Marathwada Shikshan Prasarak Mandal's
Deogiri Institute of Engineering and Management Studies, Aurangabad

Seminar Report

On

Object Detection

Submitted By

Rajeshree Shegaonkar (36072)

Dr. Babasaheb Ambedkar Technological University
Lonere (M.S.)

Department of Computer Science and Engineering
Deogiri Institute of Engineering and Management Studies, Aurangabad
(2019- 2020)

Seminar Report

Object Detection

Submitted By

Rajeshree Shegaonkar (36072)

In partial fulfillment of

Bachelor of Technology

(Computer Science & Engineering)

Guided By

Mrs. Amruta Joshi

Department of Computer Science & Engineering

Deogiri Institute of Engineering and Management Studies,

Aurangabad

(2019- 2020)

CERTIFICATE

This is to certify that, the Seminar entitled "**Object Detection**" submitted by **Rajeshree Shegaonkar** is a bonafide work completed under my supervision and guidance in partial fulfillment for award of Bachelor of Technology (Computer Science and Engineering) Degree of Dr. Babasaheb Ambedkar Technological University, Lonere.

Place: Aurangabad
Date:

Mrs. Amruta Joshi
Guide

Mr. S.B. Kalyankar
Head

Dr. Ulhas D. Shiurkar
Director,
Deogiri Institute of Engineering and Management Studies,
Aurangabad

Abstract

Object Detection

Object detection is the task of image classification with localization, although an image may contain multiple objects that require localization and classification.

This is a more challenging task than simple image classification or image classification with localization, as often there are multiple objects in the image of different types.

Often, techniques developed for image classification with localization are used and demonstrated for object detection.

Some examples of object detection include:

Drawing a bounding box and labelling each object in a street scene.
Drawing a bounding box and labelling each object in an indoor photograph.
Drawing a bounding box and labelling each object in a landscape.

Contents

List of Abbreviations

Sr.No	Acronym	Abbreviations
1	CNN	convolutional neural networks
2	RCNN	Region Based Convolution Neural Network
3	Fast RCNN	Fast Region Based Convolution Neural Network

List of Figure

1.INTRODUCTION

Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. Well-researched domains of object detection include face detection and pedestrian detection. Object detection has applications in many areas of computer vision, including image retrieval and video surveillance.

How much time have you spent looking for lost room keys in an untidy and messy house? It happens to the best of us and till date remains an incredibly frustrating experience. But what if a simple computer algorithm could locate your keys in a matter of milliseconds?

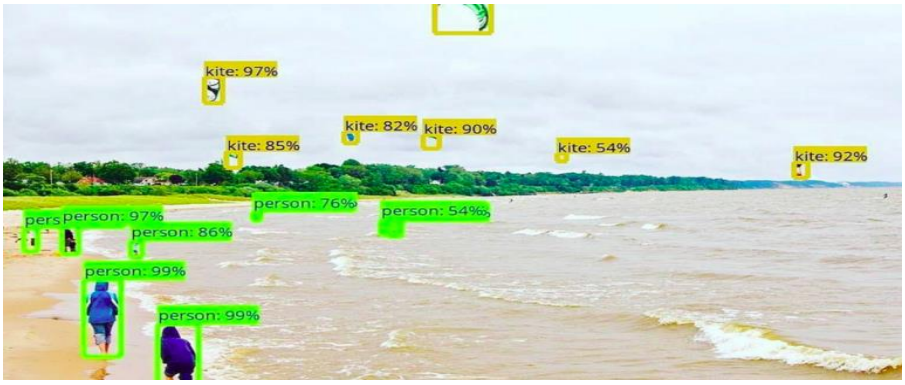


Fig1.1 example of object detection

That is the power of object detection algorithms. While this was a simple example, the applications of object detection span multiple and diverse industries, from round-the-clock surveillance to real-time vehicle detection in smart cities. In short, these are powerful deep learning algorithms.

In this article specifically, we will dive deeper and look at various algorithms that can be used for object detection. We will start with the algorithms belonging to RCNN family, i.e. RCNN, Fast RCNN and Faster RCNN. In the upcoming article of this series, we will cover more advanced algorithms like YOLO, SSD, etc.

