International Journal of Research in Engineering, Science and Management
Volume-3, Issue-8, August-2020
journals.resaim.com/ijresm | RESAIM Publishers

Object Detection Using Deep Learning

H. Harshita Kumar[1*], Harshitha Manjunath[2], Chandana Shivanna[3], Mangala Manjunath[4]
[1,2,3,4]*UG Student, Department of Computer Science and Engineering, Sri Venkateshwara College of Engineering, Bangalore, India*
*Corresponding author: harshita236@gmail.com

***Abstract*: This paper proposes an efficient and an accurate object detection system using Deep Learning [3]. Compared to the existing systems, it makes sure it deduces the drawbacks and provides a better solution. In a certain image as an input, the system divides the image into bounding boxes [7] and associates the class probabilities. In one evaluation, a single neural network predicts the bounding boxes and its probabilities. These systems use classifier for a specific object in an image and evaluate it at every location, corner of the image. Once the classification is done, it makes sure there are no duplicate detections and labels the objects with its accuracy score. Since our system uses deep learning, it makes use of Convolutional Neural Network algorithm to compute the images in a more perfect manner. Our architecture is less complicated and thus makes it fast to process.**

***Keywords*: Activation function, Bounding Box, Convolution, Deep Learning, Feature map, Object detection, Pooling, Softmax, Testing, Training.**

1. Introduction

Deep convolutional neural networks [3] have achieved state-of-the-art performance on various image recognition fields. In this work, we tend to propose a saliency-inspired neural network model to predict a collection of class-agnostic bounding boxes [7] and its associated score for every box, like its probability of containing any object of interest. There are mainly three main steps to generalize how the object detection takes place:

- *Image Classification:* In this, we present an image and allow the system to classify the objects in that image according to classes.
- *Image Localization:* In this, we predict the location of every object in that image by highlighting using a bounding box.
- *Object Detection [4]:* The complete process of classification and localization gives us the object detection system, which has bounding boxes for every object in an image along with its label containing the name and its confidence score.

Humans can understand and recognize various objects in an image very quick and efficiently.

Since machine learning [2] has been deployed and recognized as a blooming field, its mainly because of the way the machines are taught to process the task. It is mainly because of various algorithms, GPUs, larger sets of data.

Same implies to our object detection [4] system in this scenario where respected resources are used to make sure the system works efficiently, quick, reduces errors and deduces complications.

Image detection has become easy in still images without any doubt, but when it comes to the videos, the detection results should not have dramatic changes over time in terms of both bounding box locations and detection confidences [5].

While we are talking about the existing systems, ResNet it is considered to be deeper than the VGG-16 layer. But later it was found that the accuracy was decreased constantly.

Apart from this, there was SSD and MANet [9] introduced relatively for object detection [4], these two techniques localized every object at every location to label it according to the classes. When it came to its computing power, accuracy and efficiency these techniques showed poor result.

Henceforth, Deep CNN was introduced, CNN is a type of feed-forward neural network and works on principle of weight sharing. Convolution is an integration showing how one function overlaps with other function and is a blend of two functions being multiplied.

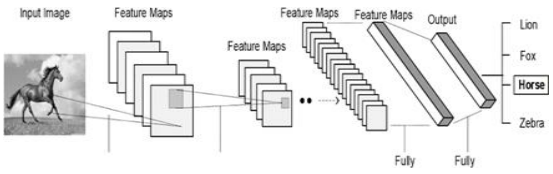

Fig. 1. Overview of project

The above figure shows an example of how detection is done.

The image is convolved with activation function to give the feature maps [1]. Each feature maps thus produces further feature maps, it undergoes pooling and convolving to reach the softmax layer which consists of class probabilities in the form of 0s and 1s. Hence the objects are classified along with its bounding boxes [7].

The advantages of Deep CNN (proposed system) are,

- A Deeper architecture provides an exponentially increased expressive capability.
- The architecture of CNN optimizes several related tasks together.
- Using deep CNN, some classical computer vision challenges can be recast as high-dimensional data transform problems and solved from a different viewpoint.

