Dining Scene Recognition Using Related Object Detection

Chaohao Wang chaohaow@stanford.edu

Gilbert Fu gilfu@stanford.edu Alan Quach alang@stanford.edu

Introduction

The goal of this project is to recognize dining scenes with the help of related object detection. For example, objects such as plates, glasses, bottles and cups will signify a restaurant setting. We will base our object detection algorithm on the paper "Histograms of Oriented Gradients for Human Detection" by N. Dalal et al. In this project, we will train objects individually by using techniques like histograms of oriented gradients (HOG) and SVM. Part of the training data are from PASCAL VOC challenge and the rest are from Bing image search and our personal pictures. After individual models are trained, test pictures will be scanned with boxes of different scales and a score for each object match will then be computed. Afterwards, the probabilities of each object model output and percentage of the image the objects appears in are used to compute the feature vectors for another SVM model that will predict a dining scene or not. The block diagram of the system is shown in Fig. 1.

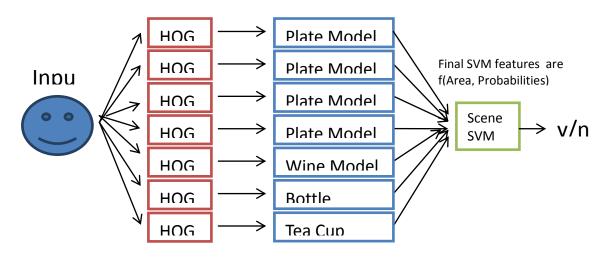


Figure 1 Block Diagram of The System

Histogram of Oriented Gradients

Histogram of Oriented Gradients (HOG) is an effective feature descriptor in computer vision for object detection. It counts the occurrences of gradient orientation in a small region of an image. The idea behind HOG is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. This normalization within a block can be done to get better invariance to changes in illumination or shadowing. In our project, we used cell size of 8, block size of 2 and 9 different orientation bins in 180 degrees. The HOG diagrams different objects are plotted in Fig. 2. The orientation of the line segments correspond to the highest weighed bins in HOG of cells.

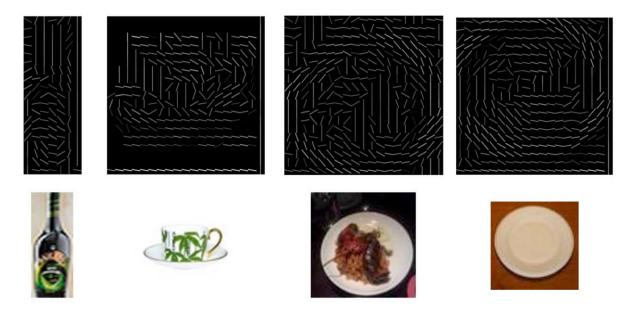


Figure 2 HOG Diagrams for different objects

SVM

We use a SVM for our classifier on each model. The SVM code we obtained from the Library of Support Vector Machines by Chih-Chung Chang and Chih-Jen Lin. We used their Matlab SVM libraries to help train and classify using the different dining scene objects' HOG vectors as our input features for positive examples. The kernel we used is the radial basis function: exp(-gamma*|u-v|^2).

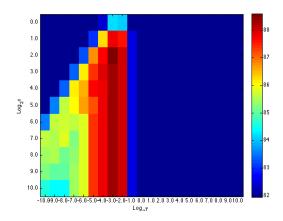


Figure 3 CR heat map for bottles

So there are 2 parameters in the optimization problem gamma in the kernel function and cost function in regulation. CR heat map is plotted for each model showing the fold-5 cross validation results for different gamma and C. CR heat map of bottle is shown in Fig. 3. Optimum values for these two parameters can be found in the heat map.

Data Collection

Part of the training data is from PASCAL VOC challenge. The data package includes images with different objects' bounding boxes labeled that we can directly feed into HOG after resizing. We also collected many personal pictures and pictures manually from Bing/Google's image search results and by using a Ruby script that interfaces Bing's search API to grab hundreds of search image results. Manual filtering

was done to remove bad images. Afterwards we had to crop and/or resize each image so that each image is uniform in dimension before calculating the HOG. Cropping and resizing was done using Matlab. About 150 pictures were collected for each object and roughly 700-800 pictures were used for negative data. The negative examples we used were HOG vectors from various vacation photos and other random photos. To reduce the number of false positives we ran the trained models over hundreds of negative pictures. The positives results from these runs over non dining pictures are then false positives, which we then re-fed into our training data as a negative example and created a more accurate retrained model.