The below image is a popular example of illustrating how an object detection algorithm works. Each object in the image, from a person to a kite, have been located and idntified with a certain level of precision.Let’s start with the simplest deep learning approach, and a widely used one, for detecting objects in images – Convolutional Neural Networks or CNNs. If your understanding of CNNs is a little rusty**.**

Humans can easily detect and identify objects present in an image. The human visual system is fast and accurate and can perform complex tasks like identifying multiple objects and detect obstacles with little conscious thought. With the availability of large amounts of data, faster GPUs, and better algorithms, we can now easily train computers to detect and classify multiple objects within an image with high accuracy. In this blog, we will explore terms such as object detection, object localization, loss function for object detection and localization, and finally explore an object detection algorithm known as "You only look once" (YOLO).

1.1 Object Localization

An image classification or image recognition model simply detect the probability of an object in an image. In contrast to this, object localization refers to identifying the location of an object in the image. An object localization algorithm will output the coordinates of the location of an object with respect to the image. In computer vision, the most popular way to localize an object in an image is to represent its location with the help of bounding boxes. Fig. 1 shows an example of a bounding box.

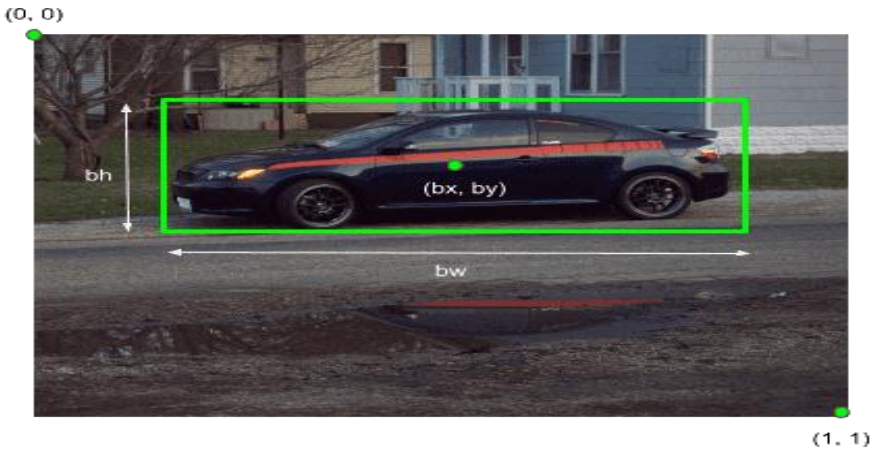


Fig1. 2. Bounding box representation used for object localization

A bounding box can be initialized using the following parameters:

bx, by : coordinates of the center of the bounding box

bw : width of the bounding box w.r.t the image width

2.LITERATURE SURVEY

2.1 Introduction

An approach to building an object detection is to first build a classifier that can classify closely cropped images of an object. shows an example of such a model, where a model is trained on a dataset of closely cropped images of a car and the model predicts the probability of an image being a car.

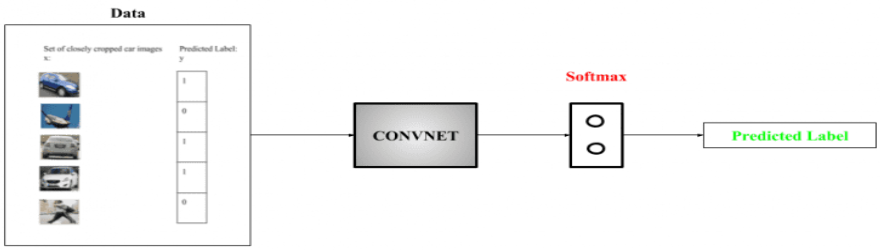


Fig2.1: Detecting car

object detection, localization, self driving cars, computer vision, deep learning, classification Now, we can use this model to detect cars using a sliding window mechanism. In a sliding window mechanism, we use a sliding window (similar to the one used in convolutional networks) and crop a part of the image in each slide. The size of the crop is the same as the size of the sliding window. Each cropped image is then passed to a ConvNet model, which in turn predicts the probability of the cropped image is a car.

Object detection is the identification of an object in the image along with its localisation and classification. It has wide spread applications and is a critical component for vision based software systems. This paper seeks to perform a rigorous survey of modern object detection algorithms that use deep learning. As part of the survey, the topics explored include various algorithms, quality metrics, speed/size trade offs and training methodologies. This paper focuses on the two types of object detection algorithms- the SSD class of single step detectors and the Faster R-CNN class of two step detectors. Techniques to construct detectors that are portable and fast on low powered devices are also addressed by exploring new lightweight convolutional base architectures. Ultimately.

