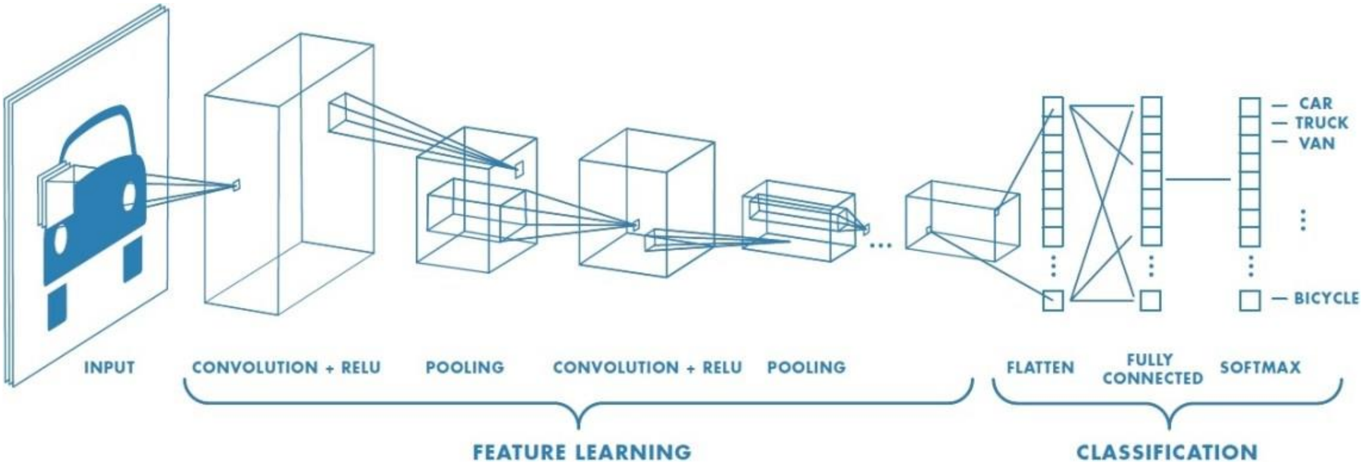
2. CNN

CNN (Convolutional Neural Networks) is a deep learning technique used to classify images. It has become popular in the field of computer vision, due to its ability to extract features from an image and classify them accurately. R-CNN (Region-based CNN) is an extension of CNN, which uses a region proposal network to identify regions of interest in an image and then uses CNN to classify those regions. A Convolutional Neural Network (CNN) is a type of artificial neural network used in image recognition and processing that is optimized to process pixel data. Therefore, Convolutional Neural Networks are the fundamental and basic building blocks for the computer vision task of image segmentation (CNN segmentation).

The Convolutional Neural Network Architecture consists of three main layers:

- *Convolutional layer: This layer helps to abstract the input image as a feature map via the use of filters and kernels.*
- *Pooling layer: This layer helps to down sample feature maps by summarizing the presence of features in patches of the feature map.*
- *Fully connected layer: Fully connected layers connect every neuron in one layer to every neuron in another layer.*

Combining the layers of a CNN enables the designed neural network to learn how to identify and recognize the object of interest in an image. Simple Convolutional Neural Networks are built for image classification and object detection with a single object in the image.



CNNs are regularized versions of multilayer perceptrons. Multilayer perceptron's usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "full connectivity" of these networks makes them prone to overfitting data. Typical ways of regularization, or preventing overfitting, include penalizing parameters during training (such as weight decay) or trimming connectivity (skipped connections, dropout, etc.) CNNs take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble patterns of increasing complexity using smaller and simpler patterns embossed in their filters. Therefore, on a scale of connectivity and complexity, CNNs are on the lower extreme. After passing through a convolutional layer, the image becomes abstracted to a feature map, also called an activation map, with shape: (number of inputs) × (feature map height) × (feature map width) × (feature map channels). Convolutional layers convolve the input and pass its result to the next layer. This is like the response of a neuron in *the visual cortex to a specific stimulus. Each convolutional neuron processes data only for its receptive field. Although fully connected feedforward neural networks can be used to learn features and classify data, this architecture is generally impractical for larger inputs such as high-resolution images. It would require a very high number of neurons, even in a shallow architecture, due to the large input size of images, where each pixel is a relevant input feature. For instance, a fully connected layer for a (small) image of size 100 × 100 has 10,000 weights for each neuron in the second layer. Instead, convolution reduces the number of free parameters, allowing the network to be deeper. For example, regardless of image size, using a 5 × 5 tiling region, each with the same shared weights, requires only 25 learnable parameters. Using regularized weights over fewer parameters avoids the vanishing gradients and exploding gradients problems seen during backpropagation in traditional neural networks. Furthermore, convolutional neural networks are ideal for data with a grid-like topology (such as images) as spatial relations between separate features are considered during convolution and/or pooling.*

