Visual Object Detection System

Srivenkata Krishnan Sesharamanujam
svkprasanna1995@gmail.com

1 Introduction

How much time have you spent in looking for the misplaced wallet in a messy environment ? It happens to the almost all of us and is an frustrating experience. But what if a simple computer algorithm could locate the wallet in a matter of milliseconds ? That is capability of object detection algorithms. The above example was a simple scenario. The applications of object detection range from round-the-clock surveillance systems to real-time detection in autonomous cars. In simple definition, Object detection is the process of identifying and locating the objects in an image. The task of the Visual Object Detection project is to find a location of a phone dropped on the floor from a single RGB camera image. The rest of the report is structured as follows: section 2 explains the state of art object detection algorithms and which algorithm will work better for our task. Section 3 explains how the visual object detection has been implemented and finally conclude on how we can improve the accuracy of the system.

2 Object Detection Algorithms

Object detection is one of the difficult problems in computer vision. In many aspects it is similar to other computer vision tasks, because it involves creating a solution that is invariant to deformation and changes in lighting and viewpoint. What makes object detection a challenging problem is that it involves both localisation and classification of regions in an image.

To detect an object, we need to have some idea where the object might be and how the image is segmented. This creates a type of chicken-and-egg problem, where, to recognize the shape and class of an object, we need to know its location, and to recognize the location of an object, we need to know its shape. Some visually dissimilar features, such as the clothes and face of a human being, may be parts of the same object, but it is difficult to know this without recognizing the object first. On the other hand, some objects stand out only slightly from the background, requiring separation before recognition.

Over the last years, there has been an drastic increase in accuracy of object detection algorithms due to the recent advancements in deep learning based computer vision models. Besides significant performance improvements, these techniques are reducing the need of extremely large data sets. In addition, with the performance of these systems are much higher, it is being used in the real time scenarios such as the autonomous cars. The following sections discuss about the current state of art in this realm.

2.1 Region-Based Convolutional Neural Network (R-CNN)

As an alternate to the exhaustive search in an image to capture object detection, the selective search method is implemented in the initial R-CNN. Small regions in an image are initialized and merged using hierarchical grouping. Now, the final group is a box containing the entire image. Then, the detected regions are merged according to a variety of color spaces and similarity metrics. The output is obtained by merging small regions which could contain the object.

The enhanced R-CNN model combines the selective search method and deep learning to find the object in these regions. A 4096- dimension vector is extracted from the input of a CNN. Each region

proposal is resized to match the input of the neural network. The features vector is then fed into multiple classifiers to produce probabilities that belong to each class. Each one of these classes has a SVM classifier trained to infer a probability to detect this object for a given vector of features. This vector also feeds a linear regressor to adapt the shapes of the bounding box for a region proposal and thus reducing the localization errors. The model produced considerable results on readily available data sets such as PASCAL VOC 2012.

The disadvantages of this approach are: First, Training is a multi stage pipeline. Second, Training is expensive in space and time. Third, object detection is slow.

2.2 Fast Region Based Convolutional Neural Network (Fast RCNN)

A Fast R-CNN network takes as input an entire image and a set of object proposals. The network first processes the whole image with several convolutional and max pooling layers to produce a convolutional feature map. Then, Region of Interests (RoIs) are detected with the selective search method applied on the produced feature maps. Each feature vector is fed into a sequence of fully connected layers that finally branch into two output layers: one that produces softmax probability estimates over K object classes and another layer that outputs four real-valued numbers for each of the K object classes. Each set of 4 values encodes refined bounding-box positions for one of the K classes.

The main disadvantage of FCNN was that it is computationally expensive. Therefore, a new network called Faster RCNN was developed to nullify the disadvantages posed by FCNN.

2.3 Faster Region Based Convolutional Neural Network (Faster RCNN)

The Faster RCNN is composed of two modules. The first is a deep convolutional network that proposes regions, and the second module is the Fast RCNN detector that uses the proposed regions. These two modules combine to form the object detection system. The Faster Region-based Convolutional Network (Faster R-CNN) is a combination between the Region Proposal Network (RPN) and the Fast R-CNN model. Faster R-CNN uses RPN to avoid the selective search method, it accelerates the training and testing processes, and improve the performances.

2.4 Mask Region Based Convolutional Network (Mask RCNN)

As an extension of the Faster R-CNN model, Mask RCNN adds a parallel branch to the bounding box detection in order to predict object mask. The mask of an object is its segmentation by pixel in an image.