2.2 Description/ Feature selection

Objects can be of two types rigid and non rigid. Rigid objects are vehicles and non rigid objects are humans. .[laptop prof ds patil].Since these object vary there identification traits also varies for example in cases humans facial expressions, height, features , speed of walking which will be mostly be very low as compared to vehicles is of major concern. In case of non rigid objects shape, color, edges, region, velocity of vehicle will be main areas of concern. When object is vehicle then again it can be divided based on homogenous and heterogeneous traffic. Homogenous traffic where all vehicles are moving with almost same speed and heterogeneous was unsynchronized and unregulated flow of traffic is there .Therefore based on image features and image movement ,many authors have proposed various mechanisms with the help of which image recognition became an easy task.

2.3 Activity Analysis and Action taken

In this phase threshold level is already fixed and in case of suspicious object going beyond it will raise an alarm and alert others. In case of object being below threshold no action will be taken.

2.4 Basic approaches for object detection and classification

1. Background subtraction algorithm: In this method current image is subtracted from reference background image, has been classified into two categories

a)Simple background subtraction :In this method problem is that if reference background is dynamically inserted then it fails.

b)Running average method: It overcomes simple background subtraction method , as it takes average of all the frames n then set the reference frames.

2.Temporal Differencing[:In this method , differencing of pixels are used for two or more consecutive frames in the video sequence to detect moving regions, but problem with this method is that it doesn’t work when object is of uniform texture or is moving slowly.

3. Statistical methods :They remove the problems encountered in background subtraction method, as it uses properties of pixels to build up a new background model

2.5 Problems in object detection and few Solutions

1. Noise in image: With the use of filters like median , low pass filters, Gaussian low pass filters and many more reduces the noise in images

2.Shadow in image: Several shadow detection methods exists like hypotheses based that shadowed zone is darker than illuminated zone. region growing method,comparison based on different shadow models.

3.Occlusion in dense scenes: Combination –of-parts(COP) and further a mechanism of global occlusion reasoning approach is used to solve this problem

4. False alarms: Threshold based method is advised to figure out this problem but still lot of scope is left in this area.

2.6 Methods

Methods for object detection generally fall into either machine learning-based approaches or deep learning-based approaches. For Machine Learning approaches, it becomes necessary to first define features using one of the methods below, then using a technique such as support vector machine (SVM) to do the classification. On the other hand, deep learning techniques that are able to do end-to-end object detection without specifically defining features, and are typically based on convolutional neural networks (CNN).

3.BRIF ON SYSTEM

3.1 Working

Every object class has its own special features that helps in classifying the class – for example all circles are round. Object class detection uses these special features. For example, when looking for circles, objects that are at a particular distance from a point (i.e. the center) are sought. Similarly, when looking for squares, objects that are perpendicular at corners and have equal side lengths are needed. A similar approach is used for face identification where eyes, nose, and lips can be found and features like skin color and distance between eyes can be found.

Methods for object detection generally fall into either machine learning-based approaches or deep learning-based approaches. For Machine Learning approaches, it becomes necessary to first define features using one of the methods below, then using a technique such as support vector machine (SVM) to do the classification. On the other hand, deep learning techniques that are able to do end-to-end object detection without specifically defining features, and are typically based on convolutional neural networks (CNN).

3.2 Algorithms

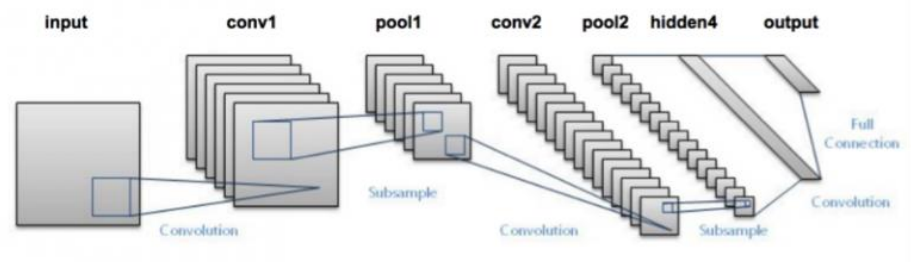
1)CNN

2)RCNN (Region Based Convolution Neural Network)

3)Fast RCNN

1)CNN

Fig 3.1: briefly summarize the inner workings of a CNN .