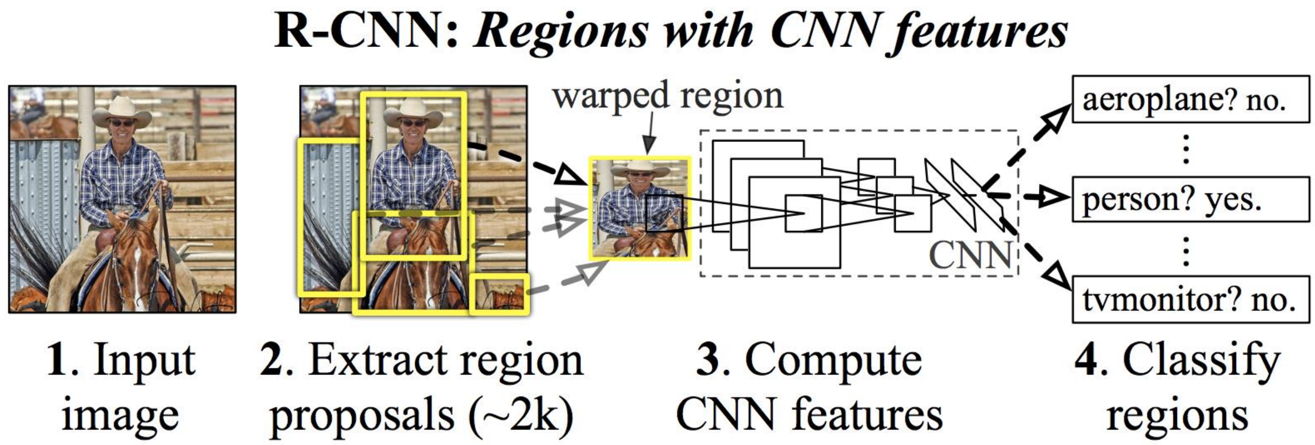
Convolutional networks may include local and/or global pooling layers along with traditional convolutional layers. Pooling layers reduce the dimensions of data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Local pooling combines small clusters, tiling sizes such as 2 × 2 are commonly used. Global pooling acts on all the neurons of the feature map. There are two common types of pooling in popular use: max and average. Max pooling uses the maximum value of each local cluster of neurons in the feature map, while average pooling takes the average value.

Fully connected layers connect every neuron in one layer to every neuron in another layer. It is the same as a traditional multilayer perceptron neural network (MLP). The flattened matrix goes *through a fully connected layer to classify the images. The vectors of weights and biases are called filters and represent features of the input (e.g., a particular shape). A distinguishing feature of CNNs is that many neurons can share the same filter. This reduces the memory footprint because a single bias and a single vector of weights are used across all receptive fields that share that filter, as opposed to each receptive field having its own bias and vector weighting.*

3. R-CNN

To bypass the problem of selecting a huge number of regions, Ross Girshick et al. proposed a method where we use selective search to extract just 2000 regions from the image and he called them region proposals. Therefore, now, instead of trying to classify a huge number of regions, you can just work with 2000 regions. These 2000 region proposals are generated using the selective search algorithm which is written below. R-CNN, or Region-based Convolutional Neural Network, is a two-stage algorithm used to detect and classify objects in an image. This algorithm first uses a Selective Search algorithm to generate region proposals. These region proposals are then passed through a Convolutional Neural Network (CNN) to classify the objects.

This algorithm was the first to use a CNN for object detection, and it achieved impressive results in its time. However, it was computationally expensive, as CNN had to be run for every region proposal.



R-CNN (Region-based CNN) is an extension of CNN, which uses a region proposal network to identify regions of interest in an image and then uses CNN to classify those regions. This allows R-CNN to be more accurate in its classification of images, but it is slower than CNN.

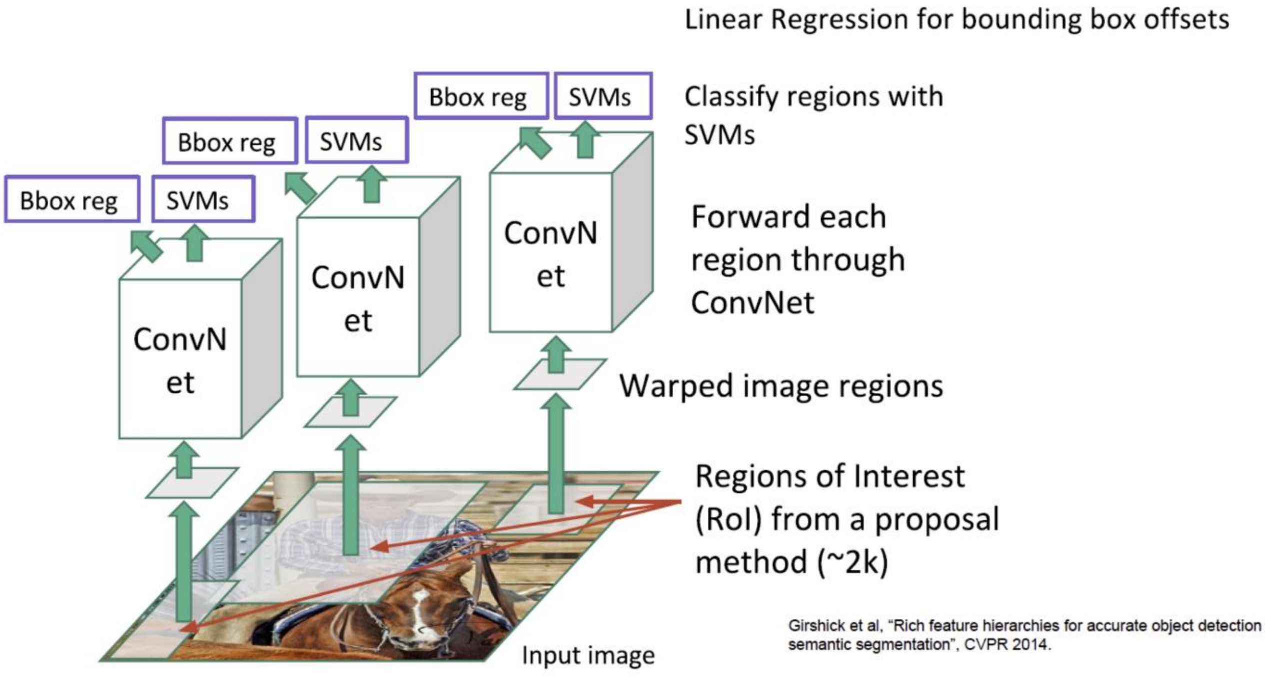
These 2000 candidate region proposals are warped into a square and fed into a convolutional neural network that produces a 4096-dimensional feature vector as output. The CNN acts as a feature extractor and the output dense layer consists of the features extracted from the image and the extracted features are fed into an SVM to classify the presence of the object within that candidate region proposal.

