Object Detection using Deep Learning

Chamarty Anusha
Research Scholar
Department of Computer Science and Systems Engineering
Andhra University College of Engineering
Visakhapatnam

P. S. Avadhani
Principal
Department of Computer Science and Systems Engineering
Andhra University College of Engineering
Visakhapatnam

ABSTRACT

Autonomous vehicles, surveillance systems, face detection systems lead to the development of accurate object detection system [1]. These systems recognize, classify and localize every object in an image by drawing bounding boxes around the object [2]. These systems use existing classification models as backbone for Object Detection purpose. Object detection is the process of finding instances of real-world objects such as human faces, animals and vehicles etc., in pictures, images or in videos. An Object detection algorithm uses extracted features and learning techniques to recognize the objects in an image. In this paper, various Object Detection techniques have been studied and some of them are implemented. As a part of this paper, three algorithms for object detection in an image were implemented and their results were compared. The algorithms are “Object Detection using Deep Learning Framework by OpenCV”, “Object Detection using Tensorflow” and “Object Detection using Keras models”.

General Terms

Deep Learning, Machine Learning, Object Detection.

Keywords

Tensorflow, Keras, Opencv, bounding boxes.

1. INTRODUCTION

Object Detection is an art of identifying even the small parts of an image with more accuracy [2]. To accomplish this task, the existing Classification algorithms are being used and then a bounding box [3] is drawn around the object in an image.

Various models have been studied for detecting objects in an image like

- Classical Object Detection Frameworks using image classification
- Deep learning object detection
- Feature-based object detection

1.1 Classical Object detection Techniques

Some Classical algorithms for Object Detection are

1. Viola-Jones object detection
2. SVM classification with histograms of oriented gradients (HOG) features
3. Image segmentation [4] and blob analysis
4. Image segmentation using background subtraction Algorithms

1.1.1 Steps of Classical Object Detection Algorithms

The following are the various steps for these Algorithms:

1) A fixed size sliding window, which slides from left to right and top to bottom, is drawn to locate the objects in an image.
2) Next, an image pyramid is used to detect objects of varying scales.
3) Classification is done using a pre trained base-network like VGG net [5], ResNet etc.

At every stop of the sliding window and image pyramid, Regions of Interest (ROI) are extracted. These ROI's are given as input to the Convolution Neural Network (CNN) [2][6], which outcomes the class label 'L' of the object with some classification probability.

A threshold Classification probability 'P' is set in prior to the above step and it is compared with the above probability 'L'. If 'L' is greater than 'P', then mark the bounding box [3] of the ROI with the label 'L'. The above process is repeated for every stop of the sliding window and image pyramid.

Finally, a Non-Maxima suppression to be applied to the above resulted bounding boxes, for detecting the objects in the image.

This method of Object Detection is a better approach, as it uses a Pre-Trained Network, which avoids the need to train the network from beginning to end. This, in turn saves a lot of time necessary for training the Neural Network. This method has some drawbacks such as slower response and tedious process.

1.2 Deep Learning

Deep learning is a class of Artificial Neural Networks (ANN), having many processing layers. It is a Machine Learning Technique that learns features from the data.

As the amount of information is increasing, the performance of Deep learning algorithms surpassed the machine learning algorithms. “Geoffrey Hinton” in his paper [7] gave a breakthrough by successfully training the networks.

Deep Learning based Object Detection method has the following advantages:

1) Use of existing pre-trained classification network as the base network reduces the network training time.
2) Creation of a complete end-to-end deep learning-based object detector facilitates increased object detection accuracy.

The following steps are used for Object Detection using Deep Learning framework:

1) The required Object detection framework is to be determined from the existing frameworks like Faster R-CNN, SSD, YOLO etc.

2) Selection of the base network is very important because Base Networks are used for Classification. Some common Base Networks are

   - VGGNet – VGG16/19 [5]
   - ResNet
   - MobileNet
   - DenseNet
   - Inception

   These networks are pre-trained to perform classification on a large image dataset, such as 'ImageNet', 'GoogleNet' etc. Examples of different Object Detection frameworks with Base Networks:

   - Region-Based Fully Convolutional Networks (R-FCN) with Resnet
   - Faster RCNN with Resnet
   - Faster RCNN with Inception Resnet
   - Single Shot Multibox Detector (SSD) with MobileNets
   - SSD with Inception

1.3 Region Based Convolutional Neural Network

Region based **C**onvolutional **N**eural **N**etwork [4], uses the following steps

1) Initially, the input image is scanned using Selective Search algorithm which outputs around 2000 region proposals.