For each input image, we get a corresponding class as an output. Can we use this technique to detect various objects in an image? Yes, we can! Let's look at how we can solve a general object detection problem using a CNN.

1. First, we take an image as input:

Fig 3.2: Input image

2. Then we divide the image into various regions:

Fig 3.3: divide the image into various regions

3. We will then consider each region as a separate image.

4. Pass all these regions (images) to the CNN and classify them into various classes.

The problem with using this approach is that the objects in the image can have different aspect ratios and spatial locations. For instance, in some cases the object might be covering most of the image, while in others the object might only be covering a small percentage of the image. The shapes of the objects might also be different (happens a lot in real-life use cases).

As a result of these factors, we would require a very large number of regions resulting in a huge amount of computational time. So to solve this problem and reduce the number of regions, we can use region-based CNN, which selects the regions using a proposal method. Let's understand what this region-based CNN can do for us.

2. Understanding Region-Based Convolutional Neural Network

2.1 Intuition of RCNN

Instead of working on a massive number of regions, the RCNN algorithm proposes a bunch of boxes in the image and checks if any of these boxes contain any object. RCNN uses selective search to extract these boxes from an image (these boxes are called regions).

Let's first understand what selective search is and how it identifies the different regions. There are basically four regions that form an object: varying scales, colors, textures, and enclosure. Selective search identifies these patterns in the image and based on that, proposes various regions. Here is a brief overview of how selective search works:

It first takes an image as input:

Fig 3.5: Input image

Then, it generates initial sub-segmentations so that we have multiple regions from this image:

Fig 3.6: Multiple regions from this image

The technique then combines the similar regions to form a larger region (based on color similarity, texture similarity, size similarity, and shape compatibility):

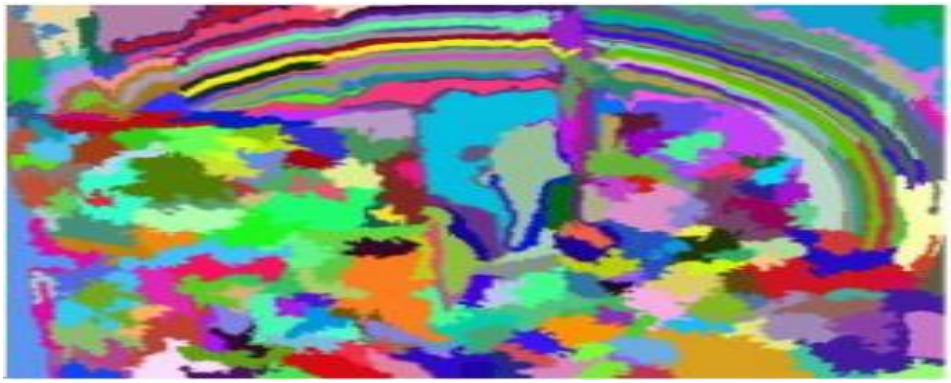


Fig 3.7: Combines the similar regions to form a larger

Finally, these regions then produce the final object locations (Region of Interest).

Below is a succint summary of the steps followed in RCNN to detect objects:

- We first take a pre-trained convolutional neural network.
- Then, this model is retrained. We train the last layer of the network based on the number of classes that need to be detected.
- The third step is to get the Region of Interest for each image. We then reshape all these regions so that they can match the CNN input size.
    - After getting the regions, we train SVM to classify objects and background.

- Finally, we train a linear regression model to generate tighter bounding boxes for each identified object in the image.You might get a better idea of the above steps with a visual example (Images for the example shown below are taken from this paper) . So let's take one!
- First, an image is taken as an input:

Fig 3.8

Then, we get the Regions of Interest (ROI) using some proposal method (for example, selective search as seen above):

