Image Based Object Detection and Classification

Ayan Kumar Dhara, Department of Information Technology

Abstract
Object detection is widely used in the field of computer vision and crucial for variety of applications, e.g., self-driving car. During the development of half a century, object detection methods have been continuously developed, and generated numerous approaches which obtained promising achievements. At present, the approach of object detection has been largely evolved into two categories which are traditional machine learning methods utilizing varied computer vision techniques and deep learning method. This paper presents a review of object detection techniques. Main schools of deep learning methods such as SSD(Single Shot Detector), RetinaNet, R-CNN and YOLO, are selected for analysis and introduction.

1. Introduction

Object recognition is a general term to describe a collection of related computer vision tasks that involve identifying objects in digital photographs.
Image classification involves predicting the class of one object in an image. *Object localization* refers to identifying the location of one or more objects in an image and drawing abounding box around their extent. *Object detection* combines these two tasks and localizes and classifies one or more objects in an image.

Much of what we know today about visual perception comes from neurophysiological research conducted on cats in the 1950s and 1960s. By studying how neurons react to various stimuli, two scientists observed that human vision is hierarchical. Neurons detect simple features like edges, then feed into more complex features like shapes, and then eventually feed into more complex visual representations.

Armed with this knowledge, computer scientists have focused on recreating human neurological structures in digital form. Like their biological counterparts, computer vision systems take a hierarchical approach to perceiving and analyzing visual stimuli.

During the summer in 1960 a professor at Stanford tasked his student to make computer describe an image, but the results were not great because of the lack of sophisticated algorithms and computational power.

For five decades growth of AI and computer vision sector was extremely small. The biggest breakthrough happened in ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The ILSVRC is an annual image classification competition where research teams evaluate their algorithms on the given data set, and then compete to achieve higher accuracy on several visual recognition tasks. From 2010-2011, the error rate for ILSVRC winners

hovered around 26%. Then, in 2012, a team from the University of Toronto entered a deep neural network called AlexNet that changed the game for artificial intelligence and computer vision projects. AlexNet was the first model to use Convolutional Neural Network (CNN) for object detection.
The use of CNN was the turning point in the field of computer vision. Now most of the robust object detection models are different variations of CNN.

2. Background

The aim of object detection is to detect all instances of objects from a known class, such as people, cars or faces in an image. Generally, only a small number of instances of the object are present in the image, but there is a very large number of possible locations and scales at which they can occur and that need to somehow be explored.

Each detection of the image is reported with some form of pose information. This is as simple as the location of the object, a location and scale, or the extent of the object defined in terms of a bounding box. For example, face detection in a face detector may compute the locations of the eyes, nose and mouth, in addition to the bounding box of the face.

Object detection systems always construct a model for an object class from a set of training examples. In the case of a fixed rigid object in an image, only one example may be needed, but more generally multiple training examples are necessary to capture certain aspects of class variability.

Now there are two options machine learning or deep learning both has pros and cons. Machine learning is a perfect option if accuracy is not a concern and training data is small this method is less robust and cannot deal with complex images sets. In case of deep learning the accuracy is considerably high but a huge training data and large computational resource is also required for training and it deals very well with complex images in varied illumination and overlapping objects. Some use cases include self-driving, automate ophthalmology, surveillance etc