The Mask Region-based Convolutional Network (Mask R-CNN) uses the Faster R-CNN pipeline with three output branches for each candidate object: a class label, a bounding box offset and the object mask. It uses Region Proposal Network (RPN) to generate bounding box proposals and produces the three outputs at the same time for each Region of Interest (RoI).

2.5 Region Based Fully Convolution Network (RFCN)

The Region-based Fully Convolutional Network (R-FCN) is a model with only convolutional layers allowing complete backpropagation for training and inference.The last layer in the R-FCN outputs feature maps, each one is specialized in the detection of a category at some location. For example, one feature map is specialized in the detection of a cat, another one in a banana and so on. Such feature maps are called position-sensitive score maps because they take into account the spatial localization of a particular object.

2.6 Single Shot Detector (SSD)

The model takes an image as input which passes through multiple convolutional layers with different sizes of filter (10x10, 5x5 and 3x3). Feature maps from convolutional layers at different position of the network are used to predict the bounding boxes.

They are processed by a specific convolutional layers with 3x3 filters called extra feature layers to produce a set of bounding boxes similar to the anchor boxes of the Fast R-CNN. Each box has 4 parameters: the coordinates of the center, the width and the height. At the same time, it produces a vector of probabilities corresponding to the confidence over each class of object.

The Non-Maximum Suppression method is also used at the end of the SSD model to keep the most relevant bounding boxes. The Hard Negative Mining (HNM) is then used because a lot of negative boxes are still predicted. It consists in selecting only a sub part of these boxes during the training. The boxes are ordered by confidence and the top is selected depending on the ratio between the negative and the positive which is at most 1/3.

2.7 Neural Architecture Search Net (NASNet)

NASNet consists in learning the architecture of a model to optimize the number of layers while improving the accuracy over a given data set. For a given range of operations and hyperparameters, multiple sequences are realized to maximize the accuracy as a signal reward for a given data set. The objective is to learn the best sequence of operations (given a maximal depth) to get an optimized architecture.

2.8 You Look Only Once (YOLO)

The main advantage of YOLO is that it avoids computationally expensive region proposal steps that detectors like Fast R-CNN and Faster-RCNN require. YOLO uses grid cells as anchors for its detections.

YOLO reframes object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. YOLO divides the input image into an S S grid. If the center of an object falls into a grid cell, that grid cell is responsible for detecting that object.

YOLOs loss function simultaneously solves the object detection and object classification tasks. This function simultaneously penalizes incorrect object detections as well as considers what the best possible classification would be.

3 Implementation of the Visual Detection System

After Research of the above object detection algorithms described in section 2, YOLO seemed to be the best option for the visual detection system. The reasons for choosing YOLO are: First, the speed of the object system is high. Since we frame detection as a regression problem we dont need a complex pipeline. Second, Network understands generalized object representation as YOLO is highly generalizable it is less likely to break down when applied to new domains or unexpected inputs. Third, YOLO makes less than half the number of background errors when compared to other models. Since, we know that the background will be office floor. So, the no of background error has to be less.

The step by step implementation process of the visual detection system is as follows.

3.1 Preparing the Dataset

A dataset which we have consists of approximately 100 jpeg images of the floor from the factory building with a phone on it. Another file has the normalized coordinates of the phone with respect to the phone position in the image.

3.1.1 Bounding Box from the center coordinates of the phone

We know that the phone model is same in all the images. So, the size of the phone is same in all images. Only the orientation changes.

On Visualizing, we can say that the maximum area is covered by the bounding box when the phone is oriented diagonally. The height of the bounding box is minimum when oriented horizontally. The width of the bounding box is minimum when oriented vertically. With these information, we

calculate the average height and width of the bounding box. Now, we can calculate the bounding box coordinates which we input to the YOLO model.

3.2 Build the Model

3.2.1 Transfer Learning of YOLO

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task.

It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems.

In our visual detection system, we use the weights from the already trained YOLO model thus saving significant amount of time. Thus , using transfer learning to good effect.

3.3 Training the model

The YOLO model has been extended to Tensor flow implementation. We use that to our advantage and train the model by passing the necessary files and weights file of pre trained YOLO model.

3.4 Phone Detection

Once the model is trained. We send the images to be detected to the model. The position of the phone is printed as output.

4 Conclusion

As an advanced version of the visual detection system these enhancements can be made to improve the performance, speed and accuracy.

- Increasing the size of the dataset using data augmentation techniques.
- Bounding boxes drawn for each image according to the size of the phone. (Ideally, the corners of the phone should be touching the bounding box)