2. System Design

Every task that has to be fed to a system, has to go through certain steps, which are as follows:

1. Initially, dataset has to be collected and splitted into training set (80%) and testing set (20%). The training set is usually when the system has to train or learn itself from the set given to it using the respective algorithms. Once it self learns and gives the expected output, it can be tested upon the other set of dataset called as testing set.
2. In this project, since we are dealing with images and its detection, we can make our system more accurate and efficient by introducing various images to make sure that our system has vast knowledge on how the model works and gives us the expected evaluation.
3. The result of evaluation for every input can be compared against the set of labels already given to the system, which is commonly known as Supervised Learning.
4. If the expected and actual output match, the model is considered to be working properly else it needs to modify certain data (which is usually the weights given to the hidden layers) and reprocess it to get the exact final output.

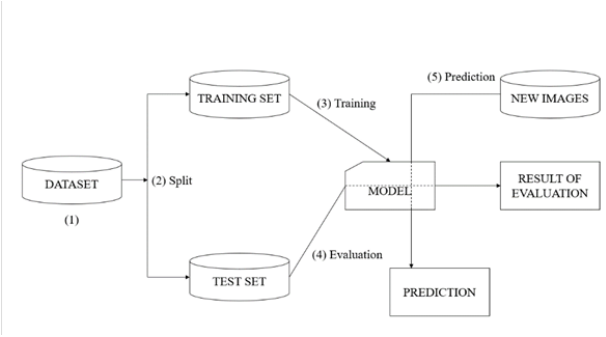

Fig. 2. Work flow

3. System Architecture

Since our design is just the outer look of how our project works. The heart of project lies in our architecture. Every neural network deal with three layers, input layer, hidden layers and output layers. The architecture starts from input layer to the output layer. The hidden layers are where the exact computation and architecture knowledge, algorithm is applied.

Input Layer:

It is merely some memory that holds the input values.

The input layer does no processing and is just a placeholder where we can temporarily store the input data.

Output Layer:

The output layer is where the network's results are communicated to the world outside of the network.

If it is a binary classifier, we can use just one neuron with output values from 0 to 1.

A multi-class classifier will typically have as many outputs as there are classes.

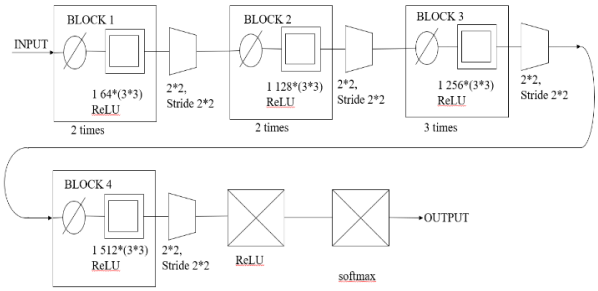

Fig. 3. Architecture of CNN

Fully Connected layer:

A fully-connected layer also called an FC or dense layer is a set of neurons that each receive an input from every neuron on the previous layer.

For example, if there are 4 neurons in the dense layer, and 4 neurons in preceding layer, then each neuron in this layer will have 4 inputs, one from each neuron in the preceding layer, for a total of 4x4 = 16 connections.

Dropout:

Overfitting is a problem for many neural networks. As soon as network starts to memorize the training data, and is therefore overfitting, we typically stop training.

Any techniques we can use that delay the onset of overfitting is a type of Regularization.

Regularization methods are great because they allow us to train our networks for longer before they overfit, giving us better performance. One such techniques for delaying overfitting is called Dropout, and it can be used in deep network with the inclusion of a dropout layer.

The dropout layer does not contain any neurons. Unlike the softmax layer, the dropout layer does not even do any computing. Instead, it just temporarily disconnects some of the neurons on the previous layer. This layer is only active during training. When we use the network for predicting, the dropout layers have no effect.