In addition to predicting the presence of an object within the region proposals, the algorithm also predicts four values which are offset values to increase the precision of the bounding box. For example, given a region proposal, the algorithm would have predicted the presence of a person but the face of that person within that region proposal could've been cut in half. Therefore, the offset values help in adjusting the bounding box of the region proposal.

Disadvantages of R-CNN:

Although R-CNN can be useful for object detection, it still has some limitations and drawbacks when it comes to computational speed. It takes fixed size input to convolutional network and if the image is not corresponding to the size, it crops the image segments it by takes it in as the input which might result in less accuracy.

Also, since there can be more than one bounding box the CNN network must parse it through and entire process as many times as the bounding boxes are detected which is computationally expensive and takes around 40-50 seconds to make predictions for each new image, which makes the model cumbersome and impossible to build when faced with gigantic datasets.

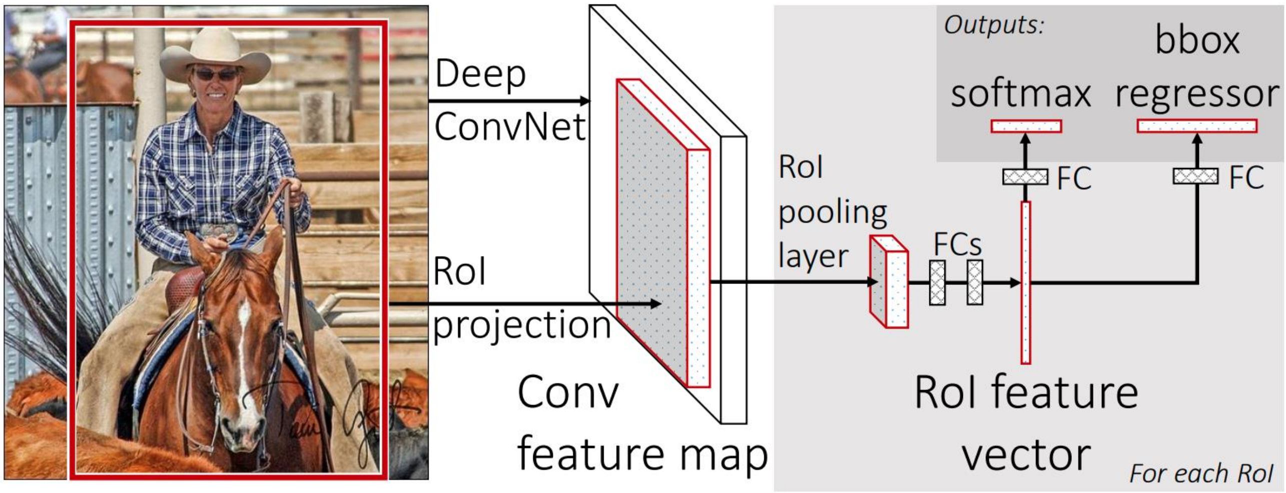

Linear Regression for bounding box offsets
Bbox reg
SVMs
Bbox reg
SVMs
Bbox reg
SVMs
Classify regions with SVMs
ConvN et
ConvN et
ConvN et
Forward each region through ConvNet
Warped image regions
Regions of Interest (RoI) from a proposal method (~2k)
Input image

Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

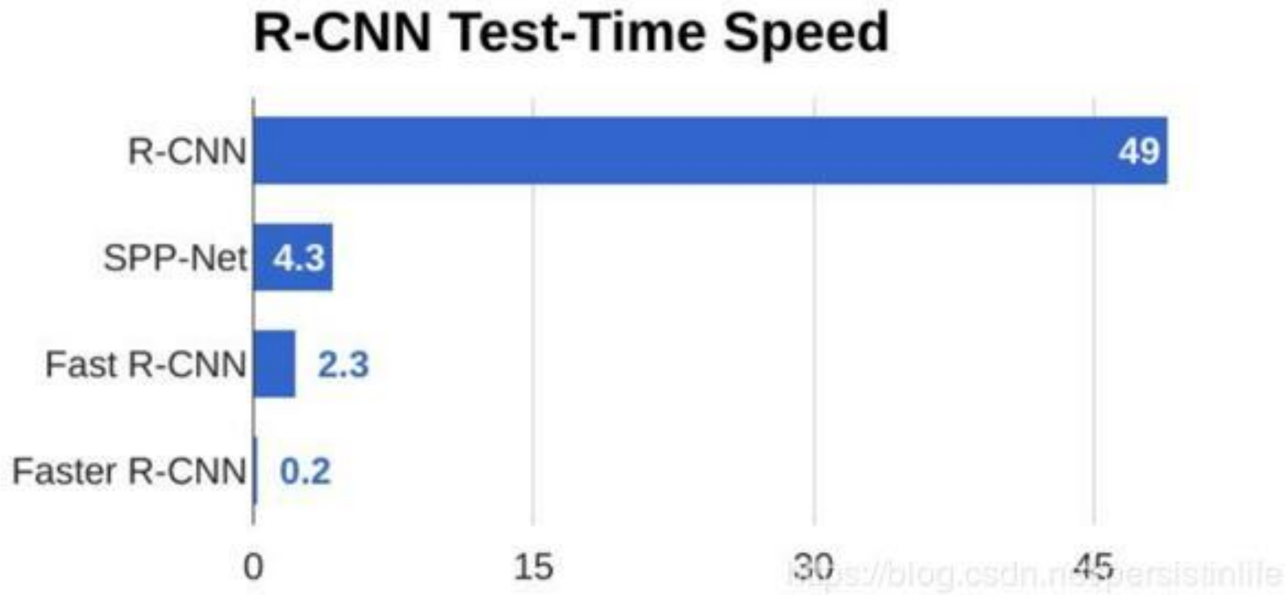
It still takes a huge amount of time to train the network as you would have to classify 2000 region proposals per image. It cannot be implemented in real time as it takes around 47 seconds for each test image. The selective search algorithm is a fixed algorithm. Therefore, no learning is happening at that stage. This could lead to a generation of bad candidate region proposals. Also, the R-CNN model consists of three models separately with much more shared computations and computing all these three models takes lots of disk space along with being computationally expensive.

4. Fast R-CNN

In 2015, the introduction of Fast R-CNN improved the network structure of R-CNN and combined with SPPnet to propose a region of interest (RoI) pooling layer. This approach avoids the problem of repeated training of multiple suggestion boxes in R-CNN, and after RoI pooling, the problem of the inconsistent size of the fully connected layer that is obtained by Singular Value Decomposition (SVD) can be solved. The backbone network uses VGG-16 which has a deeper layer network. Testing on the VOC2007 dataset obtained a map of 70%, while R-CNN and SPPnet were 66% and 63.1% respectively. The training time was also reduced from 84 hours (about 3 and a half days) on R-CNN to 9.5 hours, which is about 9 times faster than R-CNN. The whole detection process of Fast R-CNN still suffers some weaknesses. Using selective search methods to generate many suggested regions proposal, resulting in a long training and prediction time, which has not yet achieved the purpose of real-time detection. Multiple fully connected layers are used at the end of the network and are calculated separately, the weights *are not shared, which increases the number of parameters. The approach is like the R-CNN algorithm. But, instead of feeding the region proposals to CNN, we feed the input image to CNN to generate a convolutional feature map.*



From the convolutional feature map, we identify the region of proposals and warp them into squares and by using a RoI pooling layer we reshape them into a fixed size so that it can be fed into a fully connected layer. From the RoI feature vector, we use a SoftMax layer to predict the class of the proposed region and offset values for the bounding box. The reason "Fast R-CNN" is faster than R-CNN is because you do not have to feed 2000 region proposals to the convolutional neural network every time. Instead, the convolution operation is done only once per image and a feature map is generated from it. It is an improvement over R-CNN, which uses a single pass for both feature extraction and region proposal network. This makes Fast R-CNN faster than R-CNN, while still maintaining the accuracy of region-based classification.



From the above graphs, you can infer that Fast R-CNN is significantly faster in training and testing sessions over R-CNN. When you look at the performance of Fast R-CNN during testing time, including region proposals slows down the algorithm significantly when compared to not using region proposals. Therefore, region proposals become bottlenecks in Fast R-CNN algorithm affecting its performance.

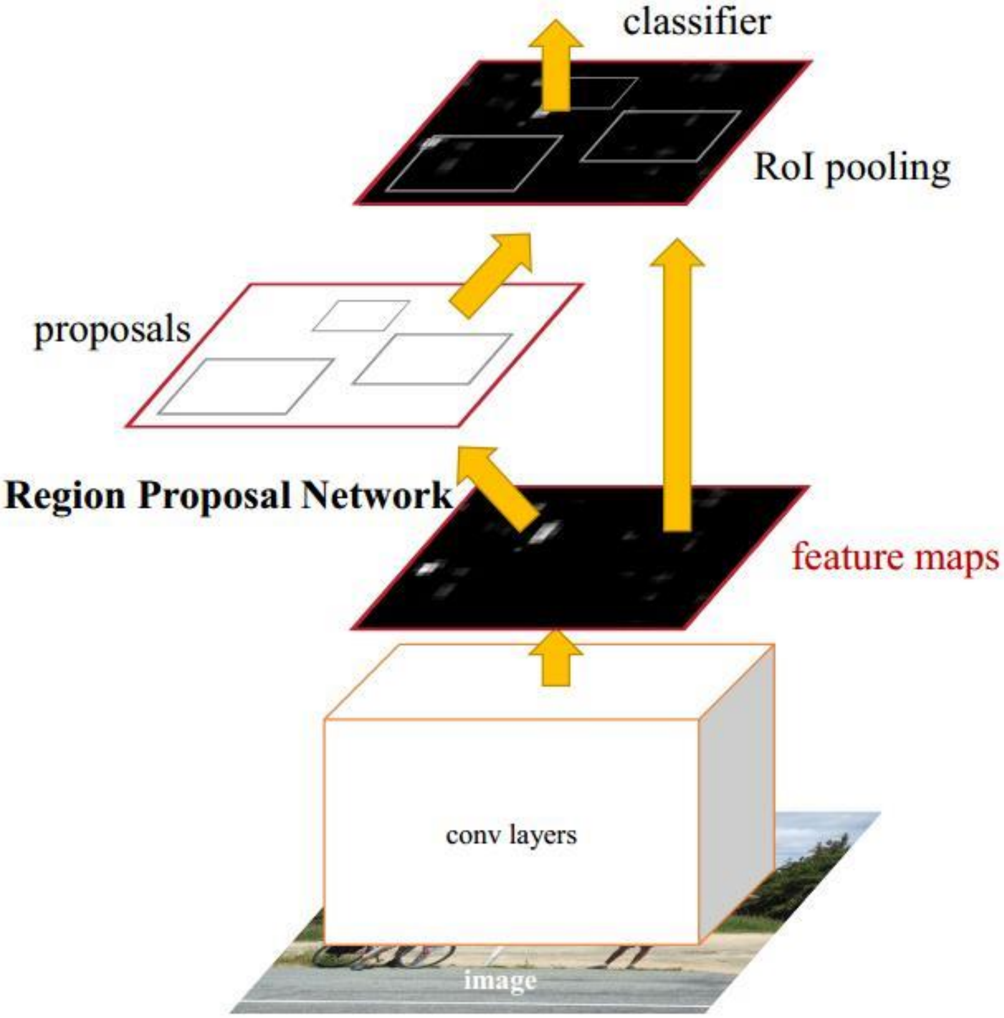
Disadvantages of Fast R-CNN:

