

Object Recognition in Live Camera Feed

Rajarshi Sahu¹, Mannat Yadav², Dr. Pankaj Agrawal³

^{1,2} Student IMS Engineering College, Ghaziabad, India, ³Associate Professor, IMSEC,

Abstract: Object recognition can be done by many techniques; in this article we present an object recognition approach using modern computer vision technology. Due to the application fields and emphasis may be different, the number of features which we can select is large. This paper introduces some common object classification method by keeping balance the accuracy and time.

I- INTRODUCTION

Object detection is a computer vision technique for analyze in images and in videos, object recognition is the output of deep learning and machine learning algorithms. When we look at an image or watch a video, we can easily spot people, objects, scenes, and visual details. The goal is to teach the computer to do what comes naturally to humans that is to gain a level of understanding of what an image contains. Every object class has its own special feature that helps in classifying the class of the object, for an instance all spheres are round. Object class detection uses these special features. For example, when looking for spheres, objects that are at a particular distance from a point are desired.

Object recognition uses Convolutional Neural Networks for recognition. A convolutional Neural Network or CNN is a part of deep neural network. It consists of various multilayer perceptron designed to require minimum processing. A convolutional neural network consists of an input and an output layer, as well as many hidden layers. The hidden layers of a CNN consist of convolutional layers, RELU layer i.e. activation function, pooling layers, fully connected layers and normalization layers.

II- METHODOLOGY

This project uses Single Shot Detection (SSD) method for classification of the images. The other models such as Regional Convolutional Neural Network (R-CNN),

Fast R-CNN, Faster R-CNN, You Only Look Once (YOLO).

SSD method is a Regression based Object Detection. Object localization and classification are done in a single forward pass of the neural network. It is based on a feed forward convolutional neural network which produces a fixed size collection of Bounding Boxes. Bounding box is a rectangle on the image which tightly fits the object in the image.

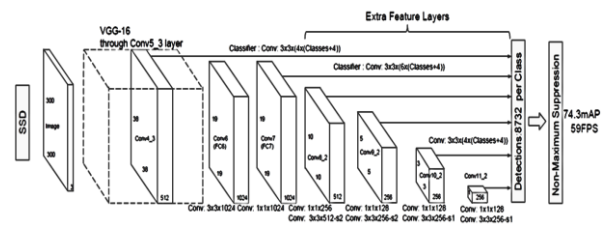


Figure (1): architectural diagram of Single shot Detection model

The architectural diagram of SSD model is shown in Figure (1). The initial network layers are based on a standard architecture used for high quality image classification, which is known as Base Network. We add convolutional feature layers at the end of the base network. These layers decrease in size and allow predictions at multiple scales. Each added feature layer can produce a fixed set of predictions on the input image using a set of convolutional filters. These are indicated on top of the SSD network architecture. The bounding box offset output values are evaluated relative to a default box position relative to every feature map location.

A set of default bounding boxes associated with each feature map cell, for multiple feature maps. The default boxes tile the feature map in an organized manner, so that the position of every box related to its corresponding cell is fixed. At each feature map cell, we predict the offsets relative to the default box shapes in the cell, as

well as the per-label scores that indicate the existence of a class instance in each of those boxes.

III- IMPLEMENTATION

There are two phases of the project. These can be divided as:

3.1. Training of the neural network.

Training of the neural network begins with labelling the images. Each image is labelled by creating manual rectangular boxes around the objects and assigning the class of the object that it belongs to. For each image a XML file is generated which will be used to generate Training data. Each XML document contains the size and position of the boxes.

On the basis of training data the neural network is trained.

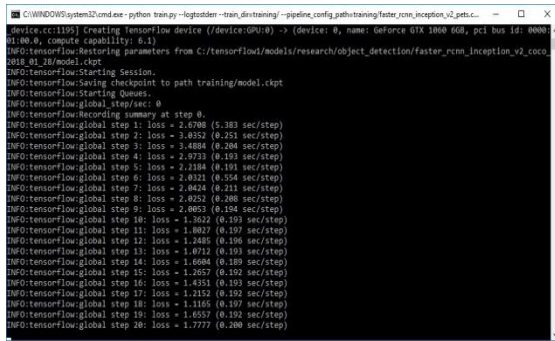


Figure 2: Training the neural network

Loss graph shows the overall loss of the classifier over the time. Figure (3) shows the loss graph in the tensorboard. The training is stopped after 40,000 iterations.