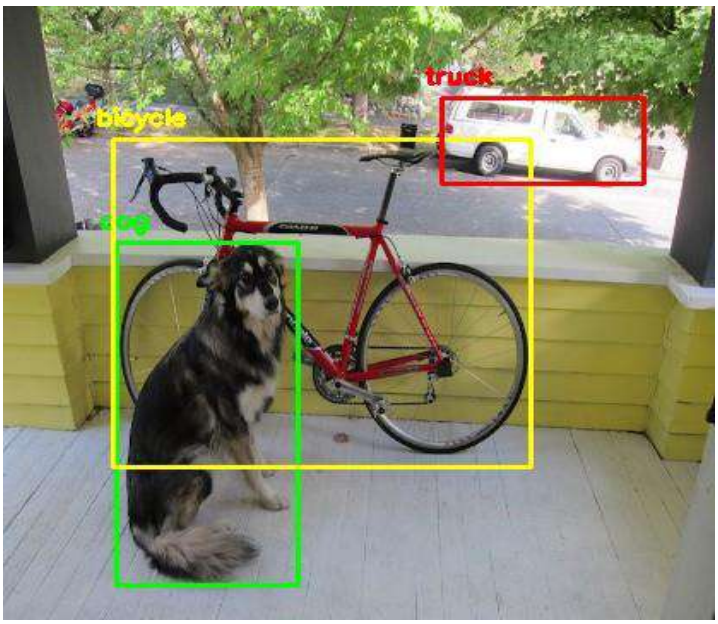


Fig.1 *Detected object marked with bounding box in the image*

3. Previous Works

The literature on object detection is vast and, in this section, we focus on few of them. In 2009, **Felzenszwalb et al. [1]** described an object detection system based on mixtures of multiscale deformable partmodels. **Leibe et al. [2]** in 2007, presented a novel method for detecting and localizing objects of a visual category in cluttered real-world scenes. **Alex et al. [3]** in 2012 designed AlexNet which competed in the ImageNet Large Scale Visual Recognition Challenge on September 30, 2012. The network achieved a top-5 error of 15.3%, more than 10.8 percentage points lower than that of the runner up. I have specifically taken into account some popular fast deep learning method such asg **Ross Girshik et al**. Faster R-CNN **[4]** as the benchmark since it is claimed to be the fastest deep network that could recognize an object in a given image with a confidence. **Girshiks Fast R-CNN [5]** (previous version of Faster R-CNN) has been implemented on **BVLCs Caffe tool 1** leveraging its capability of neural network to extract regions out of images, computing CNN features out of it and Classifying the extracted regions. They have managed to achieve an average segmentation accuracy of 47.9% on the VOC 2011 test set and 58.5% mAP on the 2007 dataset. **Redmon et al. [6]** in 2016 introduced YOLO detection of 9000 categories, YOLO detection method, both novel and drawn from prior work. The improved model, YOLOv2, is state-of-the-art on standard detection tasks like PASCAL VOC and COCO. At 67 FPS, YOLOv2 gets 76.8 mAP on VOC 2007. At 40 FPS, YOLOv2 gets 78.6 mAP, outperforming state-of-the-art methods like Faster RCNN with ResNet and SSD while still running significantly faster. In this case I am using YOLO method which gained popularity because of its speed and is competitive edge against faster RCNN.

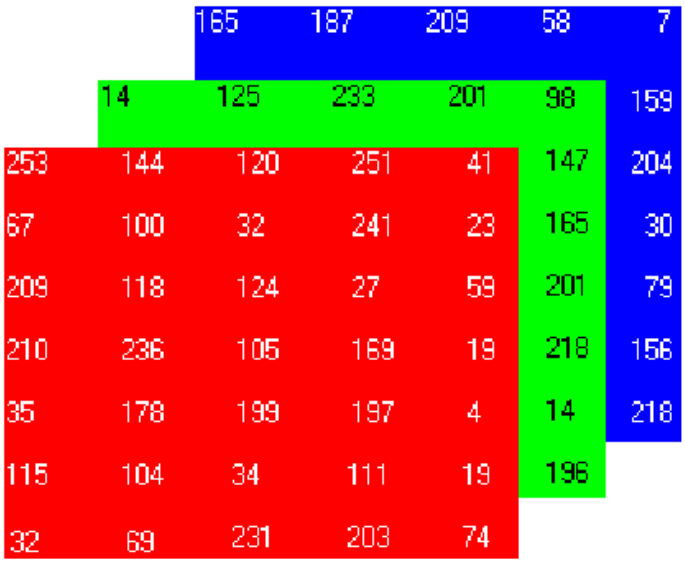
4. Overview of Convolutional Neural Networks (CNNs)

CNNs primarily focus on the basis that the input will be comprised of images. This focuses the architecture to be set up in way to best suit the need for dealing with the specific type of data.