Although Fast R-CNN is faster than R-CNN in both training and testing time, it still was slow and had certain issues when it came to real life datasets.

That is because it uses selective search to get Regions of Interest same as R-CNN which slows down the process and is time consuming process. Most of the time taken by Fast R-CNN during *detection is a selective search region proposal generation algorithm. Hence, it is the bottleneck of this architecture which was dealt with in Faster R-CNN.*

5. Faster R-CNN

Faster R-CNN was introduced in 2015 by k He et al. After the Fast R-CNN, the bottleneck of the architecture is selective search. Since it needs to generate 2000 proposals per image. It constitutes a major part of the training time of the whole architecture. In Faster R-CNN, it was replaced by the region proposal network. First, in this network, we passed the image into the backbone network. This backbone network generates a convolution feature map. These feature maps are then passed into the region proposal network. The region proposal network takes a feature map and generates the anchors (the center of the sliding window with a unique size and scale). These anchors are then passed into the classification layer (which classifies whether there is an object or not) and the regression layer (which localizes the bounding box associated with an object).



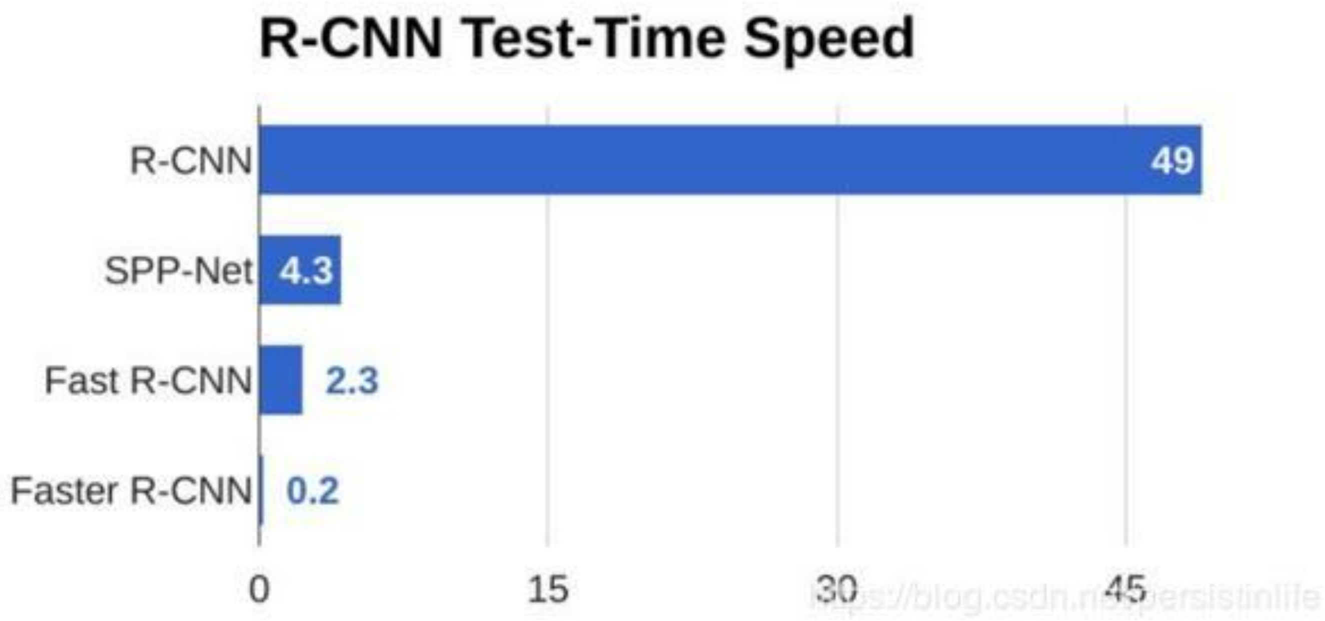
So, Faster R-CNN (Faster Region-based CNN) is an even more advanced version of Fast R-CNN, which uses a region proposal network and a convolutional neural network in a single pass. Comparison CNN is a deep learning technique used to classify images.