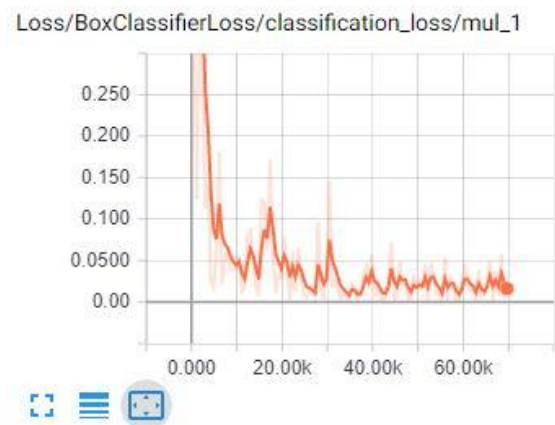


Figure (3): Loss graph

3.2 Using trained model for object recognition in image and video

To test the model, a sample image of the object is feed to the trained neural network. The model predicts the class and the dimensions and positions of the bounding box. The image is plot with the bounding box.

For detection in video, open CV captures the video in the form of frames. Each frame is passed to the trained neural network and the output with the predicted bounding box is plot. The neural network takes 0.20 to 0.25 seconds to process a single frame.

Figure 3.The GUI of the application contains two options, i.e. for object recognition in image and the object recognition in video.

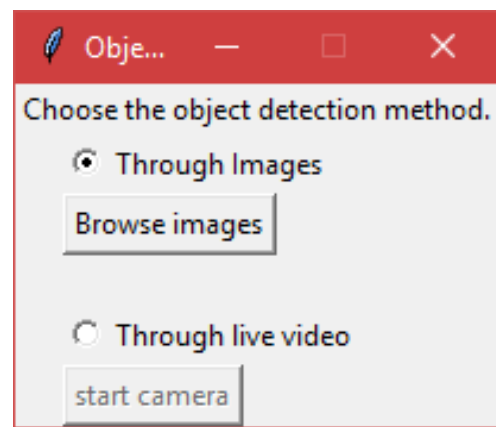
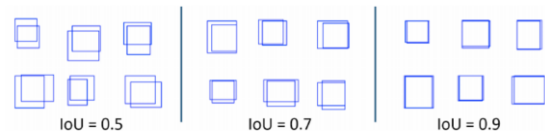


Figure 3- GUI

This application will recognize the images present in the given directory. Each individual image is taken and loaded into the numpy arrays, these numpy arrays goes through a process of feature extraction and those features are given to the trained neural network. The neural network gives out the each class score and the dimensions of Bounding box also known as Jaccard distance.



The class with most scores will be the final predicted result. The bounding box is plot in the image with the class associated to it.

IV- RESULT

Following images are tested against the trained model. Figure 3. The outputs of the images are shown in figure 4.



This is an apple.

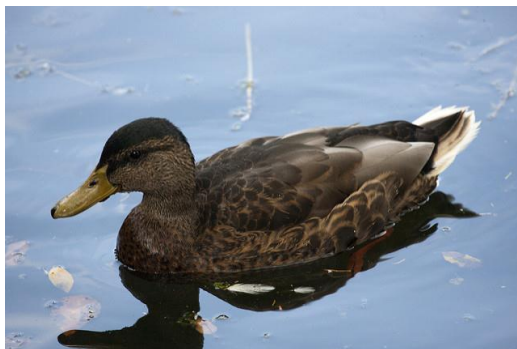
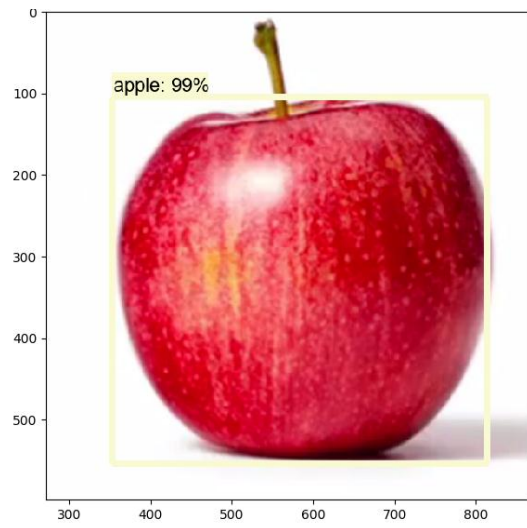


Figure 3: input images



Figure 4: output images

REFERENCES

- [1] <https://towardsdatascience.com/understanding-ssd-multibox-real-time-object-detection-in-deep-learning-495ef744fab>
- [2] <https://cv-tricks.com/object-detection/single-shot-multibox-detector-ssd/>
- [3] <https://www.cs.unc.edu/~wliu/papers/ssd.pdf>
- [4] https://www.tensorflow.org/guide/using_gpu
- [5] <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- [6] <https://medium.com/@smallfishbigsea/a-walk-through-of-alexnet-6cbd137a5637>
- [7] <https://docs.python.org/2/library/tkinter.html>
- [8] https://opencv-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_tutorials.html
- [9] <https://www.mathworks.com/discovery/object-detection.html>
- [10] <https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/>
- [11] <https://towardsdatascience.com/evolution-of-object-detection-and-localization-algorithms-e241021d8bad>
- [12] <https://www.frontiersin.org/articles/154269>