Dense Object Detection in Indoor Scenes Using Depth Information

Jinchao Ye
Stanford University
Stanford CA 94305
jcye@stanford.edu

Second Author
Institution2
First line of institution2 address
http://www.author.org/_second

Abstract

RGB-D sensors, such as Microsoft Kinect sensor, can provide us depth information which might be useful to do dense object detection. Moreover, internet images, such as images from ImageNet, have millions photos for various kinds of objects. Internet images are useful for training while the depth image can provide valuable information for detection. Using models trained from the internet images and a depth image acquired by Kinect, we want to do dense object detection on indoor scenes.

1. Introduction

1.1. Dense Object Detection

Object detection is a fundamental problem in computer vision. Recent years, Bag of Words Model and Part Based Model [1] have showed significant improvement in object classification or detection. However, these methods can only deal well with the scenario when there is only one kind or only a small number of kinds of objects in the image.

In natural indoor images, there are various kinds of objects, such as monitor, desk, keyboard, chair, books etc. Our goal is to detect almost every object in indoor images. The ideal result is that for every pixel in the foreground in the image, it belongs to a bounding box which is detected and labeled by our algorithm. The background means floor, wall and ceiling, which we usually don't care about. The ideal result is shown in figure 1. This is a very difficult task due to occlusion, low resolution, intra-class variance, etc.

1.2. Images from ImageNet

ImageNet is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. This is an excellent image dataset for training.

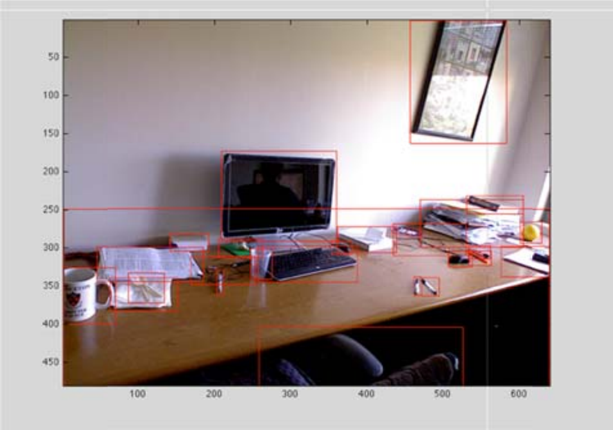

Figure 1 Ideal Result of Dense Object Detection

1.3. Depth Images from Kinect

Kinect sensor uses stereo techniques and can provide a depth image of the same scene. The effective depth sensor range of Kinect is between 0.8m and 3.5m and some pixels lack depth information. We need to do interpolation to get the useful depth image.

2. Problem Statement

Our goal is to do dense object detection on indoor images. Moreover, during the training process, we use only the ordinary 2D images from ImageNet. During detecting process, apart from the test image, we also have an additional depth image to help improve the detection performance.

2.1. Dataset

During training process, for each kind of object, we use the images from the corresponding subset in ImageNet. There are about 500 hundreds images per node in ImageNet so we can use the 500 hundreds images to train a classifier for each object.

For testing, we have collected 200 indoor images along with corresponding depth images. I have also labeled all the ground truth bounding boxes. There are 82 different kinds of objects as well as some undefined objects in the testing images. Some objects appear a lot of times in the testing images, such as monitor, desk, sink, faucet, etc. Some objects such as tennis appear only once or twice. There are also some kinds of objects which do not have corresponding subsets in ImageNet, such as chopping board. We neglect such kinds of objects during training and testing.

2.2. Expected Result and Evaluation

We expect our algorithm can achieve good results on the testing images. The ideal result is shown as in figure 1. Our algorithm might not detect every object but it should detect as many objects as possible. For each kinds of object, we can get a detection rate and false alarm rate. We can also calculate the overall detection rate and false alarm rate. We will run part based models [1] on our dataset and compare the result with the result of our own algorithm.

3. Technical Approach

3.1. Baseline Approach

We detect each object independently using sliding windows at different scales. The detector we used can be the one using part based models [1].

3.2. Overall Detection via Energy Maximization

For each kind of object, we first train a simple detector using HOG and logistic regression. We use a low threshold to propose candidates (O_i, s_i, x_i, y_i). O_i denotes the kind of object. s_i denotes the scale. x_i, y_i denote the position. i denote the candidate. For two candidates, we model the conditional probability as following:

$$P([o_j, s_j, x_j, y_j] | [o_i, s_i, x_i, y_i]) \\ = P(o_j | o_i) P(o_j, s_j | o_i, s_i) P(o_j, x_j | o_i, x_i) P(o_j, y_j | o_i, y_i)$$

where $P(o_j | o_i)$ measures the relevance of two kinds of objects.

$$P(o_j, s_j | o_i, s_i) = \frac{\left(s_j - s_i - \mu_{o_i o_j}\right)^2}{\sigma_{o_i o_j}^2}$$

is a Gaussian distribution. $P(o_j, x_j | o_i, x_i)$ and $P(o_j, y_j | o_i, y_i)$ can also be modeled by Gaussian distribution.

Then we can construct a graph model. Each node $N(i) = (o_i, s_i, x_i, y_i, d_i)$ represent a candidate proposed by the simple detector. The value for each node $N(i)$ is the probability from HOG and logistic regression. The value for each edge $E(i,j)$ is the probability above, i.e. $E(i,j) = P([o_j, s_j, x_j, y_j] | [o_i, s_i, x_i, y_i])$. Then the total energy is defined as:

$$Energy(flag) = \sum_i N(i)\, flag(i) \\ + \sum_{i \neq j} E(i,j)\, flag(i) flag(j)\}$$

It is a labeling problem. $flag$ is a labeling function which labels whether a candidate is true (value = 1) or false (value = 0).

This energy minimization approach takes relationship between objects into considerations. The relationship includes the co-occurrence between two kinds of objects and the relationship between their scales and their positions. However, it might be difficult to implement. Therefore, it might be easy to implement if we limit the relationship between a certain highly-related kinds of objects, such as monitor and keyboard, desk and chair, cooker and pan, etc.

3.3. Posterior Handle using Depth Image

Because we do not have depth images during training process, we cannot incorporate the depth information during training. However, we can use depth images to remove the false alarms by a certain detection method.

The first criterion is that the depth of an object is a continuous function of pixels. Therefore for each bounding box candidate, we compute the pixel histogram of depth. If there is a significant gap in the histogram, then the bounding box candidate should be removed.

The second criterion is using prior information. There is some common sense about the layout of certain objects. For example, from popular view points, keyboard is always in front of a monitor. We can use the depth image to check this criterion.

I am still thinking about how to incorporate depth image in the detecting process more effectively.

4. Intermediate/Preliminary Results

A lot of time is spent on labeling the cluttered testing images. I am still training the detectors using part based models.

5. References

[1] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan. Object Detection with Discriminatively Trained Part Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 9, September 2010.

[2] Hema Swetha Koppula, Abhishek Anand, Thorsten Joachims, Ashutosh Saxena. Labeling 3D scenes for Personal Assistant Robots. In *RSS workshop on RGB-D cameras*, 2011

[3] Krystian Mikolajczyk, Bastian Leibe, Bernt Schiele. Multiple Object Class Detection with a Generative Model. In *CVPR*, 2006.

6. Appendix

My course project is part of a larger project in vision lab.