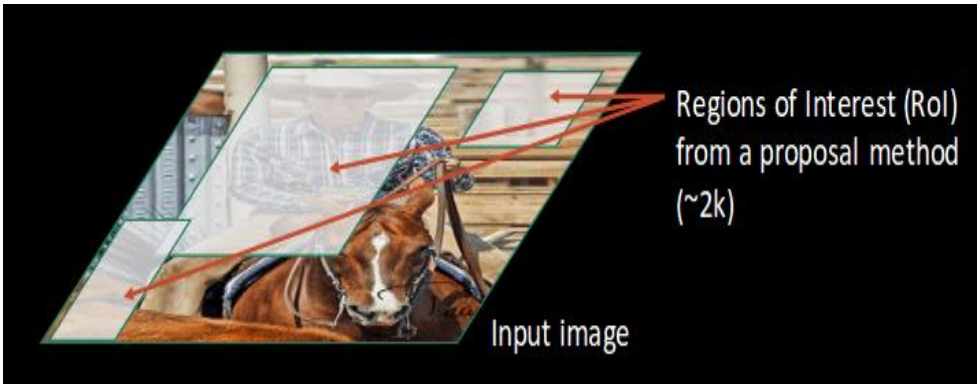


Fig 3.9: Region is passed to the Convnet

All these regions are then reshaped as per the input of the CNN, and each region is passed to the ConvNet:

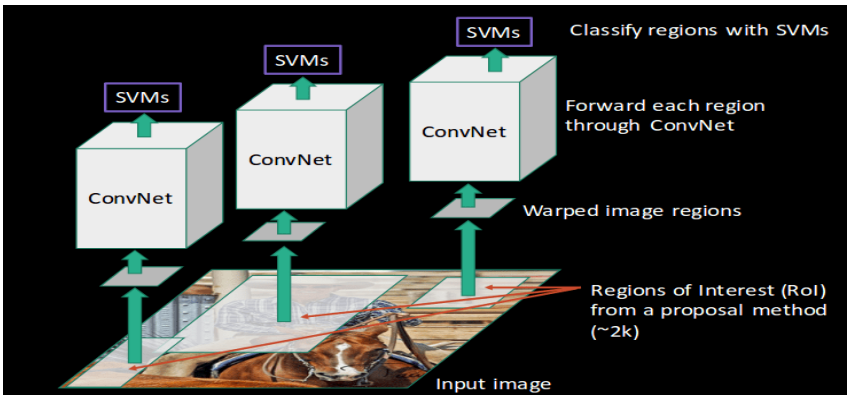


Fig 3.10: Image is taken as an input

CNN then extracts features for each region and SVMs are used to divide these regions into different classes:

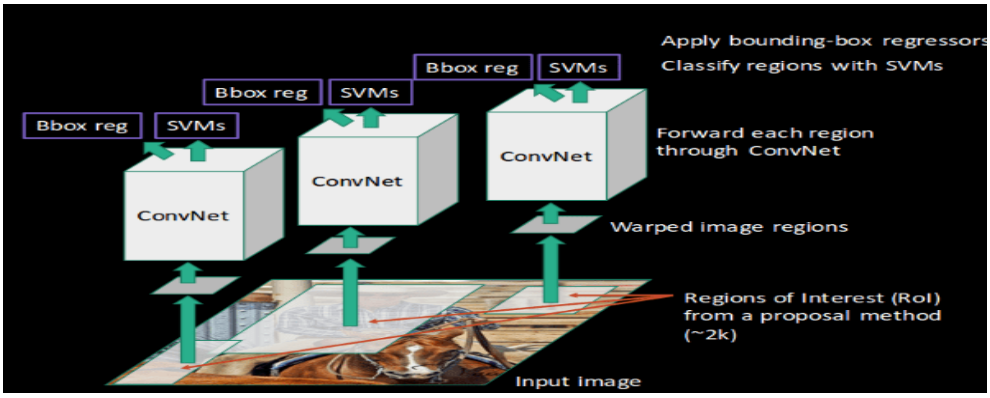


Fig 3.11: extracts features for each region

Finally, a bounding box regression (Bbox reg) is used to predict the bounding boxes for each identified region:And this, in a nutshell, is how an RCNN helps us to detect objects.

2.2 Problems with RCNN

So far, we've seen how RCNN can be helpful for object detection. But this technique comes with its own limitations. Training an RCNN model is expensive and slow thanks to the below steps:

- Extracting 2,000 regions for each image based on selective search
- Extracting features using CNN for every image region. Suppose we have N images, then the number of CNN features will be N*2,000
- The entire process of object detection using RCNN has three models:
- CNN for feature extraction

All these processes combine to make RCNN very slow. It takes around 40-50 seconds to make predictions for each new image, which essentially makes the model cumbersome and practically impossible to build when faced with a gigantic dataset.