2) In the next step convolutional neural network is applied on each of these Region proposals.

3) In the last step the output of CNN is given to

   - Support Vector Machine (SVM) for Classification.
   - Linear regressor for drawing the bounding box [3] around the object.

Hence, R-CNN [5] is specifically used for Object Detection by using the Classification algorithms as a backbone.

1.4 Fast R-CNN

R-CNN's immediate descendant was Fast-R-CNN [5] [8]. Fast R-CNN is almost similar to R-CNN in many aspects but the speed of Fast R-CNN is much more than R-CNN in detecting objects in an image.

The standard steps followed in Fast R-CNN are

1) Initially, Feature Extraction is performed over the image before proposing Regions of Interest. Then one CNN is run over the entire image instead of 2000 CNN's over 2000 overlapping regions.

2) In the second step SVM is replaced by Softmax Layer, which helps in predicting the class of output object.

So, Fast R-CNN is a development over R-CNN in terms of speed but the only problem is with the usage of selective search algorithm.

1.5 Faster R-CNN

Faster R-CNN [5] has become a standard Deep Learning model for object detection. The framework of Faster R- CNN consists of

- Region Proposal Network (RPN)
- A collection of anchors (boxes)
- The Region of Interest (ROI) pooling module
- The Region-based Convolution Neural Network

The main objective of developing Region Proposal Network (RPN) is to replace the slow Selective search algorithm with a fast-neural network. At the last layer of a CNN, a 3x3 sliding window slides across the feature map and maps it to a lower dimension. At each sliding window location, multiple regions with anchor boxes are generated. Each region proposal consists of

1) Object availability score for that region.

2) Four coordinates representing the bounding box of the region.
3) Finally, last available Feature Map's every location is considered and 'n' different bounding boxes like; a tall box, a wide box, a large box etc. are drawn.
4) Now the availability of object in each of these boxes along with their coordinates is displayed.

The main advantage of using Faster R-CNN is region proposal generation and object detection. Both of them are done by the same convolution network and which can solve even the complex computer vision problems.

1.6 Region-based Fully Convolutional Network (R-FCN)

The Region-based Fully Convolutional Network (R-FCN) [5] is developed by J. Dai, et al. (2016). This model has convolution layers with back-propagation for training and prediction of class of the object. The Fast and Faster R-CNN methodologies use region proposals to identify objects in an image. The authors of R-FCN have merged the object detection (location invariant) and its position (location variant) steps for faster results. The best R-FCNs have reached Mean Average Precision (map) score of 83.6%. The authors noticed that the R-FCN is 2.5 to 20 times faster than the Faster R-CNN.

1.7 Single-Shot Detector (SSD)

W. Liu, et al. (2016) have developed a Single-Shot Detector (SSD) to predict the bounding boxes and the class probabilities with an end-to-end CNN architecture.

This Model takes the image as input and it is further given to a convolution neural network which has different number of layers. The output of these layers is a Feature Map, which helps in predicting the bounding boxes.

Each bounding box has four parameters like the centre co-ordinates, the width and the height. SSD also results the probability of each object, belonging to a particular class. The Non-Maximum Suppression method is then applied to detect bounding boxes which accurately enclose the objects in the image.

This model replaces the deeper layers in the base network architecture with new layers, which are SSD layers, and use the new models like Faster R-CNN to perform object detection accurately. This process of removing some base network layers and replacing them with newer layers is called

"Network Surgery". A Network Surgery is to be done very carefully as only the unnecessary layers have to be removed and to be replaced with necessary layers. Then, training is done by modifying the weights of both the new layers and Base network layers.

SSD models are trained with 2007, 2012 PASCAL VOC datasets and the 2015 COCO dataset. The output map score with 2007 PASCAL VOC dataset is 83.2% and with 2012 PASCAL VOC test dataset is 82.2%.

1.8 You Only Look Once (YOLO)

The YOLO model developed by J. Redmon in his paper [9], is used for predicting class probabilities and for drawing bounding boxes around the objects in a single evaluation. YOLO model is mainly used for real time predictions.

YOLO model takes an image which is an nxn grid as input. Every cell of the grid predicts B bounding boxes with a particular confidence score.

Confidence score is "the probability to detect an object multiplied by Intersection over union". Intersection over union is the ratio of predicted and ground truth boxes. This network has 24 Convolution layers followed by 2 fully connected layers and is pre-trained with ImageNet dataset.