CNNs are comprised of three fundamental types of layers. These are convolutional layers, pooling layers and fully-connected layers. When these layers are stacked, a CNN architecture has been formed.

The basic functionality of the example CNN above can be broken down into four key areas.

i. The **input layer** will hold the pixel values of the image. Which can be binary image, monochromatic or color encode pixel value as RGB of 8-bit value.



ii. The **convolutional layer** will determine the output of neurons of which are connected to local regions of the input through the calculation of the scalar product between their weights and the region connected to the input volume.

The rectified linear unit (commonly shortened to ReLu) aims to apply an 'elementwise' activation function such as sigmoid to the output of the activation produced by the previous layer.

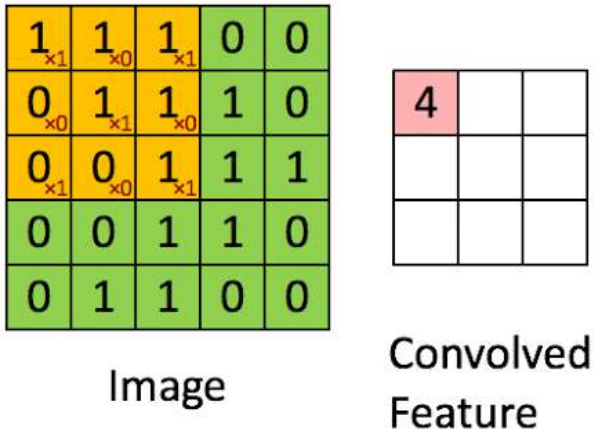


Fig.2 *Convolution with 3×3 Filter. Source: http://deeplearning.stanford.edu/wiki/index.php/Feature_extraction_using_convolution*

Iii. Pooling layers aim to gradually reduce the dimensionality of the representation, and thus further reduce the number of parameters and the computational complexity of the model.

The pooling layer operates over each activation map in the input, and scales its dimensionality using the "MAX" function. In most CNNs, In **Fig. 3** max-pooling layers with kernels of a dimensionality of 2 × 2 applied with a stride of 2 along the spatial dimensions of the input is applied. This scales the activation map down to 25% of the original size - whilst maintaining the depth volume to its standard size.

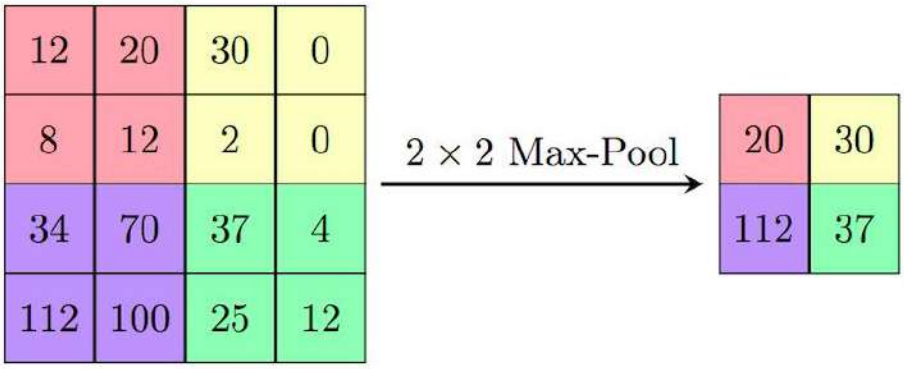
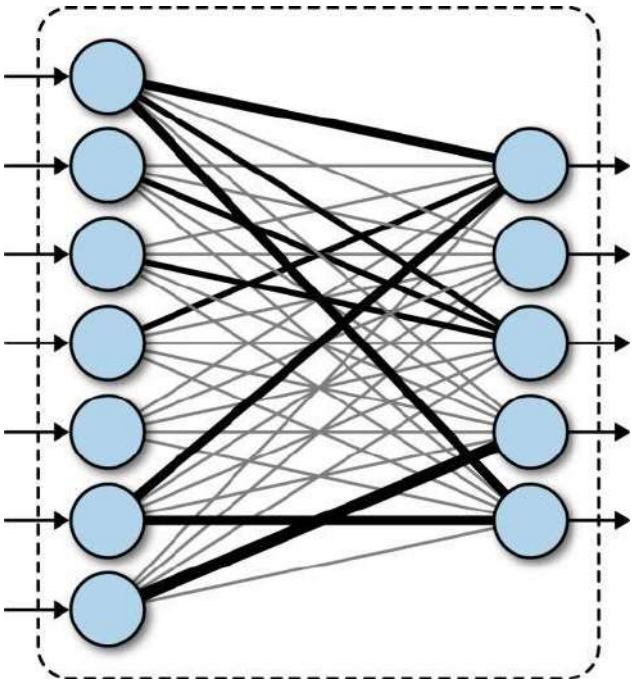