3. Understanding Fast RCNN

Intuition of Fast RCNN

What else can we do to reduce the computation time a RCNN algorithm typically takes? Instead of running a CNN 2,000 times per image, we can run it just once per image and get all the regions of interest (regions containing some object).Ross Girshick, the author of RCNN, came up with this idea of running the CNN just once per image and then finding a way to share that computation across the 2,000 regions. In Fast RCNN, we feed the input image to the CNN, which in turn generates the convolutional feature maps. Using these maps, the regions of proposals are extracted. We then use a RoI pooling layer to reshape all the proposed regions into a fixed size, so that it can be fed into a fully connected network.

Let's break this down into steps to simplify the concept:

- As with the earlier two techniques, we take an image as an input.
- This image is passed to a ConvNet which in turns generates the Regions of Interest.
- A RoI pooling layer is applied on all of these regions to reshape them as per the input of the ConvNet. Then, each region is passed on to a fully connected network.
- A softmax layer is used on top of the fully connected network to output classes. Along with the softmax layer, a linear regression layer is also used parallely to output bounding box coordinates for predicted classes.

So, instead of using three different models (like in RCNN), Fast RCNN uses a single model which extracts features from the regions, divides them into different classes, and returns the boundary boxes for the identified classes simultaneously.To break this down even further, I'll visualize each step to add a practical angle to the explanation.

We follow the now well-known step of taking an image as input:

Fig 3.12

This image is passed to a ConvNet which returns the region of interests accordingly:

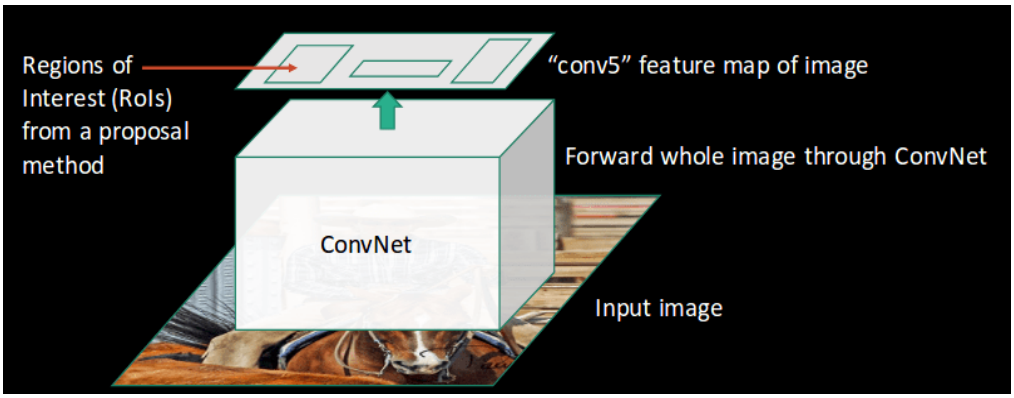
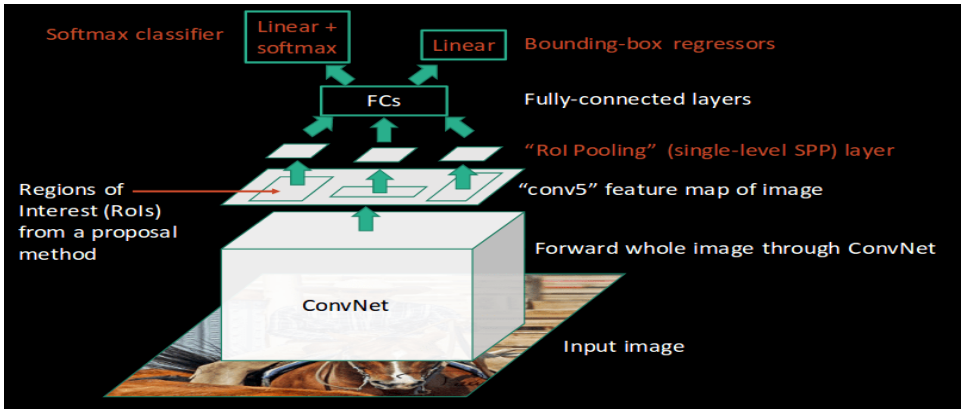


