Course Project Milestone: Real-time object detection and recognition

Andrej Karpathy
Stanford
karpathy@cs.stanford.edu

Abstract

Real-time, scalable, multi-view object instance detection is an active area of research in computer vision. An efficient template-based object detection algorithm has recently been proposed [2] that utilizes both color and depth information, and works on texture-less objects. However, the template-based approach scales linearly with the number of objects and views. This project uses the same efficient feature extraction algorithm, but we replace template matching with a part-based model and a fixed-sized template dictionary.

1. Introduction

[2] presents a method for detecting objects that works in real-time, under heavy clutter, does not require a time-consuming training stage, and can handle untextured objects. The algorithm accomplishes this goal by computing feature templates of each object from various views, and storing them in memory. During test time, an efficient matching algorithm compares every patch of an image against the stored templates to detect objects in a scene. The proposed feature templates can leverage both color information and depth information, gathered by sensors such as Kinect.

The problem with [2] is that it slows down linearly with each new object or object view learned. In addition, the features are sparse so they don't fit into existing scaling algorithms such as Locality Sensitive Hashing, or approximate nearest neighbours.

2. Approach

The features proposed in [2] have some very desirable properties, including the fact that unlike other feature representations such as HOG, they are very efficient to compute and match while retaining discriminative power. Like HOG, the features are computed based on edges and therefore have the additional benefit of being able to distinguish texture-less objects. A visualization of the computed features can be seen in Figure 4. Finally, the feature extraction step is agnostic to the source of the input, so it is relatively easy to extend the method to compute features from other modalities, such as color or depth maps.

Our goal is to take advantage of the computational benefits of this feature representation, but substitute a more scalable approach for object detection than template matching. Inspired by the success of part-based models in detection [1, 5] and constellation models [3], we detect parts of objects and combine individual detections to produce the final output. In addition, to avoid the computational costs associated with a template dictionary that grows linearly with the number of objects and views, we use a fixed-sized dictionary of random feature templates to detect individual parts. This approach is inspired by recent work [4] that suggests that even random projections of input data can yield sufficiently discriminative features.

3. Progress

The data for our experiment comes from two sources. First, we use an existing RGBD objects dataset. Second, we gather our own data using the Kinect sensor and a turntable.

3.1. Dataset

We use an existing dataset that consists of 20 objects such as cups, tea boxes, detergent bottles, cans, statues, posters, and other ordinary items. For every object there is an associated library of about 80 RGBD images from distinct angles. The object is located in the center of the turntable on each image, and a binary mask is also provided for each image to help with object segmentation during training. An example of a training image and a mask image in this dataset is shown in Figure 1.

3.2. Hardware setup

A Kinect sensor with a turntable that is colored with a calibration pattern can be used to gather additional data if necessary. Additionally, this setup can be used to detect

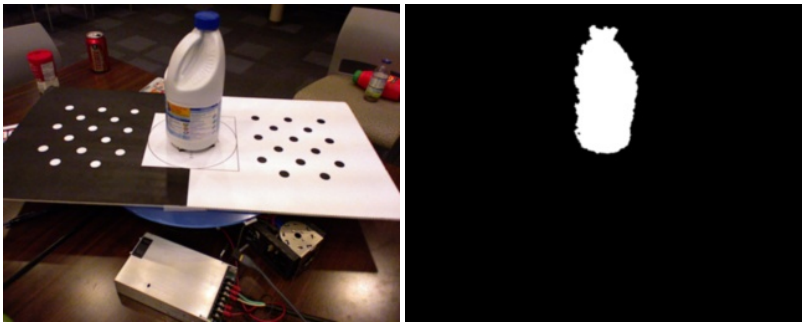

Figure 1. Example of an image and a mask from the dataset.

learned objects in real time, or potentially for online learning of new objects.

4. Preliminary results

4.1. Baseline performance

Existing OpenCV code in C++ was provided that contains functions to extract the features and detect objects using template matching. The code was modified to run on the 20 objects dataset and the resulting performance was measured as a baseline.

First, using only 3 objects from the dataset the baseline method performs well using gradient and color feature modalities, as can be seen on Figure 5. With a detector threshold set at about 0.97 we identify about three quarters of all positives while keeping the false negative rate at 0.

We also evaluated the entire dataset of 20 objects using threshold of 0.97. Out of a total of 392 test images that contain one true positive each, the algorithm identifies 246 true positives, but also incorrectly detects 343 other patches.

The baseline method takes about 4 seconds to run on a standard desktop computer per image when using the entire training set from 20 objects (1149 object/view templates).

4.2. Preliminary experiments

4.2.1 Classification with bag of words

In this experiment we investigate the discriminative power of a bag of words representation using a random dictionary of features. For now we ignore color and only use edge orientations as features.

As visual words we consider using 200 random templates of size 15x15 pixels. To produce a bag of words histogram feature for a region, we match every template on every position in the region and increment the histogram count for the template that matches best. A bag of words approach is an attractive possibility because feature histograms over rectangles in the image can be computed efficiently using dynamic programming with integral histograms.