The intention behind dropout is to prevent any of our neurons from over-specializing. Suppose that one neuron in a photo-classification system gets highly specialized to detect, say, the eyes of cats. That's useful for recognizing picture of cat's faces, but useless for all the other photographs the system might be asked to classify.

Batch Normalization:

Another regularization technique is called Batch Normalization, often referred to simply as batchnorm.

Like, dropout, batchnorm can be implemented as a layer that we include in our network, but this layer also doesn't contain neurons. Unlike dropout, batchnorm actually does perform some computation, though there are no parameters and nothing for us to control.

Batchnorm is used to modify the values that come out of a computation layer, such as a fully-connected layer, or one of the layers.

Filter:

A small window of a fixed size used to slide over the complete image to produce the feature maps.

Stride:

It is the step count at which the filter moves.

Convolution Layer [8]:

Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel.

Pooling Layer:

Once the filter slides over the complete image, feature maps are formed, the pooling layer [10] takes the specific and most appropriate feature from the filter to produce the building block of next feature map.

There are two ways to do pooling, maximum and average pooling. We use maximum pooling in our paper.

Activation function [5]: Once layer undergoes convolution and pooling; it makes sure it undergoes a function called activation function [5]. This function adds some nonlinear property which is a neural network. Without the activation functions, the neural network could perform only linear mappings from inputs x to the outputs y.

Common and most suggested form of activation function we use here is ReLU (Rectified Linear Unit).

Softmax Layer:

Gives us the class probabilities in form of 0s and 1s.

Hence an image given as an input undergoes our architecture, consisting of many layers for computation and gives us the final expected output.

4. Observations and Result

Fig. 4(a) and Fig. 4(c) gives us the output for the image input shown in Fig. 4(a).

It consists of bounding boxes [7] around the detected objects and its respective confidence scores.

Before:

Fig. 4(a). Original image

After:

```
It took 64.774 seconds to detect the objects in the image.

Number of Objects Detected: 28

Objects Found and Confidence Level:

1. person: 0.999996
2. person: 1.000000
3. car: 0.707237
4. truck: 0.933031
5. car: 0.658086
6. truck: 0.666982
7. person: 1.000000
8. traffic light: 1.000000
9. person: 1.000000
10. car: 0.997369
11. bus: 0.998023
12. person: 1.000000
13. person: 1.000000
14. person: 1.000000
15. person: 1.000000
16. person: 1.000000
17. traffic light: 1.000000
18. traffic light: 1.000000
19. handbag: 0.997282
20. traffic light: 1.000000
21. car: 0.989741
22. traffic light: 1.000000
23. traffic light: 0.999999
24. person: 0.999999
25. truck: 0.715035
26. traffic light: 1.000000
27. person: 0.999993
28. person: 0.999996
```

Fig. 4(b). Confidence scores

Fig. 4(c). Detected objects

5. Conclusion

We have deployed a very efficient form of object detection [4] which is using Deep CNN [3], which is comparatively considered to be far better than any other existing systems. It not only gives us the location of objects abounded by the box but also gives us the confidence score.

Object detection has proven to be useful and very important in many fields of medical science, engineering, self-driving cars, video surveillances detections, facial detections [6].

References

[1] S. Ren, K. He, R. B. Girshick, X. Zhang, and J. Sun. Object detection networks on convolutional feature maps. CoRR, 2015.
[2] R. Caruana. Multitask learning. Machine learning, 28(1), 1997.
[3] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In Computer Vision and Pattern

Recognition (CVPR), 2014 IEEE Conference on, pages 2155–2162. IEEE, 2014.

[4] C. P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In Computer vision, 1998 sixth international conference on, pages 555–562. IEEE, 1998.

[5] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition, 2013.

[6] P. Viola and M. J. Jones. Robust real-time face detection. International journal of computer vision, 57(2):137–154,2004.

[7] C. L. Zitnick and P. Dollar. Edge boxes: Locating object proposals from edges. In Computer Vision–ECCV 2014, pages 391–405. Springer, 2014.

[8] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. International journal of computer vision, 104(2):154–171, 2013.

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 2015.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition, 2014.