Object Models

Models for bottles, plates, wine glasses, and teacups are trained with different image sizes. We decided to use optimum image height of 80 pixels to balance the trade-off between speed and accuracy. A training set of a few hundred pictures were passed through a HOG function and then into a SVM for our classifier on each model. Here are the fold-5 cross validation results of each model.

bottle	cup	wine glass	empty plate	plate with food	Angled empty plate	Angled plate with food
89.1%	92.2%	99.1%	99.6%	96.7%	99.5%	97.7%

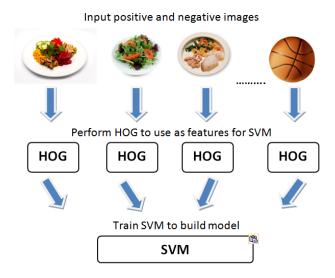


Figure 4 Object Model Training

Applying Models to Pictures

These results generated by using a Matlab code that scans through the entire image, taking small subimages of various sizes of the image and passing it through the SVM classifier with a generated model.

For testing speed purpose, all the testing images are scaled down to at most 600px wide. When applying different models to the real pictures, 6 different scales of bounding boxes (1, 1.3, 2, 2.6, 3.38, 4.39, 5.71) are used to scan through pictures to detect objects with different scales. During scanning, one quarter of the bounding box width/height is used as the scanning step as shown in Fig. 5. After grabbing the subimage inside the bounding box, the sub-image is scaled back to the original size of the model and fed into different object detectors.

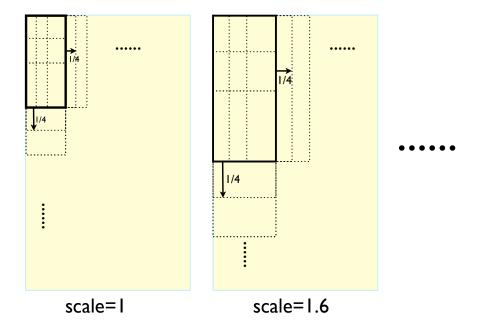


Figure 5 Scanning and Recognizing

Some of the objection detection results (including false positives) are shown in Fig. 6. The highlighted boxes indicate a positive match for a tested object and the different colors correspond to different models.



Figure 6 Object Detection Results

Dining Scene Model

Two pieces of information we get from each object detected are the area and probability of the prediction. Intuitively, the more area that objects related to dinning scene occupy the picture higher the probability the picture is positive. Also the more confident the object detector is, the more likely the picture is a dining scene picture. Besides we want to highly differentiate objects detected with low positive probability (close to 50% positive) and objects with high positive probability since the former could be false positives. So the empirical expression we use the generate feature vector element score is

$$S_j = \sum_i \frac{A_i}{A_{pic}} e^{c(P_{ij} - 0.5)}$$

Where Ai is the bounding box area, A_{pic} is the total area of the picture. P_{ij} is the probability of i_{th} -detected box for j_{th} item model. C is a scaling parameter for the probability.

Different objects will have different scores based on the equation above. Different object scores should be weighted differently when predicting the result. Object like plates should have high weight since it is a strong indicator. Here we didn't explicitly learn the weight of each object but use SVM instead. The final feature vector for scene detection is $[S_1 S_2 S_3 ... S_n]$ where n is the number of object models we have. In our case n=7. Scene SVM model is trained on 90 pictures with objects detected by our object models.

Results

After training and creating our full dining scene recognition model we tested it on a test size of 100 pictures. Our model correctly predicted 85 images to be dining or non dining scenes so its accuracy is 85%. The model's precision is 41/44 and its recall is 41/53 giving an F1 score of 84.54. We believe that for an application, it is best and most user friendly if the false positives were minimized. So we value a high precision rate more rather than a high recall. For the images shown above in the Applying Models to Pictures section, our model correctly predicts Pictures 1,2,3,4,5,7, and 8 as dining scenes and pictures 6 and 10 as non dining scenes. The one mistake is the false positive classification of Picture 9. This false positive image is reasonable for our model to classify the baby in the chair as a plate of food as the chair is also round like a plate.

Future Work

To improve our models we need to collect a lot more training data for the various different shapes of plates, glasses, and cups. Each of these different shapes should also have multiple orientations data on the different angles of plates must be collected. Currently, our prediction model runs too slow for real time classification of a photo bank as each picture takes roughly 3 to 4 minutes. Parallelizing each model on a separate process should improve the speed drastically. A deformable model detector for incomplete objects can also be useful for our project because objects are often cut off in the picture of behind other objects.

References

Chang, Chih-Chung, and Chih-Jen Lin. "LIBSVM: a library for support vector machines." ACM Transactions on Intelligent Systems and Technology (TIST)2.3 (2011): 27.

Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2005.