Four convolution layers and two fully connected layers are added at the end, and the network is re-trained with the 2007 and 2012 PASCAL VOC datasets. The output of the final layer consists of a SxS(C+Bx5) tensor, which gives the number of predictions for each cell of the grid. Where,

S → Confidence value

C → Number of estimated probabilities for each class

B → fixed number of anchor boxes per cell, and each box consists of four co-ordinates which are centre of the box, width and height of box.

In the previous models, the probability of objects being present in the bounding boxes is more but, in YOLO, the number of bounding boxes is more.

Hence, the number of bounding boxes without objects is more. To remove some bounding boxes which are having almost the same co-ordinates, the Non-Maximum suppression method is used. This method merges highly-overlapping bounding boxes of the same object into a single box. YOLO model is trained with 2007 and 2012 PASCAL VOC datasets.

The output map score with 2007 PASCAL VOC dataset is 63.7% and with 2012 PASCAL VOC test dataset is 57.9%.

1.9 YOLO 9000

This model, developed by J. Redmon and A. Farhadi [9], has the capability to detect around 9000 object categories in real time.

1.10 YOLOv2

The YOLOv2 model is developed with the idea of improving accuracy over previous models.

Batch Normalization is added to this model to prevent over-fitting. Input to this model is an image of size 608 x 608 while input to YOLO model is 448 x 448. Increasing the size of the input helps in detection of potentially smaller objects.
YOLOv2 is trained on the ImageNet dataset and 2007 PASCAL VOC dataset. The output Mean Average Precision (map) score obtained with PASCAL VOC is 76.8%.

1.11 Neural Architecture Search Net (NASNet)

Neural Architecture Search was developed by B. Zoph and Q.V. Le in 2017. NASNet learns the architecture of the model to optimise the number of layers, which in turn helps in improving the accuracy over a given dataset. NASNet network architecture is learned from CIFAR-10 dataset and this model is used for feature maps generation. NASNet is stacked into Faster R-CNN and the entire pipeline is retrained with the COCO dataset. NASNet models are trained with test-dev dataset of the COCO challenge.

The best output map score with 2007 PASCAL VOC dataset is 43.1% and with 2012 PASCAL VOC test dataset is 82.2%.

The lighter version of the NASNet obtained a map score of 29.6% over the same dataset.

2. DEEP LEARNING MODELS

2.1 CAFFE Object Detection Model

Convolution Architecture For Feature Extraction (CAFFE), is a Deep Learning Framework [2][10] created by Yangqing Jia and by the Lead Developer Evan Shelhamer. CAFFE's fundamental unit of computation is implemented as the following layers:

- Data access
- Convolution
- Pooling
- Activation Functions
- Loss Functions
- Drop out

CAFFE learns model from scratch and then resume learning from the saved models. It then applies the technique called "Fine-Tuning" to detect new objects. CAFFE model trains the network by using "CAFFE train" command and it further requires solver configuration file called "solver.prototxt". This file contains information about training of the network. CAFFE model is a repository of trained models called ZOO, which is being used by researchers and machine learning practitioners to design their CNN. CAFFE model consists of "bvlc_reference_caffenet" as the base network and design the object detection framework using Transfer Learning Technique. Transfer Learning [11] is the art of developing a model for a task and reusing that model as a starting point for another task. CAFFE model was trained with ImageNet dataset which consists of millions of images across 1000 categories.

2.2 Tensorflow Object Detection

Tensorflow Object Detection [2][11] Application Program Interface (API) is used to build powerful image recognition software. It is an Open Source Framework built on top of Tensorflow. It consists of a collection of Detection Models, which are pre-trained on COCO dataset and open images dataset. One of the models among them is Single Shot Detectors and MobileNets. This model is very fast and requires less computational capability. The TensorFlow model's GitHub repository has a large number of pre-trained models for large number of Machine Learning Algorithms.

2.3 Keras Object Detection Model

Keras [2][11] is a high-level library, used for building Neural Network models. Keras was mainly developed for fast execution of ideas. It has a simple and highly modular interface, which makes it easy to create an even complicated Neural Network model. The main features of Keras are its Modularity and Extensibility. Modularity means building the modules of a neural network using a simple interface and Extensibility is writing a new module for Keras.

3. EXPERIMENTAL RESULTS

In this paper three Object Detection models are tested and compared. They are

- Using CAFFE model
- Using Tensorflow Model
- Using Keras Model

The same image is given to the above three models and the accuracy of detection of objects in that image are analysed.

3.1 Screen Shots

3.1.1 CAFFE Model's output

CAFFE model is used for detecting objects in an image. The confidence score with which it detected the object is shown in the below screen shot.