Fig.3 *Max-pooling with 2×2 Filter. Source: https://computersciencewiki.org/index.php/Max-pooling_/_Pooling*

iv. The fully-connected layers are actually the neural net that attempt to produce class scores from the activations, to be used for classification. It is also suggested that ReLu may be used between these layers, as to improve performance.

Fig.4 *Fully connected network (Neural Network). Source:*



https://www.researchgate.net/profile/Srikanth_Tammina

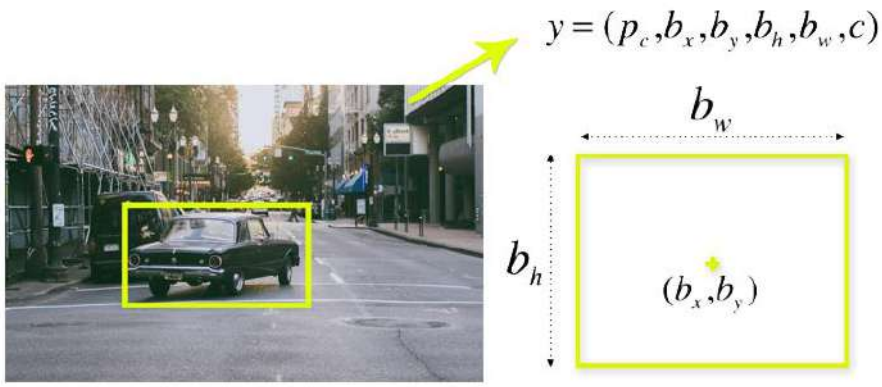
Apart from these basic layers there are other steps that can be incorporated such as Batch normalization, Regularization, Dropout, Convolution Transpose etc. Which is used according to model requirement.

5. YOLO Object detector

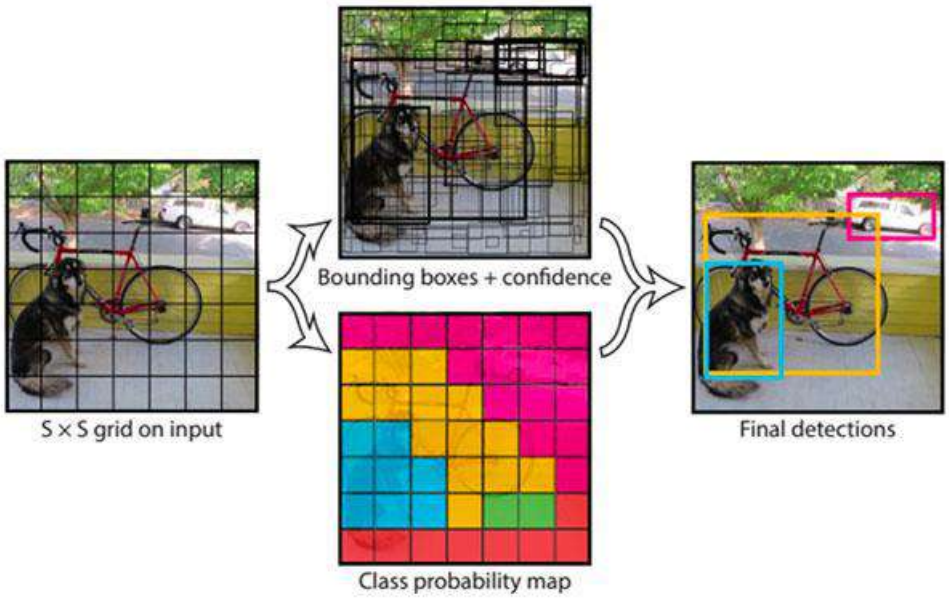
Redmon et al. [7] In 2015 came up with YOLO (You Only Look Once) network for object detection. The object detection task consists in determining the location on the image where certain objects are present, as well as classifying those objects. Previous methods for this, like R-CNN and its variations, used a pipeline to perform this task in multiple steps. This can be slow to run and also hard to optimize, because each individual component must be trained separately. YOLO, does it all with a single neural network.