To test the discriminative power of this simple approach, we use images from all objects, crop them to the part of the image that contains the object using the ground truth bounding box, and classify every bounding box using approximate nearest neighbour classifier over the normalized feature histograms. This gives a result of 20.7% accuracy (where 5% is random guessing). Using Weka machine learning toolkit and their brute force learning algorithm selection tool, the best accuracy is 46%, obtained using a Random Forest.

The above results suggest that gradient features alone with bag of words representation are not very powerful even for discrimination. Future experiments will include color and depth features, as well as representations that retain some of the spatial statistics of the data. A simple possibility is to compute a few histograms in every bounding box instead of only one. For example, we could separate the bounding box into top and bottom, and treat them independently.

4.2.2 Hough transform for detection

An attractive idea inspired by [3] is to detect objects using Hough Transform.

The idea is as follows. We consider a fixed dictionary of N random templates. During training time, we first detect features on an object, and based on their relative offset from the center of the object, we update their distribution of the location of the object in the coordinate frame of that feature. After training, we can think of every feature as a weak classifier that can vote on the location of an object in the image. At test time, we detect all features in the image and overlay their votes to produce object predictions.

A benefit of this approach is that it has the potential to be very fast. Training simply corresponds to updating counts for every activated feature, and testing reduces to adding maps of all detected features and triggering detections for any location with activations above a threshold. The downside is that the method trades space for time, as we have to maintain a separate location map for every feature, and for every object. The space requirements may prevent us from

Figure 2. Examples of detection with the baseline method and threshold of 0.97. Red and yellow indicate false positives, white indicates true positive, and the green rectangle is the ground truth bounding box of the object.

storing all maps in memory.

The idea described above was implemented in MATLAB and tested on a single object with N=100 templates of size 15x15 pixels. Preliminary results suggest that the algorithm can reliably and robustly detect the object in all images of that object using gradient features alone. However, further tests are required to investigate the discriminative power of this method. Example output is shown in figure 3.

5. Future work

More work is necessary to properly integrate color and depth features with the current gradient features. Some of the ideas outlined above must be investigated in more detail and properly evaluated against the baseline. Other part-based models should be explored as being potentially useful.

Finally, it should be possible to optimize the algorithm enough that it can run in real-time on Kinect.

References

[1] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4. http://www.cs.brown.edu/ pff/latent-release4/.

[2] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. 2011.

[3] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77(1-3):259–289, May 2008.

[4] A. Saxe, P. Koh, Z. Chen, M. Bhand, B. Suresh, and A. Ng. On random weights and unsupervised feature learning. In *Workshop: Deep Learning and Unsupervised Feature Learning (NIPS)*, 2010.

[5] Y. Yang and D. Ramanan. Articulated pose estimation using flexible mixtures of parts. 2011.

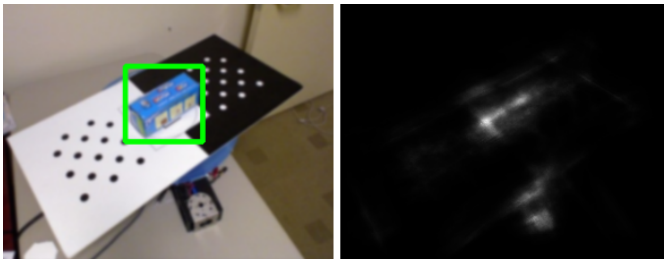

Figure 3. Result of applying the feature Hough Transform idea for one object. Here, 100 random templates of size 20x20 pixels were used. The box is correctly localized.

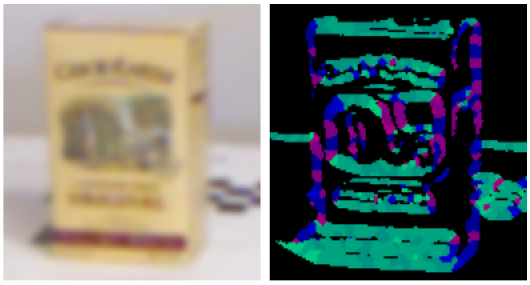

Figure 4. Example of an image and the computed gradient features. Color corresponds to the orientation of the gradient at that point. The image is slightly blurred to provide more stable output.

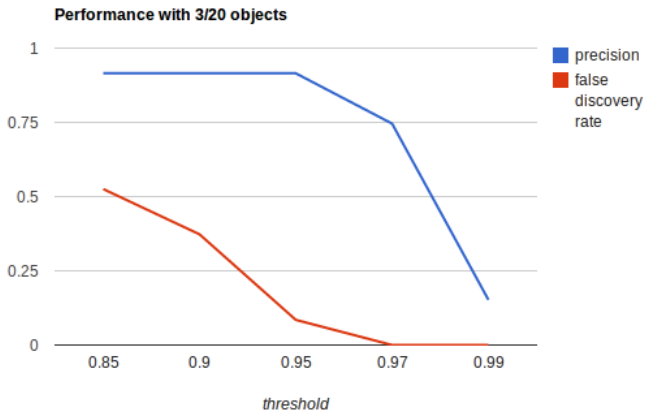

Figure 5. Baseline performance on 3/20 objects as a function of the detector threshold. Both color and gradient modalities are used in the detector.