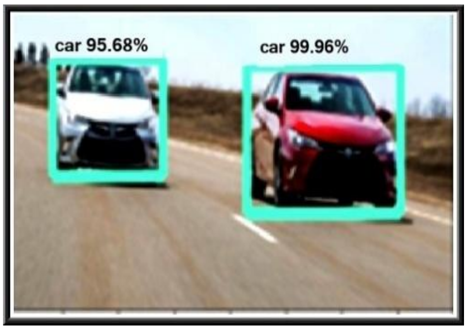
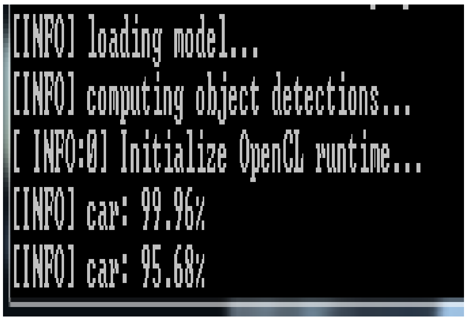


Figure 1



```
[INFO] loading model...
[INFO] computing object detections...
[ INFO:0] Initialize OpenCL runtime...
[INFO] car: 99.96%
[INFO] car: 95.68%
```

Figure 2

The above Figures 1 and 2, contains two objects and are identified as cars by CAFFE model with an accuracy of 95.68% and 99.96%.

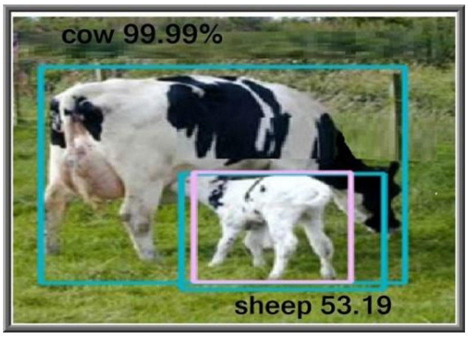
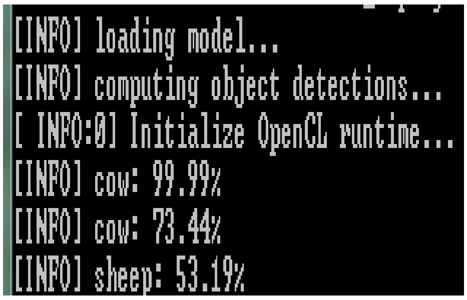


Figure 3



```
[INFO] loading model...
[INFO] computing object detections...
[ INFO:0] Initialize OpenCL runtime...
[INFO] cow: 99.99%
[INFO] cow: 73.44%
[INFO] sheep: 53.19%
```

Figure 4

Similarly, the Figures 3 and 4, contains two objects and are identified as cow & cow with a confidence scores of 99.99% and 73.44%. Sometimes, the calf is identified as sheep with a confidence score of 53.19%.

3.1.2 Tensorflow object Detection

Tensorflow model is used for detecting objects in an image. The confidence score with which it detected the object is shown in the Figure 5. Tensorflow model identified the cars with an accuracy of 85% and 99%.

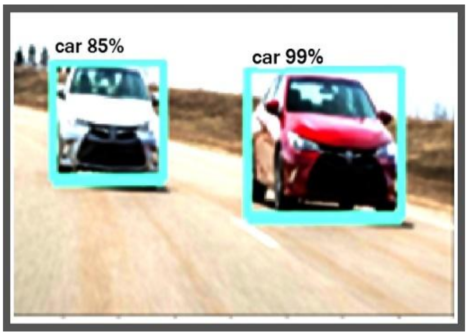


Figure 5

3.1.3 Keras Object Detection Model

Keras Object Detection model is used for detecting objects in an image. The confidence score with which it detected the object is shown in Figure 6 and 7.

