Object detection using Convolutional Neural Networks in Tensor Flow

Arun Samkaria
Student, Baba Banda Singh Bahadur Engineering College, Fatehgarh Sahib, Punjab, India

Abstract- **With the development in the field of computational machines, computer vision is one of the rapidly growing domains in the most recent decade. Advance neural networks especially deep learning has played significant role in this. The applications of object detection ranges from video surveillance to self-driving cars makes this one of the most researched topics in recent times. In this paper we are more concern about object detection using image processing with the help of Tensorflow and compare results of the two most widely used algorithms i.e. SSD (Single short detector) and Faster R-CNN (Faster region based convolutional neural networks).**

Index Terms- Object detection, Computer vision, TensorFlow, Python, SSD, Faster R-CNN, OpenCV, cuDNN, Database file.

I. INTRODUCTION

TensorFlow was developed by the researchers at google as single infrastructure for Machine learning. Later, in 2015 it was open sourced under the Apache 2.0 license. The core of the TensorFlow is implemented in C++ but main language is python.

A TensorFlow computation is described by a directed graph, which is composed of a set of nodes. The graph represents a dataflow computation, with extension for allowing nodes to maintain and update the persistent state and for branching and looping control structure with in the graph. In TensorFlow graph, nodes represent the instantiation of an operation and edges which represents the tensors (multidimensional arrays). TensorFlow uses session for graph computation. The computational graph at the beginning is empty and there is no variable in it [1].

TensorFlow creates queues of nodes without dependencies to determine order of computation of a graph. The program executes the queue in some order to ensure the decreasing unresolved dependencies until whole graph is computed. The TensorFlow can be in a CPU (central processing unit) or in a GPU (Graphical processing unit) enabled systems. The performance of GPU is better than that of CPU [2]. The support on GPU exist for specific Nvidia cards using related version of CUDA-CNN toolkit [3]. In this paper performance and accuracy of two CNN models i.e. SSD and Faster R-CNN is compared. Both models are trained on 64-bit Nvidia GPU.

Hardware used for training the models is:

- 64-bit windows run system
- 8GB RAM (Random access memory)
- 2GB GPU RAM

II. PROGRAMMING MODELS AND BASIC CONCEPTS

Object detection in TensorFlow can be done by using various models and is a procedure of determining the instance of the class to which the object belongs and estimating the location of the object by outputting the bounding box around the it with accuracy or confidence reading [4]. Single class and multiple class object detection can be done in an image. Deep CNNs have been extensively used for object detection. CNN is a type of feed forward neural networks and works on the principle of weight sharing. Image is convolved with an activation function to get feature maps. To reduce spatial complexity of the network feature maps are treated with pooling layer to get abstracted feature maps. This process is repeated for desired number of filters and accordingly feature maps are extracted. Eventually, these feature maps are processed with fully connected layers to get output of images recognition showing confidence score for the predicted class label [5].

CNN architecture:

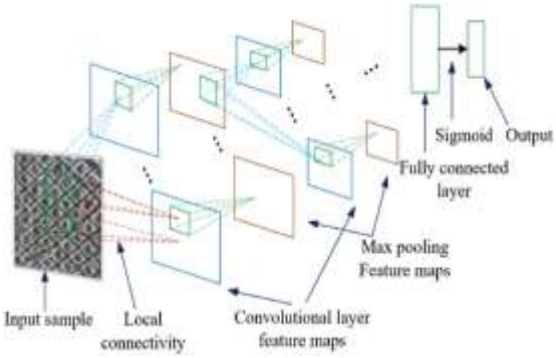

Figure 1: CNN architecture layers [6]

Single Short Detector (SSD):

SSDs are designed for object detection in real time. SSD eliminates the region of proposal to increase the overall speed of the entire process. SSD applies default boxes and multi-scale feature techniques to recover the drop-in accuracy. In SSD each prediction is composes a boundary box and 21 scores for each class. It computes both the location and class scores using small convolution filters. It complies 3 x 3 convolutional filter to each cell after extracting feature maps from an image and class with the highest score contain the object in the bounded box. SSD uses lower resolution layers (e.g. 4 x 4 feature map) to detect larger scale objects. It reserves class zero to indicate it contain no object [7].

SSD uses 59 frames per second in real time to detect an object from images, whereas Faster R-CNN uses 7 frames per second to detect an object in real time.