It uses convolutional layers, pooling layers, and fully connected layers to extract features and classify them. It is fast and accurate, but it is limited in its ability to identify regions of interest within an image. In terms of Detection time, Faster R-CNN is faster than both R-CNN and Fast R-

CNN. The Faster R-CNN also has better mAP than both the previous ones. Both above algorithms (R-CNN & Fast R-CNN) use selective search to find out the region proposals. Selective search is a slow and time-consuming process affecting the performance of the network. Therefore, Shaoqing Ren et al. came up with an object detection algorithm that eliminates the selective search algorithm and lets the network learn the region proposals.

Like Fast R-CNN, the image is provided as an input to a convolutional network which provides a convolutional feature map. Instead of using selective search algorithm on the feature map to identify the region proposals, a separate network is used to predict the region proposals.

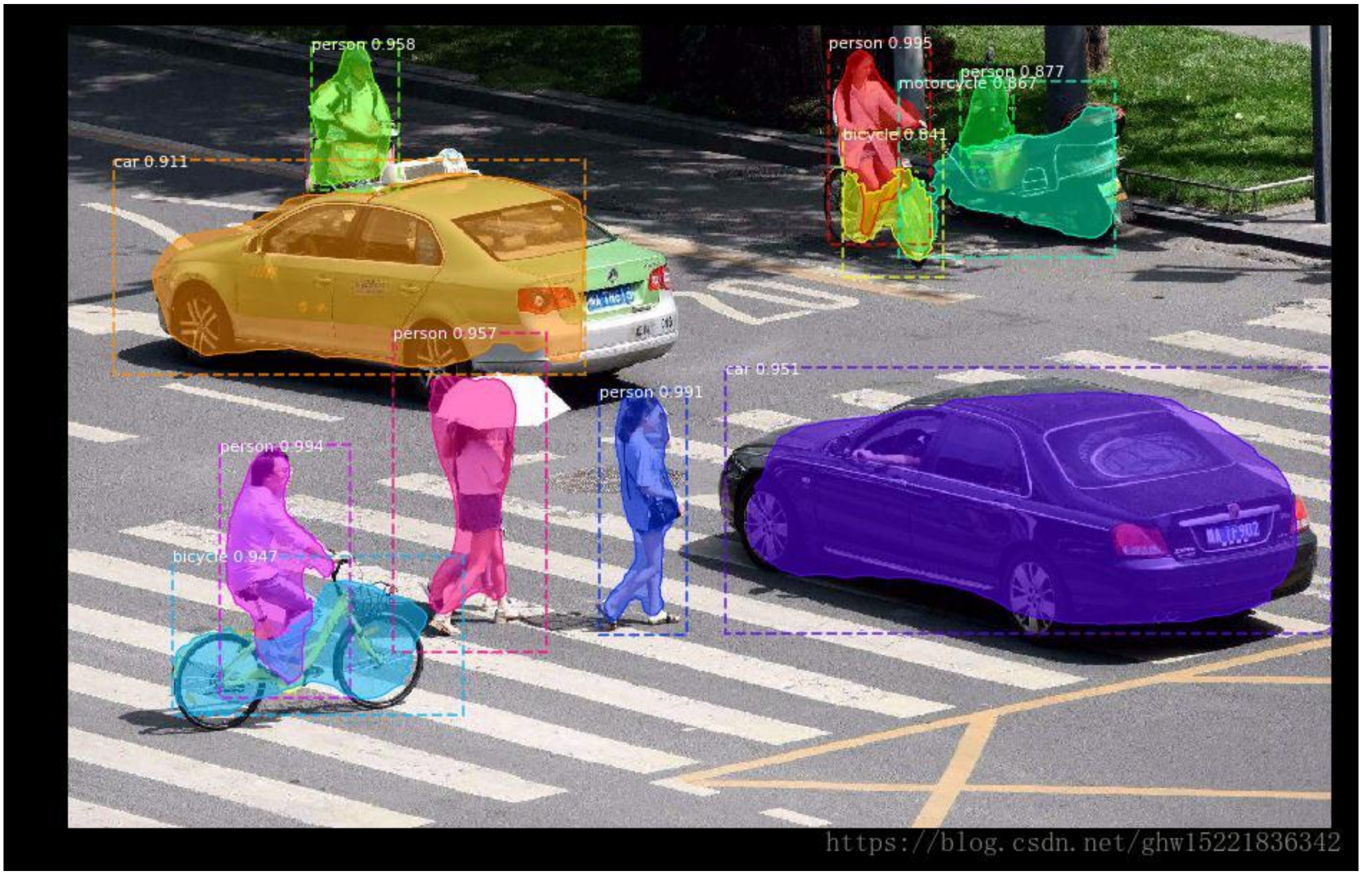
The predicted region proposals are then reshaped using a RoI pooling layer which is then used to classify the image within the proposed region and predict the offset values for the bounding boxes.



From the above graph, you can see that Faster R-CNN is much faster than its predecessors. Therefore, it can even be used for real-time object detection.

6. Mask R-CNN

So far, we've seen how we've been able to use CNN features in many interesting ways to effectively locate different objects in an image with bounding boxes.