The input image is divided into an $S \, x \, S$ grid of cells. For each object that is present on the image, one grid cell is said to be "responsible" for predicting it. That is the cell where the center of the object falls into.

Each grid cell predicts B bounding boxes as well as p_c class probabilities. The bounding box prediction has 6 components: $(b_x, b_y, b_h, b_w, confidence)$. The (b_x, b_y) coordinates represent the center of the box, relative to the grid cell location. These coordinates are normalized to fall between 0 and 1. The (b_w, b_h) box dimensions are also normalized to [0, 1], relative to the image size.



It is also necessary to predict the class probabilities, $P_r(Class(i) \mid Object)$. This probability is conditioned on the grid cell containing one object. In practice, it means that if no object is present on the grid cell, the loss function will not penalize it for a wrong class prediction, as we will see later. The network only predicts one set of class probabilities per cell, regardless of the number of boxes B. That makes $S \times S \times p_c$ class probabilities in total.
Adding the class predictions to the output vector, we get a $S \times S \times (B * 5 + p_c)$ tensor as output.



YOLOv3 is the latest iteration which is extremely fast and accurate. In mAP measured at 0.5 IOU(Intersection over union) YOLOv3 is on par with Focal Loss but about 4x faster. Moreover, you can easily tradeoff between speed and accuracy simply by changing the size of the model (either by using the complete model or the scaled down version).

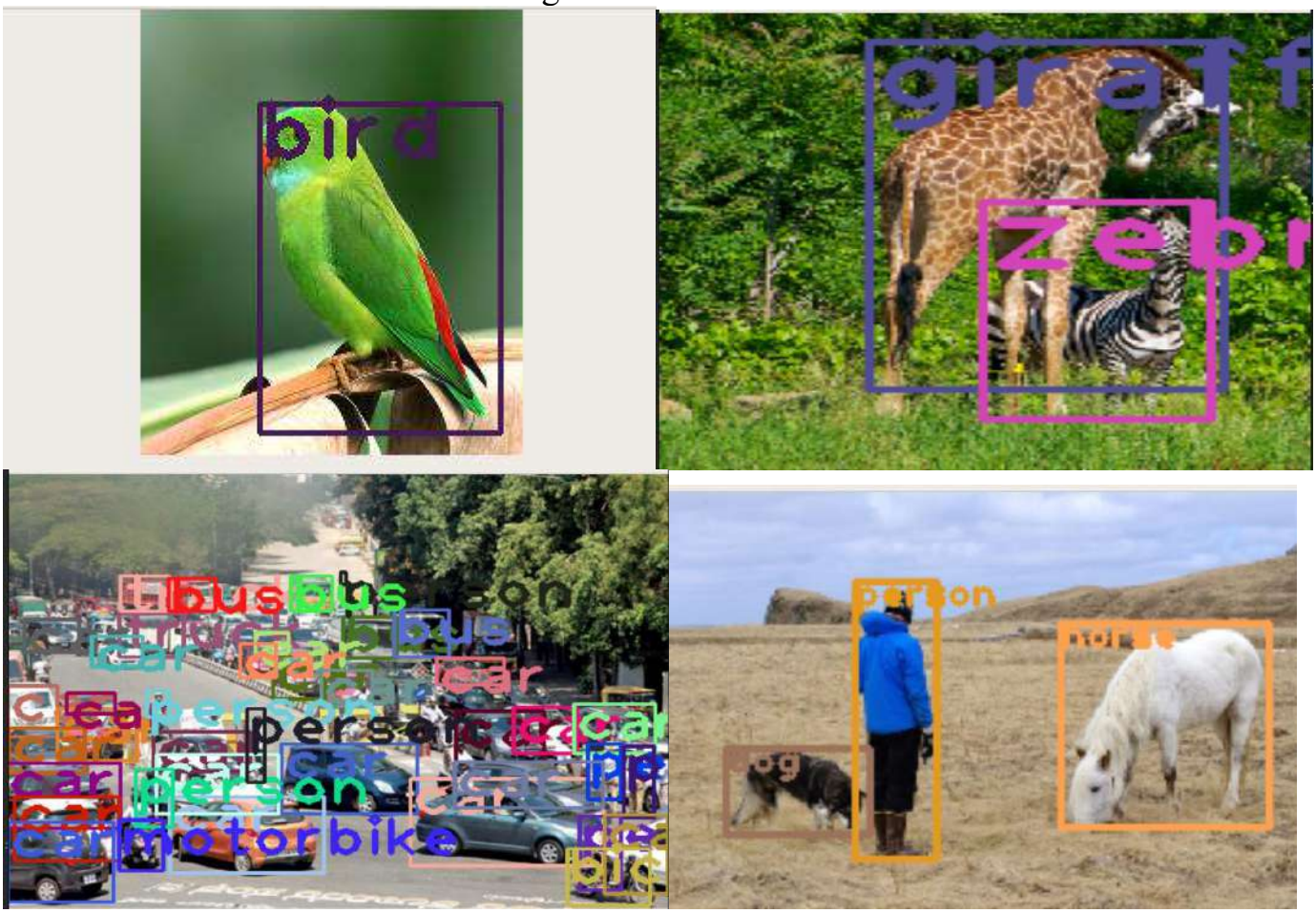
Other Fast multi-object detection algorithm that can be used