Fig 3.13

Then we apply the RoI pooling layer on the extracted regions of interest to make sure all the regions are of the same size:



Finally, these regions are passed on to a fully connected network which classifies them, as well as returns the bounding boxes using softmax and linear regression layers simultaneously:

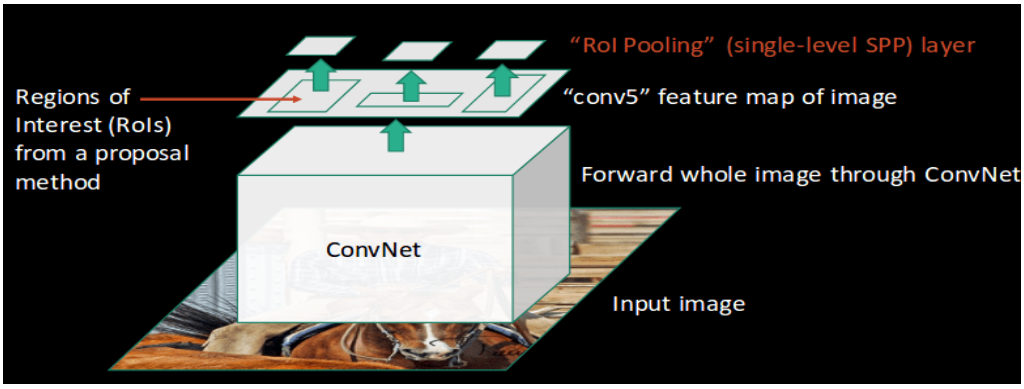


Fig 3.14: a fully connected network

This is how Fast RCNN resolves two major issues of RCNN, i.e., passing one instead of 2,000 regions per image to the ConvNet, and using one instead of three different models for extracting features, classification and generating bounding boxes.

3.2 Problems with Fast RCNN

But even Fast RCNN has certain problem areas. It also uses selective search as a proposal method to find the Regions of Interest, which is a slow and time consuming process. It takes around 2 seconds per image to detect objects, which is much better compared to RCNN. But when we consider large real-life datasets, then even a Fast RCNN doesn't look so fast anymore.

4.CONCLUSION

4.1 Conclusion

Object detection is a fascinating field, and is rightly seeing a ton of traction in commercial, as well as research applications. Thanks to advances in modern hardware and computational resources, breakthroughs in this space have been quick and ground-breaking.

We now have a better understanding of how we can localize objects while classifying them in an image. We also learned to combine the concept of classification and localization with the convolutional implementation of the sliding window to build an object detection system. In the next blog, we will go deeper into the YOLO algorithm, loss function used, and implement some ideas that make the YOLO algorithm better. Also, we will learn to implement the YOLO algorithm in real time.

4.2 Applications

1. Object detection and recognition is applied in many areas of computer vision, including image retrieval, security, surveillance, automated vehicle systems and machine inspection.
2. Hospitals,industries
3. It is widely used in computer vision tasks such as face detection, face recognition, video object co-segmentation. It is also used in tracking objects, for example tracking a ball during a football match, tracking movement of a cricket bat, or tracking a person in a video.

References

[1]https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

[2]https://www.hackerearth.com/blog/developers/introduction-to-object-detection/

[3]https://en.m.wikipedia.org/wiki/Object_detection#cite_ref-5

[4]Multiple object class detection

[5]https://arxiv.org/abs/1808.07256

ACKNOWLEDGEMENT

I would like to place on record my deep sense of gratitude to **Prof.S.B.Kalyankar**, HOD-Dept. of Computer Science and Engineering, Deogiri Institute of Engineering and management Studies Aurangabad, for his generous guidance, help and useful suggestions.

I express my sincere gratitude to **Prof.Amruta Joshi**, Dept. of Computer Science and Engineering, Deogiri Institute of Engineering and management Studies Aurangabad, for her stimulating guidance, continuous encouragement and supervision throughout the course of present work.

I am extremely thankful to **Dr.Ulhas.D.Shiurkar**, Director, Deogiri Institute of Engineering and management Studies Aurangabad, for providing me infrastructural facilities to work in, without which this work would not have been possible.

Signature of Student

Name of Student

Rajeshree Shegaonkar

Sign