Faster-Region based convolutional neural network (Faster R-CNN): The architecture of Faster R-CNN is complex. The input images are represented as multidimensional arrays (tensors), when they passed through pre-trained CNN provides the convolutional feature map. This technique is commonly used in the context of transfer learning, especially for training a classifier on a small on a small dataset using the weights of a network trained on a bigger dataset []. It uses anchors, fixed sized reference bounding boxes throughout an image uniformly. Once the features are extracted by CNN in bounding boxes with relevant object ROI (Region of interest) pooling is applied and extract those features in new tensor. Finally, the R-CNN module classify the content in bounding boxes (or discard it) and adjust the bounding box coordinates. This is generally how Faster R-CNN works [8].

III. RESULT AND COMPARISON

Both the models were trained using same dataset in TensorFlow. Significant differences were noted down during the training time. The overall loss function in both cases were checked with the help of Tensor Board graph outputs. In case of the Faster R-CNN the training was more linear and the training graph for loss function shows this and it achieves required loss (0.015) in 3 times faster than the time taken by SSD model for training. Not only the training time is less in case of Faster R-CNN, but the loss function also doesn't vary to much once it comes under 1. The variation in loss function values in case of SSD function is very high and same is visible in the graph also.

Table 1 below shows the difference in training time and loss function of both the models respectively:

Model	No. of steps for training	Time taken	Loss function
SSD	More than 53,000	10 to 11 hrs	0.5 to 1
Faster R-CNN	6000 to 7000	2 to 2.5 hrs	0.01 to 0.15

Table 1: comparison of training time and loss function for SSD and Faster R-CNN.

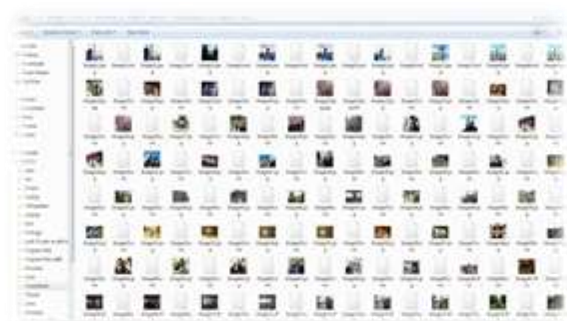

Figure 2: Dataset used for training models

Figure 3: Loss functioan grph for SSD model

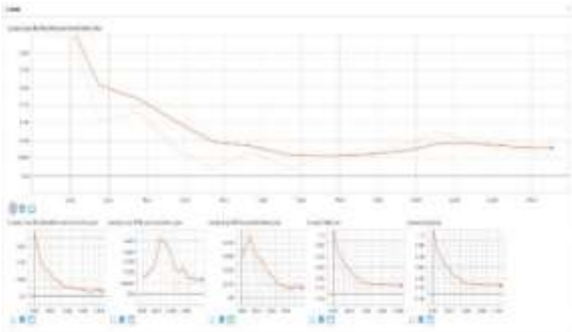

Figure 4: Loss function graph for Faster R-CNN

Although, SSD model consume more time for training but it is faster than that of Faster R-CNN but are less accurate. I have trained these models on only one class. When presented with single object in an image as input to model, both the models detected the object in the image with almost similar accuracy. But when presented with an input image with different objects in an image along with the trained object, SSD model was not able to recognize the object while Faster R-CNN was able to predict the questioned object with 99% confidence score and a bounded box around the class with its name.

IV. CONCLUSION

Both models are suitable for their purpose, if the major requirement is accuracy and is having larger datasets, Faster R-CNN is the one. But for real time object detection with 50 to 60 frames per second, SSD model can we used, and training should be done on little blurry images with small dimensions (300 x 300). Overall training of Faster R-CNN is less tedious process with less training time and greater accuracy. If all the steps are followed properly and all boxes ticked, there is always a scope of improvement in every project.

REFERENCES

[1] Sudharshan Duth P, Swathi Raj – Object recognition in images using convolutional neural network. IEEE Xplore document no. 8398912.

[2] Damir Demirovic, Emir Skejic, Amira Serifovic-Trbalic – Performance of some image processing algorithms in Tensorflow. IEEE Xplore document no. 7993784

[3] CUDA toolkit: http://www.nvidia.com

[4] TensorFlow: http://www.Tensorflow.org

[5] Ajeet Ram Pathak, Manjusha Pandey, Siddharth Rautaray: Application of deep learning for object detection, International Conference of Computational Intelligence and data Science (ICCIDS 2018): http://www.sciencedirect.com

[6] CNN architecture: http://www.researchgate.net

[7] Jonathan hui: SSD object detection single shot multibox detector for real time processing http://www.medium.com

[8] Faster R-CNN on modern object detection: http://www.tryolabs.com