Can we extend such techniques to go one step further and locate exact pixels of each object instead of just bounding boxes? This problem, known as image segmentation, is what Kaiming He and a team of researchers, including Girshick, explored at Facebook AI using an architecture *known as Mask R-CNN. Mask R-CNN does this by adding a branch to Faster R-CNN that outputs a binary mask saying whether a given pixel is part of an object. The branch (in white in the above image), as before, is just a Fully Convolutional Network on top of a CNN based feature map. Here are its inputs and outputs:*

- *Inputs: CNN Feature Map.*
- *Outputs: Matrix with 1s on all locations where the pixel belongs to the object and 0s elsewhere (this is known as a binary mask).*

But the Mask R-CNN authors had to make one small adjustment to make this pipeline work as expected.

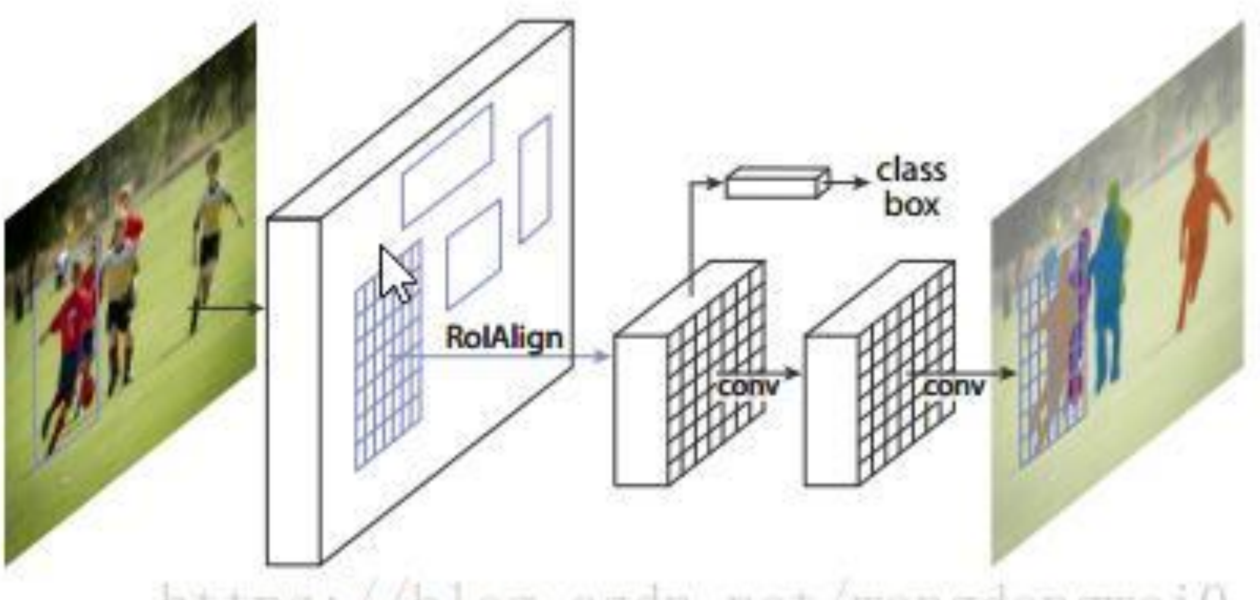


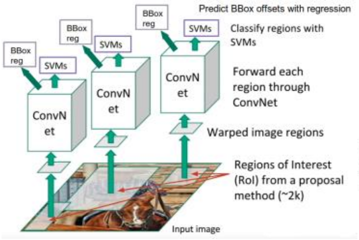
Figure 1. The **Mask R-CNN** framework for instance segmentation.

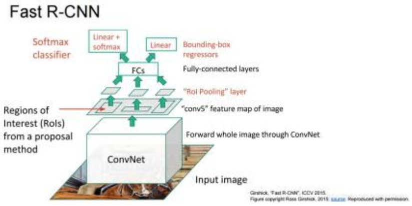
When run without modifications on the original Faster R-CNN architecture, the Mask R-CNN authors realized that the regions of the feature map selected by RoIPool were slightly *misaligned from the regions of the original image. Since image segmentation requires pixel level specificity, unlike bounding boxes, this naturally led to inaccuracies.*

This problem by cleverly adjusting RoIPool to be more precisely aligned using RoIAlign. In RoIPool, we would round this down and select 2 pixels causing a slight misalignment. However, in RoIAlign, we avoid such rounding. Instead, we use bilinear interpolation to get a precise idea of what would be at pixel 2.93. This, at a high level, is what allows us to avoid the misalignments caused by RoIPool.

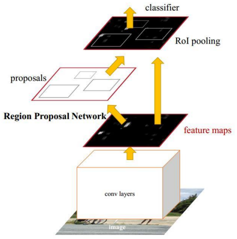
Once these masks are generated, Mask R-CNN combines them with the classifications and bounding boxes from Faster R-CNN to generate such wonderfully precise segmentation.

7. Comparison


Predict BBox offsets with regression
BBox reg
SVMs
Classify regions with SVMs
ConvNet
Forward each region through ConvNet
Warped image regions
Regions of Interest (RoI) from a proposal method (~2k)
Input image



Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.