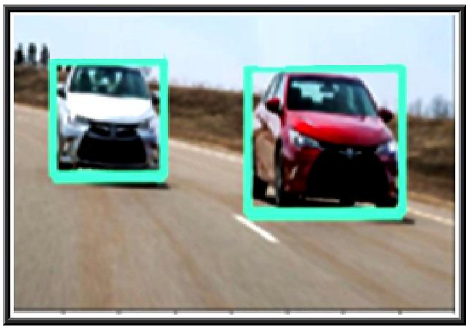


Figure 6

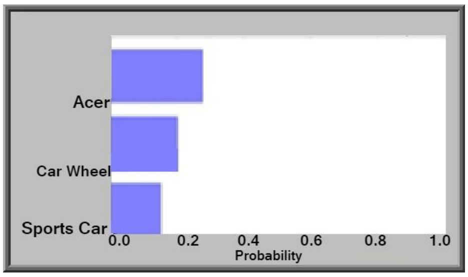


Figure 7

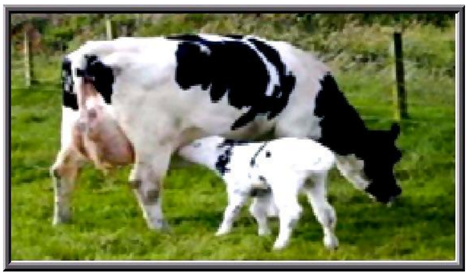


Figure 8

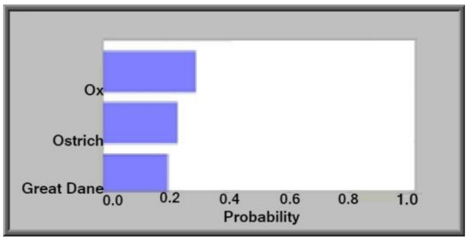


Figure 9

Keras model identified the cars with an accuracy of around 20% as per Figure 6 and 7 and cow is identified as OX with an accuracy of 30% as per Figure 8 and 9.

3.2 Result Analysis

Every model has different number of Convolution, Pooling, Activation functions and Fully Connected Layers in their architectures. As the number of layers is varying, the detection accuracy of model is also varying.

Table 1

Sno	Model Name	Base Network	No of Layers	Top 5 Error Rate
1.	Caffe Model	Alex Net[6], VGG Net-16/19	5 CNN layers	84.7%
2.	Tensor flow Model	Google Net[6], Inception v1/v2/v3	22 deep CNN layers	93.3%
3.	Keras Model	Inception v3, Imagenet	159 layers	94.4%

The architecture of the above three models is depicted in Table 1. From the above screenshots, it is clear that cars are detected by CAFFE model with greater accuracy than Tensorflow and Keras models. Similarly, the other models detect some other objects with greater accuracy than CAFEE model. Accuracy with which the object is detected depends on the internal architecture used in the model.

4. CONCLUSION

The Base Networks plays a key role in designing an accurate Object Detection Model. This is because each base network has different number of convolutions, pooling and fully connected layers [6]. For designing a better Object Detector, a better Convolution Network is to be designed by continuously varying the weights of various layers of CNN.

5. ACKNOWLEDGMENTS

This work is done under the scholarship of Visvesvaraya PhD Scheme for Electronics and IT.

6. REFERENCES

[1] Mohannad Elhamod, Martin D. Levine, Automated Real-Time Detection of Potentially Suspicious Behaviour in Public Transport Areas. In IEEE Transactions on Intelligent Transportation systems, vol. 14, no. 2, June 2013

[2] Ajeet Ram Pathak, Manjusha Pandey, Siddharth Rautaray, Application of Deep Learning for Object Detection. In International Conference on Computaional Intelligence and Data Science(ICCIDS) may 2018 pp.1-11.

[3] Christian Szegedy, Alexander Toshev, Dumitru Erhan, Deep Neural Networks for Object Detection. In International conference on neural information processing systems 2013.

[4] Xiaofeng Ning, Wen Zhu , Shifeng Chen, Recognition, Object Detection and Segmentation of White Background Photos Based on Deep Learning. In Youth Academic Annual Conference of Chinese Association of Automation (YAC) May 2017 pp. 1-6.

[5] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, Xindong Wu, Object Detection with Deep Learning: A Review. In Journal of Latex Class Files, Vol. 14, no. 8, March 2017

[6] Sakshi Indolia, Anil Kumar Goswani, S. P. Mishra, Pooja Asopa, Conceptual Understanding of Convolutional Neural Network- A Deep Learning Appraoch. In International Conference on Computational Intelligence and Data Sience(ICCIDS) 2018 pp.1-10.

[7] Yann LeCun, Yoshua Bengio , Geoffery Hinton, Deep Learning. In Review, vol. 521, May 2015.

[8] Ross Girshick, Fast R-CNN. In IEEE International Conference on Computer Vision December 2015 pp. 1-9.

[9] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, You Only Look Once: Unified, Real-Time Object Detection. In IEEE Conference on Computer Vision and Pattern Recognition june 2016 pp. 1-10.

[10] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, Trevor Darrell, Convolutional architecture for fast feature embedding. In ACM international conference on Multimedia Nov 2014 pp. 1-4.

[11] Maarten C. Kruithof, Henri Bouma, Noelle M. Fischer Klamer Schutte, Object recognition using deep convolutional neural networks with complete transfer and partial frozen layers. In SPIE Security+ Defense Conference Oct 2016 pp. 1-8.