- Region Proposals (R-CNN, Fast R-CNN, Faster R-CNN.)
- SSD(Single Shot Detector)
- RetinaNet

- ResNet-34
- DarkNet-53
- etc

6. Results

The model is tested on real world images -



7. Conclusion

With specific hardware YOLO is perfect for self-driving application because of its high accuracy and fast algorithm.

It also makes predictions with a single network evaluation unlike systems like R-CNN which require thousands for a single image. This makes it extremely fast, more than 1000x faster than R-CNN and 100x faster than Fast R-CNN.

It is good for computer vision starter if pre-trained model is used because of abstraction most of the complexities are hidden.

OpenCV does not support GPU therefore the original weights cannot be used for real time object detection because of significant frame drop, the problem can be solved by using scaled down version of the weights.

Re-training the weights require powerful graphics cards.

8. References

[1] Felzenszwalb, Pedro & Girshick, Ross & Mcallester, David & Ramanan, Deva. (2010). Object Detection with Discriminatively Trained Part-Based Models. IEEE transactions on pattern analysis and machine intelligence. 32. 1627-45. 10.1109/TPAMI.2009.167.

[2] Leibe, Bastian & Leonardis, Ales & Schiele, Bernt. (2008). Robust Object Detection with Interleaved Categorization and Segmentation. International Journal of Computer Vision. 77. 259-289. 10.1007/s11263-007-0095-3.

[3] Pedraza, Anibal & Gallego, Jaime & Lopez, Samuel & Gonzalez, Lucia & Laurinavicius, Arvydas & Bueno, Gloria. (2017). Glomerulus Classification with Convolutional Neural Networks. 839-849. 10.1007/978-3-319-60964-5_73.

[4] Ren, Shaoqing & He, Kaiming & Girshick, Ross & Sun, Jian. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence. 39. 10.1109/TPAMI.2016.2577031.

[5] Ren, Shaoqing & He, Kaiming & Girshick, Ross & Sun, Jian. (2016). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. 1-10.

[6] Redmon, Joseph & Farhadi, Ali. (2016). YOLO9000: Better, Faster, Stronger.

[7] Redmon, Joseph & Farhadi, Ali. (2018). YOLOv3: An Incremental Improvement.