	R-CNN	**Fast R-CNN**	**Faster R-CNN**
Method for Generating Region Proposal	**Selective Search**	**Selective Search**	**Region Proposal Network**
The mAP on Pascal VOC 2007 test dataset (%)	**58.5**	**66.9 (when trained with VOC 2007 only)** **70.0 (when trained with VOC 2007 and 2012 both)**	**69.9 (when trained with VOC 2007 only)** **73.2(when trained with VOC 2007 and 2012 both)**

			78.8 (when trained with VOC 2007 and 2012 and COCO)

The mAP on Pascal VOC 2012 test dataset (%)	**53.3**	**65.7 (when trained with VOC 2012 only)**
		68.4 (when trained with VOC 2007 and 2012 both)
			67.0(when trained with VOC 2012 only)
			70.4 (when trained with VOC 2007 and 2012 both)
			75.9(when trained with VOC 2007 and 2012 and COCO)

Detection Time (sec)	**~49 (with region proposal generation)**	**~2.32(with region proposal generation)**	**0.2 (with VGG),** **0.059 (with ZF)**

R-CNN generates the region proposals using selective search algorithm first and then computes features for each proposal using a large CNN. It takes R-CNN 50s to test one image. And because R-CNN involves three models separately without much shared computation , it takes 84 hours to train the model.

Fast R-CNN also applies an external selective search algorithm to find regions in the image, but it swaps the sequence of generating region proposals and the use of CNN, so the computation of convolutional layers among proposals for an image is shared, which successfully shorter the test time to 2s. And Fast R-CNN trains the whole system end-to- end all at once, so it only takes 9.5 hours (using VGG-16 CNN on PASCAL VOC 2007) to train the model. The mAP is improved as well compared to R-CNN.

Faster R-CNN does not use external region proposal method anymore, instead it inserts a Region Proposal Network (RPN) after the last convolutional layer, so this really bring down the test time per image to 0.2s, which is nearly cost-free.

Mask R-CNN uses the same basic architecture as Faster R-CNN, and addition to that, it adds a fully convolution layer to locate objects at the pixel level and further increase the accuracy of object detection.

Pascal VOC2007, VOC2007, and MSCOCO are three most commonly used datasets for evaluating detection algorithms. Pascal VOC2012 and VOC2007 are mid-scale datasets with 2 or 3 objects per image and the range of object size in VOC dataset is not large. For MSCOCO, there are nearly 10 objects per image and the majority object are small objects with large scale ranges [5]. R-CNN, Fast R-CNN, Faster R-CNN used these datasets to train and test, so the results should be convincing.

8. Conclusion

A deep convolutional network (DCN) is a type of artificial neural network that is often used for object detection tasks. DCNs are well-suited for this task because they can learn hierarchical representations of objects, which allows them to effectively identify objects in images and video.

One popular type of DCN for object detection is the convolutional neural network (CNN). CNNs are composed of multiple layers of interconnected nodes, which are designed to automatically *and adaptively learn spatial hierarchies of features from input data. This allows CNNs to effectively learn to recognize objects in images.*

Another type of DCN for object detection is the region-based convolutional neural network (R-CNN). R-CNNs are a type of DCN that first uses a selective search algorithm to identify potential object regions in an image, and then uses a CNN to classify each region and refine the bounding box coordinates. This allows R-CNNs to achieve high accuracy in object detection.

A further development of R-CNNs is the fast R-CNN, which uses a single convolutional network to both identify potential object regions and classify each region. This makes fast R-CNNs faster and more efficient than R-CNNs.

Finally, a more recent development is the faster R-CNN, which uses a region proposal network (RPN) to generate object region proposals. This allows faster R-CNNs to be even faster and more efficient than fast R-CNNs.

Overall, DCNs have proven to be effective for object detection tasks, and the various types of DCNs (including CNNs, R-CNNs, fast R-CNNs, and faster R-CNNs) have each contributed to the advancement of this field.

References

1. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation - Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 580-587

2. Feature Pyramid Networks for Object Detection - Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2117-2125

3. Mask R-CNN With Pyramid Attention Network for Scene Text Detection - Zhida Huang; Zhuoyao Zhong; Lei Sun; Qiang Huo

4. Object Detection with Discriminatively Trained Part-Based Models - Ross Girshick; Jeff Donahue; Trevor Darrell; Jitendra Malik

5. Object detection based on RGC mask R-CNN - Minghu Wu,Hanhui Yue,Juan Wang, Yongxi Huang, Min Liu, Yuhan Jiang,Cong Ke, Cheng Zeng

6. Fast R-CNN - Ross Girshick; Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440-1448

7. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks - Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun

8. Research on Semantic Segmentation of High-resolution Remote Sensing Image Based on Full Convolutional Neural Network - Xiaomeng Fu, Huiming Qu

9. Image Semantic Segmentation Using Deep Convolutional Nets, Fully Connected Conditional Random Fields, and Dilated Convolution - Degui Xiao, Pei Zhong

10. Object Detection System for Self-Checkout Cashier System Based on Faster Region-Based Convolution Neural Network and YOLO9000 - Michael Ariyanto, Prima Dewi Purnamasari

11. Detection and Recognition of Security Detection Object Based on Yolo9000 - Zhongqiu Liu, Jianchao Li, Yuan Shu, Dongping Zhang

12. Neural Networks and Gradient-Based Learning in OCR – Y. LeCun

13. Improved Object Detection in Video Surveillance Using Deep Convolutional Neural Network Learning - Dhiyanesh B, Rajesh Kanna K, Rajkumar S, Radha R

14. Object Detection and Tracking Based on Convolutional Neural Networks for High-Resolution Optical Remote Sensing Video - Biao Hou, Jingliang Li, Xiangrong Zhang, Shuang Wang, Licheng Jiao

15. Complex network classification with convolutional neural network - Ruyue Xin, Jiang Zhang, Yitong Shao